PRACTICALLY EFFICIENT NONLINEAR ACOUSTIC ECHO CANCELLERS USING CASCADED BLOCK RLS AND FLMS ADAPTIVE FILTERS

Yiteng (Arden) Huang, Jan Skoglund, Alejandro Luebs

Google Inc., USA
{ardenhuang, jks, aluebs }@google.com

ABSTRACT

This paper presents a practically efficient implementation for nonlinear acoustic echo cancellation (NAEC). The echo path is modeled by a novel hybrid Taylor-Volterra pre-processor followed by a linear FIR filter. A cascaded block RLS and unconstrained FLMS adaptive algorithm is developed to jointly identify the pre-processor and the FIR filter. This implementation is validated via simulations.

Index Terms— Nonlinear acoustic echo cancellation, block RLS, FLMS, hybrid Taylor-Volterra model

1. INTRODUCTION

Acoustic echo cancellation (AEC) is one of the enabling technologies for two-party or multi-party voice communications, e.g., videoconferencing. Since the idea was first developed in the early 1960s, traditional AEC approaches have had many successful applications and real-time systems. But when coming to implementations on mobile devices, they are found to fall short of required performance, mainly because of the common assumption they made that echo paths are linear and can be modeled by linear finite impulse response (FIR) filters. On typically low cost and power efficient mobile devices, low-quality and overdriven audio components (e.g., converter, amplifier, loudspeaker, and microphone) can introduce nonlinear distortion which FIR filters cannot represent and deal with. This leads to strong residual echoes, which greatly impair communication quality.

In the last two decades, great efforts have been made to compensate nonlinearities in AEC and a large number of nonlinear AEC (NAEC) methods have been proposed. These methods may differ in what type of nonlinearities (static or dynamic) are involved. But a practically popular, common choice is to place a nonlinear preprocessor in front of the linear FIR filter, including Volterra filter based algorithms [1, 2] and polynomial-based approaches [3, 4, 5]. The normalized least mean squares (NLMS) [3] and recursive least squares (RLS) [4] approaches have been developed for the preprocessor's adaptation.

This paper considers a novel hybrid Taylor (polynomial) and Volterra filter for the pre-processor, and presents a more efficient NAEC implementation which jointly adapts the pre-processor and the FIR filter using a block RLS method and the frequency-domain LMS (FLMS) algorithm, respectively.

2. SIGNAL MODELS

Mobile phone audio systems are complicated and many parts can introduce nonlinear distortion. The dominant source of nonlinearity may vary from phone to phone. So there are a large number of signal models that have been considered in the literature, including

1) Static power filter [3, 5, 6],
2) Cascaded dynamic power filter [7],
3) Parallel dynamic power filter [8], and
4) Second-order Volterra filter [1].

Figure 1 depicts the structures of these models. They all have pros and cons. The static power filter is simple but cannot handle dynamic nonlinearities. Both the cascaded and parallel dynamic power filters may suffer from the system identifiability ambiguity issue, which means that they can be reduced. The second-order Volterra filter sacrifices the power of modeling high-order (higher than the 2nd order) static nonlinearities for mathematical tractability. Through a little bit of tweaking, we can reach a better balance point between simplicity and power of representation: we put a 2nd-order Volterra filter side-by-side with a higher (than 2nd) order Taylor (power) filter as shown in Fig. 2. This still rather simple structure can model not only static amplifier soft-clipping saturation but also dynamic loudspeaker nonlinearities. We refer to this new proposed model as the hybrid Taylor-Volterra (HTV) model for NAEC.

Following this HTV model, the output echo is mathematically expressed as

$$y(n) = \mathbf{h}^T \left[\mathbf{X}^{(1)}(n)\mathbf{g}_1 + \mathbf{X}^{(2)}(n)\mathbf{g}_2 + \mathbf{X}^{(3+)}(n)\mathbf{g}_{3+}\right] + v(n), \tag{1}$$

where n is the discrete time index, $(\cdot)^T$ denotes the transpose of a vector or a matrix,

$$\mathbf{h} \triangleq \left[\, h_0 \;\; h_1 \;\; \cdots \;\; h_{L-1} \,\right]^T,$$

