EVOLUTIONARY ADAPTIVE FILTERING BASED ON COMPETING FILTER STRUCTURES

Marcus Zeller and Walter Kellermann

Multimedia Communications and Signal Processing,
Friedrich-Alexander-Universität Erlangen-Nürnberg,
Cauerstr. 7, 91058 Erlangen, Germany,
e-mail: {zeller,wk}@LNT.de, web: http://lms.lnt.de/

ABSTRACT

This paper presents a novel filtering scheme that realizes a general, fully adaptive structure where both coefficients and required memory size are identified automatically. In particular, no distinction between linear or nonlinear models is made, since the filter structure can evolve into either a linear or a second-order Volterra filter. This is achieved by monitoring the mixing variables of various combinations where differently-sized competing filters are used. Using a set of intuitive rules along with desired step sizes for memory size changes, a dynamically growing/shrinking model structure is realized. The effectiveness of the approach for a fast-converging identification of arbitrary unknown systems is shown by means of an acoustic echo cancellation task where realistic linear and nonlinear systems as well as stationary and nonstationary input signals are considered.

1. INTRODUCTION

Adaptive Filters are an essential tool in a number of digital signal processing applications, most notably for the tasks of equalization, prediction and system identification [1]. Although perfect linearity is usually assumed for the sake of simplicity, many practical scenarios will exhibit a certain amount of nonlinearity as well, and therefore would benefit from suitable nonlinear models. To this end, the Volterra filter has received much attention since its transversal structure is rather general and adaptive realizations can straightforwardly be obtained from corresponding algorithms in the linear case. However, a major challenge of these models is the large number of coefficients to be estimated if no *a priori* knowledge on the model size is available.

In this work, we extend the methodology of competing filters as first presented in [2] and recently refined in [12], so as to provide a *completely adaptive* Volterra structure whose kernel sizes are automatically adjusted to fit the underlying system. Thereby, no distinction between linear or second-order nonlinear models has to be made, since the continuous probing for nonlinear contributions is inherently achieved by the employed hierarchical combination of different components. The performance of all individual combinations is then used to draw conclusions about the superior memory configuration and adjust the actual kernels. This essentially results in a fully adaptive filter structure where both the filter coefficient values as well as the size dimensions of the model are estimated and tracked throughout the whole processing.

The rest of this paper is structured as follows: A generic system identification scenario is reviewed in Sec. 2 where the output of an unknown system is to be replicated by an appropriate adaptive filter. To this end, the evolutionary model structure approach is presented in Sec. 3 and the corresponding memory control algorithm is outlined in Sec. 4. The effectiveness of this approach is demonstrated for a nonlinear acoustic echo cancellation scenario with fixed and time-varying systems as well as noise and speech data in Sec. 5, before Sec. 6 gives some conclusions.

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) under contract number KE 890/5-1.

2. GENERIC SYSTEM IDENTIFICATION SCENARIO

The task of identifying an arbitrary unknown system is depicted in Fig. 1. Thereby, $x(k)$ denotes the discrete-time input signal, whereas $y(k), n(k)$ and $d(k) = y(k) + n(k)$ represent the system's output, measurement noise and obtained reference signal, respectively. In order to identify the underlying parameters of the system, the coefficient vector $\underline{\widehat{\mathbf{h}}}(k)$ of an adaptive model is adjusted in parallel to this signal path, such that its output $\widehat{y}(k)$ minimizes the residual error

$$e(k) = d(k) - \widehat{y}(k) = \big[y(k) - \widehat{y}(k)\big] + n(k) \tag{1}$$

However, as opposed to the usual scenario and as indicated by the dashed box, the adaptive filter *structure* itself is assumed to be unknown here and is subject to an iterative refinement. In particular, this implies that $\underline{\widehat{\mathbf{h}}}(k)$ may either represent a purely linear impulse response or comprise all coefficients of a nonlinear transversal model.

For illustration, we restrict the consideration here to a second-order Volterra filter (VF) [3]. The discrete-time output of such a structure is given by the superposition

$$\widehat{y}(k) = \widehat{y}_1(k) + \widehat{y}_2(k), \tag{2}$$

where $\widehat{y}_1(k) = \sum_{n=0}^{N_1-1} \widehat{h}_{1,n}(k)x(k-n)$ is given by linear convolution, whereas the second-order nonlinear output is computed as

$$\widehat{y}_2(k) = \sum_{w=0}^{W-1} \sum_{n=0}^{N_2-w-1} \widehat{h}_{2,w,n}(k)x_w(k-n). \tag{3}$$

Note that (3) uses the so-called diagonal-coordinate representation, where only the regression data $x_w(k) := x(k)x(k-w)$ within a certain *width* W of the two-dimensional quadratic kernel $\widehat{h}_{2,n_1,n_2}(k)$ are accounted for (see [4] for illustration). In concise notation,

$$\underline{\mathbf{x}}(k) := \big[x(k), \ldots, x(k-N_1+1)\big]^{\mathrm{T}}, \tag{4}$$

$$\underline{\mathbf{x}}_w(k) := \big[x_w(k), \ldots, x_w(k-N_2+w+1)\big]^{\mathrm{T}}, \tag{5}$$

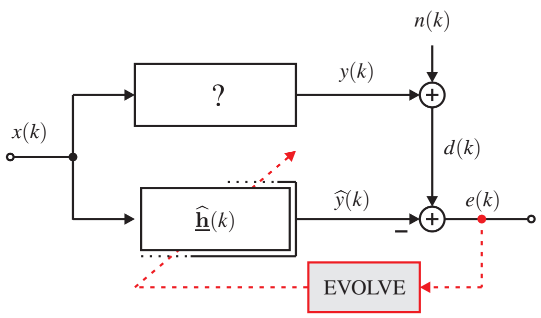

Figure 1: Generic identification scenario for an unknown system by an evolutionary adaptive filter structure (EVOLVE).

are formed by proper stacking of the linear input and the data of all diagonals $0 \leq w \leq W-1$. Likewise, the filter coefficients are grouped as

$$\widehat{\underline{\mathbf{h}}}_1(k) := \left[\widehat{h}_{1,0}(k), \ldots, \widehat{h}_{1,N_1-1}(k)\right]^{\mathrm{T}}, \tag{6}$$

$$\widehat{\underline{\mathbf{h}}}_{2,w}(k) := \left[\widehat{h}_{2,w,0}(k), \ldots, \widehat{h}_{2,w,N_2-w-1}(k)\right]^{\mathrm{T}}. \tag{7}$$

Using these definitions, (2) is simplified to

$$\widehat{y}(k) = \widehat{\underline{\mathbf{h}}}_1^{\mathrm{T}}(k)\,\underline{\mathbf{x}}(k) + \sum_{w=0}^{W-1} \widehat{\underline{\mathbf{h}}}_{2,w}^{\mathrm{T}}(k)\,\underline{\mathbf{x}}_w(k), \tag{8}$$

which reveals a multiple input/single output (MISO) structure [5] with convolutions in all "channels" having inputs $\underline{\mathbf{x}}(k)$ and $\underline{\mathbf{x}}_w(k)$.

As can be seen from the definitions (4) – (7), the overall complexity of the filtering (8) crucially depends on the size parameters N_1, N_2 and W. Assuming that any underlying system can be characterized by some *optimum but unknown* memory parameters $N_{1,\mathrm{opt}}, N_{2,\mathrm{opt}}$ and W_{opt}, it is clearly desirable to perform the identification task with an adaptive filter of nearly matched size. Otherwise, either under- or overmodeling situations are likely to occur, resulting in either loss of performance or wasteful use of resources and increased gradient noise [1, 3]. In the following, we thus extend the previous approaches [8, 12] such that time-variant estimates of *all* parameters are computed concurrently to the actual adaptation of the coefficients and used for a self-configuration of the model.

In this context, it should also be pointed out that the above diagonal-coordinate VF comprises several special cases: First, a purely linear filter is obtained for $N_2 = 0$, resulting in $y_2(k) = 0$ in (2). On the other hand, a fully populated Cartesian-coordinate VF is realized for $W = N_2$, whereas $W < N_2$ describes second-order kernels with a smaller "width", down to the power filter case ($W = 1$) where only input products from the same time instant are taken into account [6]. Hence, the VF is well-suited to dissolve the strict discrimination between linear and nonlinear models and thus is highly attractive for a more generic identification of arbitrary systems.

3. COMPETING FILTERS SCHEME

As mentioned above, the required memory sizes for the identification of an unknown system are generally unknown as well. Therefore, the goal of the proposed approach is to obtain reasonably close estimates $\widehat{N}_{1,\mathrm{opt}}, \widehat{N}_{2,\mathrm{opt}}$ of the principal kernel sizes as well as an estimate $\widehat{W}_{\mathrm{opt}}$ of the number of necessary (i.e. non-zero) diagonals in the quadratic kernel. To this end, the methodology of filter combinations as first presented in [7] has already been adopted to determine the optimum number of diagonals for the identification of a second-order Volterra kernel [2]. Moreover, the estimation of the optimum length of an adaptive FIR filter given any unknown linear system has been proposed in [8].

In this contribution, however, we seek a *joint estimation* of all size parameters of a second-order VF. Since both memory sizes *and* coefficients are estimated concurrently, the resulting EVOLVE (EVOLutionary Volterra Estimation) scheme ultimately yields a fully adaptive filter structure and thus offers great flexibility, including a smooth transition from linear to nonlinear models. The proposed scheme is depicted in Fig. 2, where it can be seen that two Volterra *kernels of the same type* are combined at each stage. As opposed to the usual step-size control application [9], all components in the considered scheme are operated with same step sizes, but employed with different memory parameters. Using $\mathrm{c}, \mathrm{s} \in \{\mathrm{A}, \mathrm{B}\}$ to denote all components and subcomponents, respectively, the corresponding parameters $N_1^{[\mathrm{c}]}(k), N_2^{[\mathrm{s}]}(k)$ and $W^{[\mathrm{c}]}(k)$ are realized and updated over time.