L is the length of the adaptive filter that models the linear part of the echo path centered by the near-end surrounding sound field,

$$\mathbf{X}^{(1)}(n) \triangleq \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-M_1+1) \\ x(n-1) & x(n-2) & \cdots & x(n-M_1) \\ \vdots & \vdots & \ddots & \vdots \\ x(n-L+1) & x(n-L) & \cdots & x(n-L-M_1+2) \end{bmatrix},$$

$$\mathbf{g}_1 \triangleq \left[\, g_{1,0} \;\; g_{1,1} \;\; \cdots \;\; g_{1,M_1-1} \,\right]^T,$$

M_1 is the memory length of the first-order Volterra kernel,

$$\begin{aligned} \mathbf{X}^{(2)}(n) &\triangleq \left[\, \mathbf{x}^{(2)}(n) \;\; \mathbf{x}^{(2)}(n-1) \;\; \cdots \;\; \mathbf{x}^{(2)}(n-L+1) \,\right]^T, \\ \mathbf{x}^{(2)}(n) &\triangleq \big[\, x^2(n) \;\; \cdots \;\; x(n)x(n-M_2+1) \\ &\qquad x^2(n-1) \;\; \cdots \;\; x(n-1)x(n-M_2+1) \\ &\qquad \cdots \;\; x^2(n-M_2+1) \,\big]^T, \\ \mathbf{g}_2 &\triangleq \big[\, g_{2,(0,0)} \;\; \cdots \;\; g_{2,(0,M_2-1)} \;\; g_{2,(1,1)} \;\; \cdots \;\; g_{2,(1,M_2-1)} \\ &\qquad \cdots \;\; g_{2,(M_2-1,M_2-1)} \,\big]^T, \end{aligned}$$

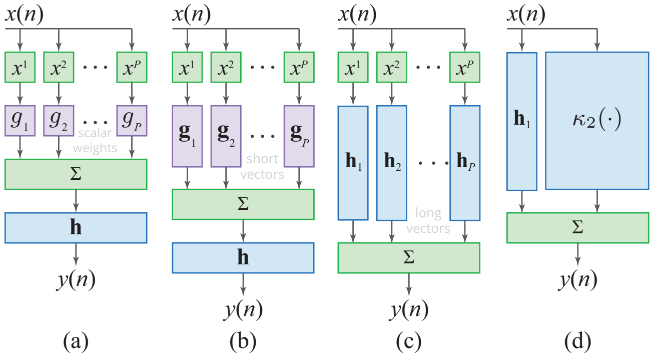

Fig. 1: Structures of the signal models that were considered in the literature for NAEC: (a) static power filter, (b) cascaded dynamic power filter, (c) parallel dynamic power filter, and (d) second-order Volterra filter.

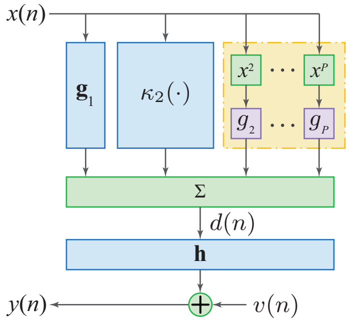

Fig. 2: Structure of the proposed hybrid Taylor-Volterra (HTV) model for NAEC.

M_2 is the memory length of the second-order Volterra kernel,

$$\mathbf{X}^{(3+)}(n) \triangleq \begin{bmatrix} x^3(n) & \cdots & x^P(n) \\ x^3(n-1) & \cdots & x^P(n-1) \\ \vdots & \cdots & \vdots \\ x^3(n-L+1) & \cdots & x^P(n-L+1) \end{bmatrix},$$

$$\mathbf{g}_{3+} \triangleq \begin{bmatrix} g_3 & g_4 & \cdots & g_P \end{bmatrix}^T,$$

P is the order of the polynomial, and $v(n)$ denotes the sum of local speech and additive noise.

Let's define

$$\mathbf{X}(n) \triangleq \begin{bmatrix} \mathbf{X}^{(1)}(n) & \mathbf{X}^{(2)}(n) & \mathbf{X}^{(3+)}(n) \end{bmatrix}, \quad (2)$$

$$\mathbf{g} \triangleq \begin{bmatrix} \mathbf{g}_1^T & \mathbf{g}_2^T & \mathbf{g}_{3+}^T \end{bmatrix}^T. \quad (3)$$

Then (1) is reduced to a more compact form

$$y(n) = \mathbf{h}^T \mathbf{X}(n)\mathbf{g} + v(n). \quad (4)$$

It is worth noting that if $M_1 = M_2 = 1$, then the HTV filter is simply a P-th order power filter. If $P = 2$, then it is a pure second-order Volterra filter. But they all share the same mathematical form of (4), differing only in how $\mathbf{X}(n)$ is constructed from the input $x(n)$ and how many coefficients $\mathbf{g}$ has.

3. CASCADED NLMS ADAPTIVE FILTERS

In the theory of nonlinear system identification, the widely used and studied Hammerstein model consists of a *static* nonlinear unit followed by a *dynamic* linear time-invariant (LTI) FIR filter. The HTV model is not a typical Hammerstein system since its input nonlinearity is dynamic and is represented by a 2nd-order Volterra filter. But the cascaded NLMS adaptive filtering algorithm that was developed for Hammerstein systems [3] can also work for the HTV model.

To begin, we define the HTV error signal at time n and the mean square error (MSE) function as follows

$$e(n) \triangleq y(n) - \hat{y}(n) = y(n) - \hat{\mathbf{h}}^T \mathbf{X}(n)\hat{\mathbf{g}}, \quad n = 0, 1, 2, \cdots \quad (5)$$

$$J \triangleq E\{e^2(n)\}, \quad (6)$$

where $E\{\cdot\}$ denotes the mathematical expectation. To minimize the MSE, the nonlinear and linear filter coefficients can be jointly adapted via the following NLMS updating rules:

$$e(n) = y(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{X}(n)\hat{\mathbf{g}}(n-1), \quad (7)$$

$$\hat{\mathbf{g}}(n) = \hat{\mathbf{g}}(n-1) + \frac{\mu_g \cdot e(n)\mathbf{X}^T(n)\hat{\mathbf{h}}(n-1)}{\hat{\mathbf{h}}^T(n-1)\mathbf{X}(n)\mathbf{X}^T(n)\hat{\mathbf{h}}(n-1) + \delta_g}, \quad (8)$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{\mu_h \cdot e(n)\mathbf{X}(n)\hat{\mathbf{g}}(n-1)}{\hat{\mathbf{g}}^T(n-1)\mathbf{X}^T(n)\mathbf{X}(n)\hat{\mathbf{g}}(n-1) + \delta_h}, \quad (9)$$

where μ_g and μ_h are step sizes, δ_g and δ_h are regularization factors.

4. CASCADED BLOCK RLS AND FLMS ADAPTIVE FILTERS

It is well known that the NLMS is simple, but slow in convergence particularly for colored speech inputs. It has been shown in [3] that a sophisticated step-size control is crucial to the performance of joint NLMS adaptation of a polynomial pre-processor and an FIR filter. The optimum step-size control depends on the assumed form of input nonlinearity and hence is difficult to implement. As a result, the NLMS is deemed practically ineffective for AECs.

In this paper, we suggest to use the RLS method for the adaptation of the nonlinear preprocessor coefficients $\hat{\mathbf{g}}(n)$ in (8), similar to what was proposed in [9]. The RLS has much faster convergence and much smaller final misalignment than the NLMS. Furthermore we intend to replace the NLMS adaptation of the FIR filter by the FLMS approach. The FLMS enjoys two main advantages: 1) low complexity thanks to the use of the Fast Fourier Transform (FFT), and 2) fast convergence due to the fact that the discrete Fourier transform (DFT) can approximately decorrelates the input signals and adaptation can be independently optimized in each frequency bin. So the FLMS adaptation is a popular choice in real-time linear AECs. Since the FLMS processes the input data block by block, we want to derive a block-based RLS adaptation scheme for the nonlinear pre-processor.