As can be seen from Fig. 2, the total output $\widehat{y}(k)$ is given according to (2). The linear kernel output is thereby given as

$$\widehat{y}_1(k) = \eta_1(k)\,\widehat{y}_1^{[\mathrm{A}]}(k) + \left[1 - \eta_1(k)\right]\widehat{y}_1^{[\mathrm{B}]}(k), \tag{9}$$

representing the combined output from both linear kernel components $\widehat{\underline{\mathbf{h}}}_1^{[\mathrm{A}]}(k)$ and $\widehat{\underline{\mathbf{h}}}_1^{[\mathrm{B}]}(k)$, where $\eta_1(k)$ is a convex mixture weight. On the other hand, the quadratic kernel is implemented by a two-stage, hierarchical combination, resulting in a total of four different kernel configurations with $N_2^{[\mathrm{s}]}(k)$ and $W^{[\mathrm{c}]}(k)$. Hence, the mixture

$$\widehat{y}_2(k) = \eta_2(k)\,\widehat{y}_2^{[\mathrm{A}]}(k) + \left[1 - \eta_2(k)\right]\widehat{y}_2^{[\mathrm{B}]}(k), \tag{10}$$

in the outer stage is given similar to the linear kernel (9), whereas

$$\widehat{y}_2^{[\mathrm{c}]}(k) = \eta_2^{[\mathrm{c}]}(k)\,\widehat{y}_2^{[\mathrm{c,A}]}(k) + \left[1 - \eta_2^{[\mathrm{c}]}(k)\right]\widehat{y}_2^{[\mathrm{c,B}]}(k), \tag{11}$$

yields the inner combinations with the respective subcomponents. According to these individual outputs $\widehat{y}_1^{[\mathrm{c}]}(k), \widehat{y}_2^{[\mathrm{c}]}(k)$ and $\widehat{y}_2^{[\mathrm{c,s}]}(k)$, various associated residual errors can be computed as well. Modifying (1), these are defined by

$$e_1^{[\mathrm{c}]}(k) = d(k) - \left[\widehat{y}_1^{[\mathrm{c}]}(k) + \widehat{y}_2(k)\right], \tag{12}$$

$$e_2^{[\mathrm{c}]}(k) = d(k) - \left[\widehat{y}_1(k) + \widehat{y}_2^{[\mathrm{c}]}(k)\right], \tag{13}$$

$$e_2^{[\mathrm{c,s}]}(k) = d(k) - \left[\widehat{y}_1(k) + \widehat{y}_2^{[\mathrm{c,s}]}(k)\right], \tag{14}$$

and thus are always based on the individual kernel of the same type and the combined kernel of different type. Note that this can easily be extended to higher-orders as outlined in [9]. Using (12) – (14), NLMS-type coefficient updates are finally given by

$$\widehat{\underline{\mathbf{h}}}_1^{[\mathrm{c}]}(k+1) = \widehat{\underline{\mathbf{h}}}_1^{[\mathrm{c}]}(k) + \frac{\alpha_1}{P_1^{[\mathrm{c}]}(k) + \delta}\,e_1^{[\mathrm{c}]}(k)\,\underline{\mathbf{x}}^{[\mathrm{c}]}(k), \tag{15}$$

$$\widehat{\underline{\mathbf{h}}}_{2,w}^{[\mathrm{c,s}]}(k+1) = \widehat{\underline{\mathbf{h}}}_{2,w}^{[\mathrm{c,s}]}(k) + \frac{\alpha_2}{P_2^{[\mathrm{c,s}]}(k) + \delta}\,e_2^{[\mathrm{c,s}]}(k)\,\underline{\mathbf{x}}_w^{[\mathrm{c,s}]}(k), \tag{16}$$

and hence performed separately for each linear kernel and second-order diagonal. Although α_1 and α_2 denote kernel-dependent but fixed step-size parameters (δ is a small regularization constant), a normalization to the short-time kernel energies is given by

$$P_1^{[\mathrm{c}]}(k) = \sum_{n=0}^{N_1^{[\mathrm{c}]}(k)-1} x^2(k-n), \tag{17}$$

$$P_2^{[\mathrm{c,s}]}(k) = \sum_{w=0}^{W^{[\mathrm{c}]}(k)-1} \sum_{n=0}^{N_2^{[\mathrm{s}]}(k)-w-1} x_w^2(k-n). \tag{18}$$

It should moreover be emphasized that the corresponding vectors (4) – (7) are also based on the *current* values for the kernel sizes $N_1^{[\mathrm{c}]}(k), N_2^{[\mathrm{s}]}(k)$ and numbers of diagonals $W^{[\mathrm{c}]}(k)$.

For simplicity and robustness, the actual weighting involved in the combinations (9) – (11) performs a convex mixing such that

$$0 < \eta_1(k), \eta_2(k), \eta_2^{[\mathrm{c}]}(k) < 1 \quad \forall\, k \tag{19}$$

holds. This is ensured by defining the mixing via a sigmoid, such that

$$\eta_2^{[\mathrm{c}]}(k) = \left[1 + \mathrm{e}^{-a_2^{[\mathrm{c}]}(k)}\right]^{-1} \tag{20}$$

where $a_2^{[\mathrm{c}]}(k)$ denotes a time-variant control parameter that is adapted at each time step so as to minimize the resulting squared error after combination [10]. For robustness, normalized updates

$$a_2^{[\mathrm{c}]}(k+1) = a_2^{[\mathrm{c}]}(k) + \mu_a \frac{\eta_2^{[\mathrm{c}]}(k)\left[1 - \eta_2^{[\mathrm{c}]}(k)\right]}{P_{\Delta,2}^{[\mathrm{c}]}(k)}\,e_2^{[\mathrm{c}]}(k)\,\Delta e_2^{[\mathrm{c}]}(k) \tag{21}$$

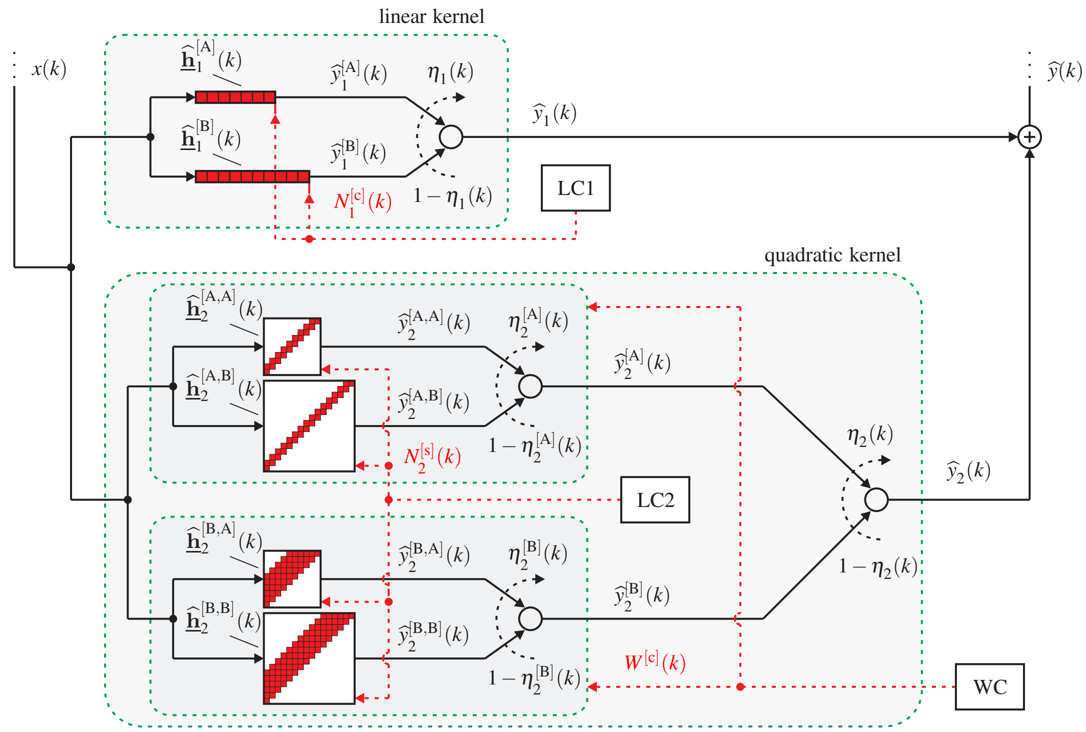

Figure 2: Adaptive filter structure based on the EVOLVE scheme with evolutionary linear and second-order nonlinear contributions (LC1/LC2 = length control, WC = width control). Competing Volterra kernels of different size are employed in order to obtain estimates $\widehat{N}_{1,\mathrm{opt}}, \widehat{N}_{2,\mathrm{opt}}, \widehat{W}_{\mathrm{opt}}$ of the unknown optimum memory parameters.

with a common fixed step size μ_a are implemented for all combinations in Fig. 2. These updates are obviously driven by the "input"

$$\Delta e_2^{[c]}(k) := e_2^{[c,B]}(k) - e_2^{[c,A]}(k), \tag{22}$$

representing the error difference between the (sub-)components and are normalized by the smoothed power [11]

$$P_{\Delta,2}^{[c]}(k) := 0.9\,P_{\Delta,2}^{[c]}(k-1) + 0.1\left[\Delta e_2^{[c]}(k)\right]^2. \tag{23}$$

Note that $\eta_1(k)$ and $\eta_2(k)$ and their updates are defined analogously and that the range of all controls $a_1(k), a_2(k)$ and $a_2^{[c]}(k)$ is typically limited to some interval $[-a_{\max}, +a_{\max}]$ [10].

As analyzed in detail in [10], the effect of the combination is as follows: The mixing variables (19) are continuously adjusted in order to obtain the best possible performance based on its given (sub-)components A or B. For instance, $\eta_2^{[A]}(k) \to 1$ whenever $\widehat{\mathbf{h}}_{2,w}^{[A,A]}(k)$ is a better model for the corresponding part of the true system than $\widehat{\mathbf{h}}_{2,w}^{[A,B]}(k)$. In that sense, the value of each mixing variable can also be interpreted as a *soft switch*, deciding for the currently superior of two competing models that also produces a lower mean squared error.