Given the data $\{x(i), y(i)\}$ and all the linear filter estimates $\hat{\mathbf{h}}(i-1)$ for $1 \leq i \leq n$, the RLS algorithm minimizes the sum of squared errors equipped with a leaky memory:

$$J_{\hat{\mathbf{g}}}(n) \triangleq \sum_{i=1}^{n} \lambda_g^{n-i} e^2(i) = \sum_{i=1}^{n} \lambda_g^{n-i} \left[y(i) - \hat{\mathbf{h}}^T(i-1)\mathbf{X}(i)\hat{\mathbf{g}} \right]^2$$
$$= \hat{\mathbf{g}}^T \mathbf{R}(n)\hat{\mathbf{g}} - 2\mathbf{r}^T(n)\hat{\mathbf{g}} + \sum_{i=1}^{n} \lambda_g^{n-i} y^2(i), \quad (10)$$

where $0 < \lambda_g < 1$ is an exponential forgetting factor, and

$$\mathbf{R}(n) \triangleq \sum_{i=1}^{n} \lambda_g^{n-i} \mathbf{X}^T(i)\hat{\mathbf{h}}(i-1)\hat{\mathbf{h}}^T(i-1)\mathbf{X}(i), \quad (11)$$

$$\mathbf{r}(n) \triangleq \sum_{i=1}^{n} \lambda_g^{n-i} y(i)\mathbf{X}^T(i)\hat{\mathbf{h}}(i-1). \quad (12)$$

Table 1: Joint adaptation of the HTV model using the block RLS and FLMS algorithms (pseudocode).

```
$L$: filter length,   $L_s$: block shift ($L_s \leq L$),   $2L$: FFT size
$\mu_h$: step size,   $\lambda_g, \lambda_h$: forgetting factors
$\mathbf{F}_{2L\times 2L}$: Fourier matrix
$\mathbf{W}^{01}_{L\times 2L} \triangleq \left[\begin{array}{cc} \mathbf{0}_{L\times L} & \mathbf{I}_{L\times L} \end{array}\right]$ and $\mathbf{W}^{10}_{L\times 2L} \triangleq \left[\begin{array}{cc} \mathbf{I}_{L\times L} & \mathbf{0}_{L\times L} \end{array}\right]$
Initialize $\hat{\mathbf{g}}(0), \hat{\mathbf{h}}(0), \mathbf{R}(0), \mathbf{r}(0), \mathbf{S}(0)$
$\hat{\mathbf{h}}_{f,2L} = \mathbf{F}_{2L\times 2L} \left[\begin{array}{cc} \hat{\mathbf{h}}^T(0) & \mathbf{0}^T_{L\times 1} \end{array}\right]^T$
For $m = 1, 2, \cdots$   (block iteration)
  For $i = 1, 2, \cdots, L_s$   (iterate over all samples of the block)
        $n = (m-1)L_s + i$;
      $\mathbf{R}(n) = \lambda_g \mathbf{R}(n-1) + \mathbf{X}^T(n)\hat{\mathbf{h}}(m-1)\hat{\mathbf{h}}^T(m-1)\mathbf{X}(n)$;
      $\mathbf{r}(n) = \lambda_g \mathbf{r}(n-1) + y(n)\mathbf{X}^T(n)\hat{\mathbf{h}}(m-1)$;
  end
  $\hat{\mathbf{g}}(m) = \mathbf{R}^{-1}(n)\mathbf{r}(n)$
  Use $\hat{\mathbf{g}}(m)$ to pre-process $\mathbf{X}(mL_s)$ and $\mathbf{X}((m-1)L_s)$ and get
    $\mathbf{d}_{2L}(m) \triangleq \left[\begin{array}{cc} \mathbf{X}^T(mL_s) & \mathbf{X}^T((m-1)L_s) \end{array}\right]^T \hat{\mathbf{g}}(m)$
  Construct $\mathbf{y}_L(m) \triangleq \left[\begin{array}{ccc} y(mL_s - L + 1) & \cdots & y(mL_s) \end{array}\right]^T$
  Compute
    $\mathbf{D}(m) = \mathrm{diag}\{\mathbf{F}_{2L\times 2L}\mathbf{d}_{2L}(m)\}$
    $\mathbf{S}(m) = \lambda_h \mathbf{S}(m-1) + (1-\lambda_h)\mathbf{D}^*(m)\mathbf{D}(m)$
    $\mathbf{e}_L(m) = \mathbf{y}_L(m) - \mathbf{W}^{01}_{L\times 2L}\mathbf{F}^{-1}_{2L\times 2L}\mathbf{D}(m)\hat{\mathbf{h}}_{f,2L}(m-1)$
  Extract the last $L_s$ samples from $\mathbf{e}_L(m)$ to get
    $\mathbf{e}(m) \triangleq \left[\begin{array}{ccc} e(mL_s - L_s + 1) & \cdots & e(mL_s) \end{array}\right]^T$
  Update
    $\mathbf{e}_{f,2L}(m) = \mathbf{F}_{2L\times 2L} \left[\begin{array}{cc} \mathbf{0}^T_{L\times 1} & \mathbf{e}^T_L(m) \end{array}\right]^T$
    $\hat{\mathbf{h}}_{f,2L}(m) = \hat{\mathbf{h}}_{f,2L}(m-1) + \mu_h(1-\lambda_h)\times$
                 $\left[\mathbf{S}(m) + \delta_h \mathbf{I}_{2L\times 2L}\right]^{-1} \mathbf{D}^*(m)\mathbf{e}_{f,2L}(m)$
    $\hat{\mathbf{h}}(m) = \mathbf{W}^{10}_{L\times 2L}\mathbf{F}^{-1}_{2L\times 2L}\hat{\mathbf{h}}_{f,2L}(m)$
end
```