4. MEMORY EVOLUTION ALGORITHM

In the following, the above-mentioned soft decision property is now exploited in order to control the used memory of the complete adaptive filter. Due to a continuous monitoring of the achieved filter performance, an estimate of the optimum memory parameters can then be obtained over time. As graphically indicated in Fig. 2 all existing kernels are realized with different sizes. In more detail,

$$N_1^{[B]}(k) = N_1^{[A]}(k) + N_{1,\mathrm{dist}}, \tag{24}$$
$$N_2^{[B]}(k) = N_2^{[A]}(k) + N_{2,\mathrm{dist}}, \tag{25}$$
$$W^{[B]}(k) = W^{[A]}(k) + W_{\mathrm{dist}}, \tag{26}$$

such that the (sub-)components B are always larger. Although the effective size parameters are implemented time-variantly, $N_{1,\mathrm{dist}}, N_{2,\mathrm{dist}}$ and W_{dist} denote fixed "distances" between the competing models that are kept throughout the whole processing.

In order to reliably assess the general trend of the filter performance, only long-term observations are to be taken into account [2]. Therefore, smoothed versions of all mixing variables, namely

$$\widetilde{\eta}_1(k) = \lambda\,\widetilde{\eta}_1(k-1) + [1-\lambda]\;\eta_1(k), \tag{27}$$
$$\widetilde{\eta}_2^{[c]}(k) = \lambda\,\widetilde{\eta}_2^{[c]}(k-1) + [1-\lambda]\;\eta_2^{[c]}(k), \tag{28}$$

are computed for the *length control* units (LC1 and LC2) of both kernels. Likewise,

$$\widetilde{\eta}_2(k) = \lambda\,\widetilde{\eta}_2(k-1) + [1-\lambda]\;\eta_2(k) \tag{29}$$

provides a lowpass-filtered version of $\eta_2(k)$ for the *width control* unit (WC) of quadratic kernel. For simplicity, the same forgetting factor is chosen for all decision variables and has to be chosen very close to one, i.e., $\lambda = 0.9995$ is used for all results in Sec. 5.

The actual changes in kernel memory are performed, according to a set of intuitive rules as follows. First, the length of *both* linear kernels $\mathrm{c} \in \{\mathrm{A}, \mathrm{B}\}$ are adjusted by

$$N_1^{[c]}(k+1) := \begin{cases} N_1^{[c]}(k) - \Delta N_1, & \text{if } 1-\varepsilon_{\mathrm{LC1}}^{-} \le \widetilde{\eta}_1(k) \\ N_1^{[c]}(k) + \Delta N_1, & \text{if } \quad \widetilde{\eta}_1(k) \le \varepsilon_{\mathrm{LC1}}^{+} \\ N_1^{[c]}(k), & \text{otherwise} \end{cases}, \tag{30}$$

where ΔN_1 denotes an (integer) size increment and $\varepsilon_{\mathrm{LC1}}^{+/-}$ are specified thresholds for the length increase/decrease. In a similar way, the two-stage control of the quadratic kernel size is realized. The principal length affecting all diagonals $\widehat{\underline{\mathbf{h}}}_{2,w}^{[\mathrm{c,s}]}(k)$ in the second-order kernels is thus changed according to

$$N_2^{[\mathrm{s}]}(k+1) := \begin{cases} N_2^{[\mathrm{s}]}(k) - \Delta N_2, & \text{if } \begin{array}{l} 1-\varepsilon_{\mathrm{LC2}}^{-} \leq \widetilde{\eta}_2^{[\mathrm{A}]}(k) \\ \wedge\ 1-\varepsilon_{\mathrm{LC2}}^{-} \leq \widetilde{\eta}_2^{[\mathrm{B}]}(k) \end{array} \\ N_2^{[\mathrm{s}]}(k) + \Delta N_2, & \text{if } \begin{array}{l} \widetilde{\eta}_2^{[\mathrm{A}]}(k) \leq \varepsilon_{\mathrm{LC2}}^{+} \\ \wedge\ \widetilde{\eta}_2^{[\mathrm{B}]}(k) \leq \varepsilon_{\mathrm{LC2}}^{+} \end{array} \\ N_2^{[\mathrm{s}]}(k), & \text{otherwise} \end{cases} . \tag{31}$$

Here, ΔN_2 again denotes the size increment and $\varepsilon_{\mathrm{LC2}}^{+/-}$ are thresholds that can be specified to obtain a desired adaptation sensitivity (see Sec. 5 for typical values). However, as can be seen from (31), the size will only be changed in case of coherent indications of *both* mixing variables $\widetilde{\eta}_2^{[\mathrm{c}]}(k)$, as otherwise contradicting situations may occur. In the outer combination, the number of diagonals in the second-order kernel is again adapted similarly to (30), i.e.,

$$W^{[\mathrm{c}]}(k+1) := \begin{cases} W^{[\mathrm{c}]}(k) - \Delta W, & \text{if } 1-\varepsilon_{\mathrm{WC}}^{-} \leq \widetilde{\eta}(k) \\ W^{[\mathrm{c}]}(k) + \Delta W, & \text{if } \quad\ \widetilde{\eta}(k) \leq \varepsilon_{\mathrm{WC}}^{+} \\ W^{[\mathrm{c}]}(k), & \text{otherwise} \end{cases} . \tag{32}$$

As before, the increase/decrease in diagonal width is defined using thresholds $\varepsilon_{\mathrm{WC}}^{+/-}$ and increments ΔW. Unlike the length increments $\Delta N_1, \Delta N_2$, it is, however, reasonable to change the number of diagonals only in very small steps such that $\Delta W = 1$ is a typical choice.

Since any change event enforces a readaptation phase, none of the above rules (30) – (32) are applied for a certain *settling time*. The latter is given by the average number of coefficients in the affected kernel type scaled by a factor τ_p, i.e.,

$$K(k) := \tau_p C_{p,\mathrm{avg}}(k), \tag{33}$$

where, depending on the kernel type associated to the changes,

$$C_{p,\mathrm{avg}}(k) := \begin{cases} \frac{1}{2} \sum\limits_{\mathrm{c}\in\{\mathrm{A,B}\}} N_1^{[\mathrm{c}]}(k) & , \text{ if } p=1 \\ \frac{1}{4} \sum\limits_{\mathrm{c,s}\in\{\mathrm{A,B}\}} \sum\limits_{w=0}^{W^{[\mathrm{c}]}(k)} N_2^{[\mathrm{s}]}(k) - w, & \text{if } p=2 \end{cases} . \tag{34}$$

To ensure a proper adaptation of the total filter structure, moreover, a precedence of (30) over (31) and (32) is enforced, since the linear kernel is generally the largest and most important.

Throughout the whole operation, desired estimates of the optimum memory sizes are then obtained by always regarding the smaller components which is briefly illustrated in the sequel: Starting from an undermodeling situation with small initial sizes, the larger B components will always be preferred as they provide an increased modeling power, thus triggering increasing lengths and/or widths. On the other hand, if the used size parameters of both adaptive filter components exceed the corresponding memory of the unknown system, the respective decision variable will tend towards zero, since the smaller component A still produces less gradient noise. In turn, this yields a decrease of the associated memory parameter. Accordingly, for $p = 1, 2$

$$\widehat{N}_{p,\mathrm{opt}}(k) := N_p^{[\mathrm{A}]}(k) \quad \text{as well as} \quad \widehat{W}_{\mathrm{opt}}(k) := W^{[\mathrm{A}]}(k). \tag{35}$$

Hence, the proposed EVOLVE approach allows for an identification of optimum filter sizes that can also be tracked continuously.

In addition to the straightforward implementation, inevitably requiring the parallel operation of several full linear and quadratic kernels, a much more efficient version can be outlined. The latter relies on a *virtualization* of filter components by exploiting the very close behavior of the first $N_p^{[\mathrm{A}]}(k)$ coefficients ($W^{[\mathrm{A}]}(k)$ diagonals) of all components and results in a significantly reduced complexity. For a detailed description we refer to the illustrations in [2, 12].

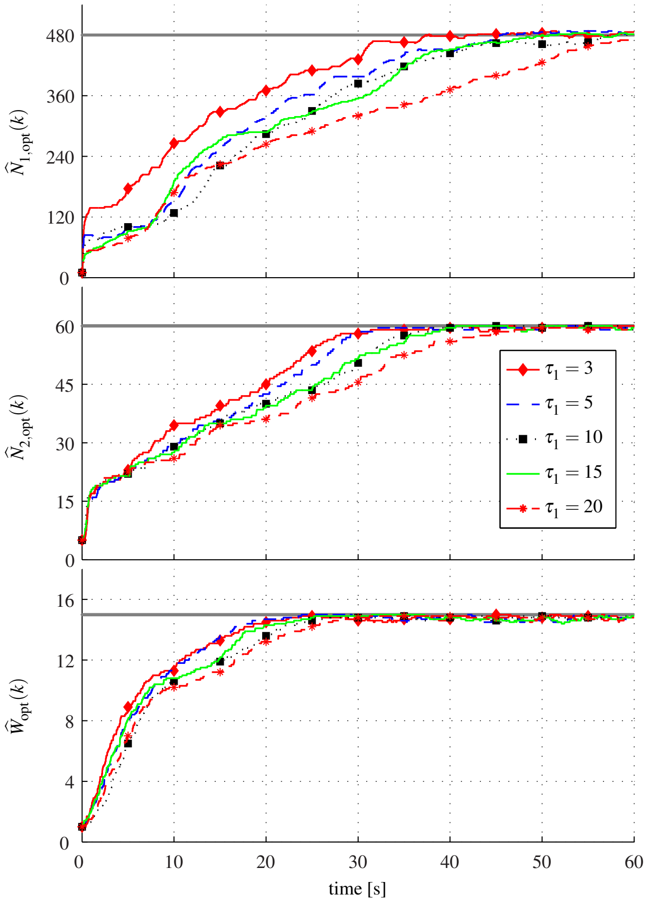

Figure 3: Evolution of kernel size estimates for white Gaussian noise input and various settling time factors τ_1 and $\tau_2 = 20$.