Differentiating (10) and equating the result to zero yields

$$\mathbf{R}(n)\hat{\mathbf{g}} = \mathbf{r}(n). \tag{13}$$

Solving this normal equation (13) for $\hat{\mathbf{g}}$ leads to the RLS estimate of the pre-processor coefficients at time n

$$\hat{\mathbf{g}}(n) = \mathbf{R}^{-1}(n)\mathbf{r}(n). \tag{14}$$

In a block-processing framework, the nonlinear pre-processor coefficients will be updated only once at the end of each block and so the matrix inversion in (14) needs to be computed just once for each data block. But $\mathbf{R}(n)$ and $\mathbf{r}(n)$ should still be recursively updated sample by sample. For brevity and clarity, the joint RLS and FLMS adaptations of the HTV model filters are summarized in Table 1. Note that here an unconstrained version of the FLMS algorithm [10] is incorporated.

5. SIMULATIONS

Here we consider a synthesized HTV nonlinear echo system. The input nonlinearity is a static soft-clipping function given by

$$f(x) = \frac{\gamma x}{\sqrt{x^2 + \gamma^2}}, \tag{15}$$

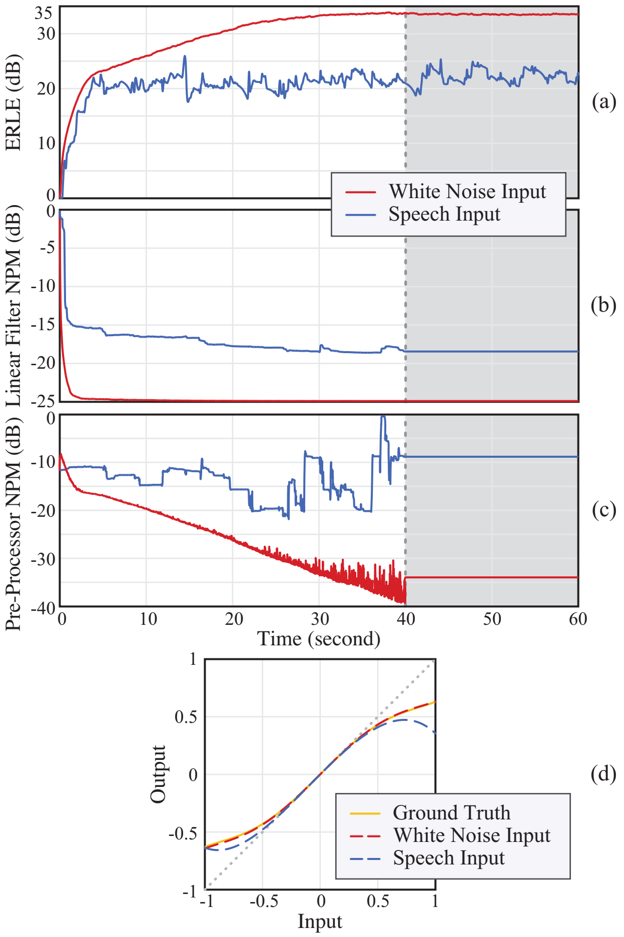

Fig. 3: Comparison of the performance of the joint NLMS adaptation of the HTV nonlinear pre-processor and FLMS adaptation of the linear FIR filter on white and speech inputs in terms of (a) ERLE, (b) linear FIR NPM, (c) nonlinear pre-processor NPM, and (d) estimated input nonlinearity.

where γ is the maximum output amplitude as x goes to infinity. Such a nonlinearity is memoryless such that $M_1 = M_2 = 1$. The acoustic impulse response is extracted from the AIR multichannel impulse response response database [11]. The 60 dB reverberation time is 160 ms. The sampling rate is 16 kHz and we truncate the impulse response to $L_t = 1024$ taps. The input signal to the echo system can be either a white random signal or clean speech. Without loss of generality, the speech is taken from a continuous speech story corpora and the speaker is male. At the output of the echo system, only white noise is added at a signal-to-noise ratio (SNR) of 35 dB, implying no local speech.

In the simulations, we keep the length of data to 60 s. For the first 40 s, the simulated NAEC algorithms adapt as they are supposed to do. But we freeze them afterwards, which simulates the case of double talks and helps check whether the adaptive algorithms may have overfitted the data.