5. SIMULATION RESULTS

In order to show the effectiveness of the presented scheme in its efficient implementation [2, 12] under challenging conditions, Fig. 1 is interpreted as a nonlinear acoustic echo cancellation (NLAEC) scenario [6] at 8 kHz sampling rate. The microphone signal is thereby composed such that a signal-to-noise ratio of 30 dB as well as a 10 dB ratio of linear to nonlinear signal components is obtained. For the NLMS-type adaptation of all filter components, parameters $\alpha_{1/2} = 0.1/0.05$ and $\delta = 10^{-4}$ are used along with a simple kernel step-size control that compensates for the different speeds of convergence of different orders [12]. Moreover, the combination mixing is updated by $\mu_a = 0.3$ in all experiments. In order to represent the typical use case, the initial mixing values are defined by $a_1(0) = a_2(0) = a_2^{[\mathrm{c}]}(0) \equiv -4$, whereas all memory sizes are kept quite small and adjusted by $\Delta N_1 = 20, \Delta N_2 = 5$ and $\Delta W = 1$. Note that all results are averaged over 10 independent noise realizations such that non-integer parameter values may occur as well.

In a first experiment the temporal evolution of all optimum size estimates $\widehat{N}_{1,\mathrm{opt}}, \widehat{N}_{2,\mathrm{opt}}$ and $\widehat{W}_{\mathrm{opt}}$ is demonstrated for a second-order nonlinear system with white Gaussian noise input. The VF is thereby defined by $N_{1,\mathrm{opt}} = 480$, $N_{2,\mathrm{opt}} = 60$ and $W_{\mathrm{opt}} = 15$ whereas the competing kernels are initialized such that distances $N_{1,\mathrm{dist}} = 40$, $N_{2,\mathrm{dist}} = 10$ and $W_{\mathrm{dist}} = 2$ are used for probing increases in memory size. For the change decisions, $\varepsilon_*^* = 0.1$ is chosen for all threshold parameters (i.e. $*$ denotes any option). As can be seen from the results in Fig. 3, all kernel sizes are reliably estimated given sufficient convergence time. The influence of the settling time is thereby illustrated for various factors τ_1 whereas $\tau_2 = 20$. Although this mainly

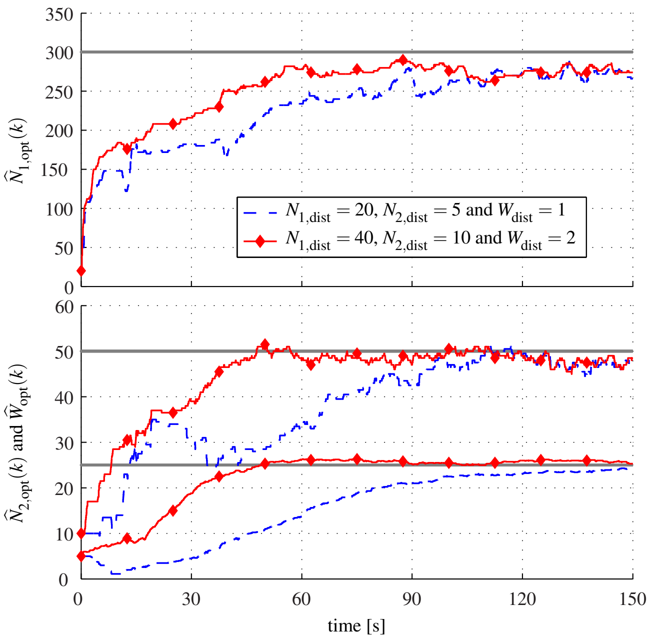

Figure 4: Estimates for two different "distance" settings and non-stationary input signal (male speech).

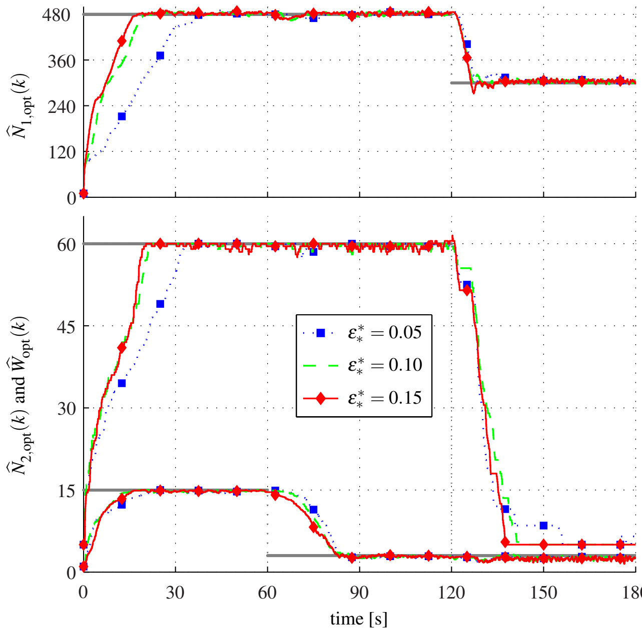

Figure 5: Tracking property of the EVOLVE scheme with white noise input and various changes (after 60 and 120 seconds).

affects the speed of convergence of $\widehat{N}_{1,\mathrm{opt}}(k)$, the corresponding estimates of the quadratic kernel differ as well, revealing the highly dynamic adaptive structure with ongoing memory reconfiguration.