The performance of NAEC algorithms is assessed by three measures:

1) Echo return loss enhancement (ERLE):

$$\mathrm{ERLE} \triangleq 10\log_{10}\frac{\langle y^2(n)\rangle_t}{\langle e^2(n)\rangle_t}, \tag{16}$$

where $\langle\cdot\rangle_t$ denotes the average of data over a period of t and $e(n)$ is the NAEC residual signal. In our simulations, t is set to 200 ms and the ERLE analysis window is shifted once every 10 ms.

2) Normalized projection misalignment (NPM) of the linear FIR filter (which can absorb a gain ambiguity) [12]:

$$\text{NPM}\left(\hat{\mathbf{h}}, \mathbf{h}_\text{t}\right) \triangleq 20\log_{10}\frac{\left\|\mathbf{h}_\text{t} - \alpha\hat{\mathbf{h}}_\text{z}\right\|}{\|\mathbf{h}_\text{t}\|}, \tag{17}$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector, $\alpha \triangleq \left(\mathbf{h}_\text{t}^T\hat{\mathbf{h}}_\text{z}\right) / \left(\hat{\mathbf{h}}_\text{z}^T\hat{\mathbf{h}}_\text{z}\right)$, and $\hat{\mathbf{h}}_\text{z} \triangleq \left[\hat{\mathbf{h}}^T \;\; \mathbf{0}^T_{(L_\text{t}-L)\times 1}\right]^T$ is a zero-padded vector of $\hat{\mathbf{h}}$ that matches the length of $\mathbf{h}_\text{t}$.

3) NPM of the static nonlinear pre-processor [13]:

$$\text{NPM}\left\{\hat{f}(x|\hat{\mathbf{g}}), f_\text{t}(x)\right\} \triangleq 10\log_{10}\frac{\displaystyle\int_a^b \left[f_\text{t}(x) - \beta\hat{f}(x|\hat{\mathbf{g}})\right]^2 dx}{\displaystyle\int_a^b f_\text{t}^2(x)dx}, \tag{18}$$

where $f_\text{t}(x)$ and $\hat{f}(x|\hat{\mathbf{g}})$ denote respectively the true and estimated pre-processor nonlinearity, (a, b) is the interested range of inputs, and

$$\beta \triangleq \frac{\int_a^b f_\text{t}(x)\hat{f}(x|\hat{\mathbf{g}})dx}{\int_a^b \hat{f}^2(x|\hat{\mathbf{g}})dx}.$$

The first experiment compares the performance of NLMS adaptation of the HTV nonlinear pre-processor on white and speech inputs. The linear FIR filter is adapted by using the FLMS algorithm. The results are plotted in Fig. 3. Simulation parameters are: soft clipping $\gamma = 0.75$, $L = 768$, $\mu_g = 5\times10^{-3}$, $\mu_h = 0.3$, $P = 5$, $L_\text{s} = 128$.