Regarding nonstationary excitation signals, Fig. 4 shows similar results for a male speech input to a VF of size $N_{1,\mathrm{opt}} = 300$, $N_{2,\mathrm{opt}} = 50$ and $W_{\mathrm{opt}} = 25$ as indicated in the plots. Note that both second-order kernel parameters have been combined in the lower plot to save space. Here, threshold parameters $\varepsilon_*^* = 0.05$ and controls $\tau_{1/2} = 10$ are used in combination with two different size distance sets as given in the legend box of Fig. 4. As can clearly be seen from the obtained estimates, employing larger size differences generally improves the decision mechanism of the combination and, hence, the true optimum values are found more quickly and robustly. Nevertheless, this improvement comes at the expense of an increased steady-state complexity (see [12] for more information).

Finally, the principal tracking capability of the EVOLVE scheme is briefly demonstrated by the results in Fig. 5. Again, a second-order VF is employed with white Gaussian noise input and $\tau_{1/2} = 0.10$ as before. In the beginning the true system from Fig. 3 is used, i.e., $N_{1,\mathrm{opt}} = 480$, $N_{2,\mathrm{opt}} = 60$ and $W_{\mathrm{opt}} = 15$. However, after 60 and 120 seconds, the number of active diagonals is reduced to $W_{\mathrm{opt}} = 3$ whereas $N_{2,\mathrm{opt}}$ is kept before the quadratic kernel size is also truncated to $N_{2,\mathrm{opt}} = 3$. Note that the latter two stages therefore approximate the situations of a power filter [6] or an almost linear system (with $N_{1,\mathrm{opt}} = 300$ after 120 seconds). This shows the great flexibility of the approach, where all memory modifications can be followed well for the given threshold parameters ε_*^* — despite continuous adaptation of the incorporated filter coefficients.

6. CONCLUSIONS

We have presented a comprehensive adaptive filtering scheme where not only the coefficients but also the structure of the underlying system is continuously adjusted. This evolutionary behavior is realized by observing various competing filter kernels of different size and evaluating the performance of their convex combination weights. According to a set of intuitive rules and defined size increment, the filter structure is dynamically reconfigured and allows for growing/shrinking memory. In addition, the strict discrimination between linear and nonlinear models is overcome. Based on a nonlinear acoustic echo cancellation scenario using a second-order Volterra filter, the effectiveness of the approach has been demonstrated for both noise and speech signals and time-variant systems.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 4th edition, Upper Saddle River (NJ), USA, 2002.

[2] M. Zeller, L. A. Azpicueta-Ruiz, and W. Kellermann, "Online estimation of the optimum quadratic kernel size of second-order Volterra filters using a convex combination scheme," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009, pp. 2965–2968.

[3] W. J. Rugh, *Nonlinear System Identification*, John Hopkins University Press, Baltimore (MD), USA, 1980.

[4] M. Zeller and W. Kellermann, "Fast and robust adaptation of DFT-domain Volterra filters in diagonal coordinates using iterated coefficient updates," *IEEE Trans. on Signal Processing*, vol. 58, no. 3, pp. 1589–1604, March 2010.

[5] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*, Wiley, New York (NY), USA, 2000.

[6] F. Küch and W. Kellermann, *Topics in Acoustic Echo and Noise Control (Eds. Hänsler and G. Schmidt)*, chapter Nonlinear Acoustic Echo Cancellation, pp. 205–261, Springer, Heidelberg, Germany, June 2006.

[7] A. C. Singer and M. Feder, "Universal linear prediction by model order weighting," *IEEE Trans. on Signal Processing*, vol. 47, no. 10, pp. 2685–2699, Oct. 1999.

[8] M. Zeller, L. A. Azpicueta-Ruiz, and W. Kellermann, "Adaptive FIR filters with automatic length optimization by monitoring a normalized combination scheme," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz (NY), USA, Oct. 2009, pp. 149–152.

[9] L. A. Azpicueta-Ruiz, M. Zeller, A. Figueiras-Vidal, J. Arenas-García, and W. Kellermann, "Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 97–110, Jan. 2011.

[10] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean square performance of a convex combination of two adaptive filters," *IEEE Trans. on Signal Processing*, vol. 54, no. 3, pp. 1078–1090, March 2006.

[11] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-García, "A normalized adaptation scheme for the convex combination of two adaptive filters," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas (CA), USA, April 2008, pp. 3301–3304.

[12] M. Zeller, L. A. Azpicueta-Ruiz, J. Arenas-García, and W. Kellermann, "Adaptive Volterra filters with evolutionary quadratic kernels using a combination scheme for memory control," *IEEE Trans. on Signal Processing*, vol. 59, no. 4, pp. 1449–1464, April 2011.