When the input signal is white noise, the NLMS adaptation of the pre-processor can effectively identify the input nonlinearity while the FLMS adaptation of the linear FIR filter yields very low final misalignment. But when the input signal becomes speech, the algorithm does not work as well: it is clear from Fig. 3(d) that the estimated input nonlinearity is significantly different from the ground truth. Note that the soft-clipping input nonlinearity merely affect data with large magnitudes. When the input is white noise which has a uniform distribution over $(-1, 1)$, the soft-clipping can be fully excited and so it is easy to identify. But speech is generally a super-Gaussian random process and only occasionally has very large samples (not even to mention frequent silent periods in speech signals). So the input nonlinearity is not fully excited and the identification problem becomes difficult for the NLMS adaptive algorithm.

The second experiment intends to validate the performance of the proposed block RLS+FLMS algorithm. A comparison is made against the NLMS+FLMS algorithm. The input is speech, soft clipping $\gamma = 0.75$, $L = 768$, $P = 5$, $L_\text{s} = 128$, $\mu_g = 5\times10^{-4}$ and $\mu_h = 0.3$ for the NLMS+FLMS, $\lambda g = 0.999995$ and $\mu_h = 0.3$ for the RLS+FLMS implementation. The results are collected in Fig. 4.

The block RLS converges much faster than the NLMS for identifying the nonlinear pre-processor. After convergence, the input nonlinearity estimated by the RLS is very close to the ground truth. Note that what we are interested here is the nonlinear function only over the range of inputs. For the NLMS, we have chosen a much smaller step size $\mu_g = 5\times10^{-4}$ (in comparison with 5×10^{-3} in the previous experiment) but unfortunately a satisfactory convergence has not yet been reached at 40 s.

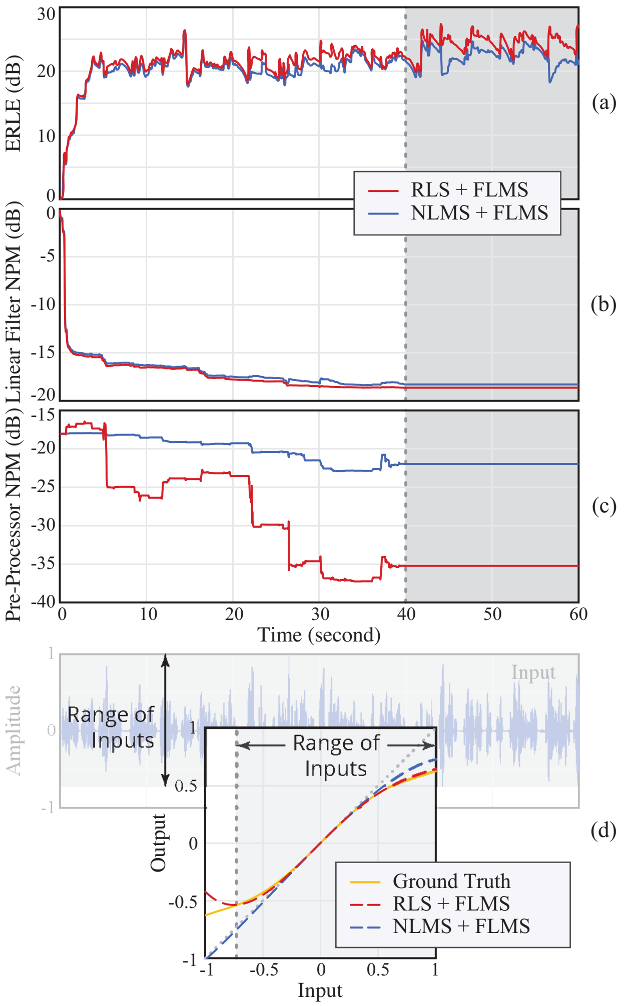

Fig. 4: Comparison of the joint RLS+FLMS and joint NLMS+FLMS adaptations of the HTV model for AEC in terms of (a) ERLE, (b) linear FIR NPM, (c) nonlinear pre-processor NPM, and (d) estimated input nonlinearity.

The presence of a mild soft-clipping input nonlinearity does not seem to significantly affect the performance of the linear filter FLMS adaptation: the accuracy of the estimated input nonlinearities by the NLMS and block RLS algorithms makes no meaningful difference in linear filter NPM [see Fig. 4(b)]. But including a nonlinear pre-processor in the echo-path model and being able to accurately identify it leads to higher ERLE levels, particularly after adaptation is frozen [see Fig. 4(a)].

6. CONCLUSIONS

This paper studied the NAEC problem and presented a practically efficient implementation. A hybrid Taylor-Volterra model was proposed for parameterizing the echo path input nonlinearities on mobile devices. It makes a better balance between simplicity and power of representation. A block RLS algorithm was developed for adaptation of the nonlinear pre-processor and the linear FIR filter was jointed adapted by the unconstrained FLMS algorithm. Simulation results showed that the presented NAEC implementation is fast in convergence and is free from data overfitting.

7. REFERENCES

[1] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters," in *Proc. IEEE ICASSP*, 1999, vol. 2, pp. 877–880.

[2] A. Guerin, G. Faucon, and R. Le Bouquin-Jeannes, "Nonlinear acoustic echo cancellation based on Volterra filters," *IEEE Trans. Speech Audio Proc.*, vol. 11, no. 6, pp. 672–683, Nov. 2003.

[3] A. Stenger, W. Kellermann, and R. Rabenstein, "Adaptation of acoustic echo cancellers incorporating a memoryless nonlinearity," in *Proc. IEEE Workshop on Acoustic Echo and Noise Control (IWAENC)*, 1999, pp. 168–171.

[4] A. Stenger and W. Kellermann, "RLS-adapted polynomial for nonlinear acoustic echo cancelling," in *Proc. EUSIPCO*, 2000, pp. 1867–1870.

[5] K. Shi, X. Ma, and G. Zhou, "A channel shortening approach for nonlinear acoustic echo cancellation," in *Proc. IEEE Workshop on Statistical Signal Processing*, 2007, pp. 351–354.

[6] M. Z. Ikram, "Non-linear acoustic echo cancellation using cascaded Kalman filtering," in *Proc. IEEE ICASSP*, 2014, pp. 1320–1324.

[7] M. I. Mossi, C. Yemdji, N. W. D. Evans, C. Beaugeant, and P. Degry, "Robust and low-cost cascaded non-linear acoustic echo cancellation," in *Proc. IEEE ICASSP*, 2011, pp. 89–92.

[8] F. Kuech, A. Mitnacht, and W. Kellermann, "Nonlinear acoustic echo cancellation using adaptive orthogonalized power filters," in *Proc. IEEE ICASSP*, 2005, vol. 3, pp. 105–108.

[9] A. Stenger and W. Kellermann, "Adaptation of a memoryless preprocessor for nonlinear acoustic echo cancelling," *Signal Processing (Elsevier)*, vol. 80, no. 9, pp. 1747–1760, Sept. 2000.

[10] D. Mansour and A. H. Gray Jr., "Unconstrained frequency-domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 30, no. 5, pp. 726–734, Oct. 1982.

[11] Rwth Aachen University (Germany) and Bar-Ilan University (Israel), "Multichannel impulse response database," `http://http://www.ind.rwth-aachen.de/en/research/tools-downloads/multichannel-impulse-response-database/`, 2014, [Online; Latest Accessed 16-Sep-2015].

[12] D. R. Morgan, J. Benesty, and M. M. Sondhi, "On the evaluation of estimated impulse responses," *IEEE Signal Process. Lett.*, vol. 5, pp. 174–176, July 1998.

[13] G. Enzner, T. C. Kranemann, and P. Thüne, "Evaluation of estimated Hammerstein models via normalized projection misalignment of linear and nonlinear subsystems," in *Proc. IEEE ICASSP*, 2016, pp. 4234–4238.