Hybrid Filterbank ADCs With Blind Filterbank Estimation

by Damián Edgardo Marelli, Kaushik Mahata, and Minyue Fu

This is an author-prepared version of the article, reprinted from IEEE Transactions on Circuits and Systems, 1: Regular Papers, Vol. 58, Issue 10, p. 2446 to 2457.

http://dx.doi.org/10.1109/TCSI.2011.2123750

Hybrid Filterbank ADCs with Blind Filterbank Estimation

Damián Marelli[1], Kaushik Mahata[1] and Minyue Fu[2], *Fellow, IEEE*

***Abstract*—The hybrid filterbank architecture permits implementing accurate, high speed analog-to-digital converters. However, its design requires an accurate knowledge of the analog filterbank parameters, which is difficult to have due to the non-stationary nature of these parameters. This paper proposes a blind estimation method for the analog filterbank parameters, which is able to cope with non-stationary input signals. This is achieved by using the notion of averaged input spectrum. The estimated parameters are used to reconstruct the samples in a least mean squares (LMS) sense. The proposed LMS design generalizes existing approaches by dropping the bandlimited assumption on the input signal. Instead, it assumes that the input has an arbitrary power spectrum which is adaptively estimated. Numerical experiments are presented showing the good performance of the blind estimation stage, and the clear advantage of the proposed LMS design.**

I. Introduction

A high speed analog-to-digital converter (ADC) can be realized by using the so-called time-interleaved ADC (TI-ADC) architecture [1]. It consists of using a number of parallel ADCs having the same sampling rate but different sampling phases, as if they were a single ADC operating at a higher sampling rate. In spite of its conceptual simplicity, the design of a TI-ADC needs to account for mismatches between different channel ADCs [2], [3]. A drawback of this technique is its extreme sensitivity to timing mismatches [4], [5]. To overcome this limitation, the hybrid filterbank ADC (HFB-ADC) architecture was proposed in [4]. This technique uses a continuous-time analysis filterbank to split the input signal into different frequency bands, each of which is assigned to a different ADC. In contrast to the TI-ADC architecture, all the ADCs in a HFB-ADC are synchronously sampled. A discrete-time synthesis filterbank is then used to reconstruct the required samples. A variant of this technique carries out the frequency band splitting using lowpass filtering and frequency translation, to relax the design constraints on sample-and-hold devices at high-frequencies [6]. This simplification in design has motivated HFB-ADCs in challenging applications with wideband and bandpass signals [7], as well as efficient architectures for the digital synthesis bank [8].

The design of the discrete-time synthesis filterbank requires the knowledge of the frequency response of the analysis filters. It is often unrealistic to assume that this is known in advance accurately enough, since analog circuits are subject to imperfections, e.g., deviations from nominal values, aging, temperature drifts, etc. An approach to deal with this uncertainty is to use a reference input signal to estimate the analog filterbank parameters [9]. A similar approach is used in [10], [11], [12], where instead of estimating the analog filterbank, the digital synthesis bank is directly tuned, via an adaptive filtering technique, to minimize the reconstruction error. However, as pointed out in [13], a blind estimation technique (i.e., one carrying out the estimation without the knowledge of the input signal) is preferred, since it does not interfere with the ADC operation, and is able to track analog parameter drifts during the ADC operation. Towards this goal, in this paper we propose a blind method for estimating the analog filterbank parameters. The proposed method is adaptive (i.e., on-line), so it can run continuously in parallel with the ADC operation.

Once the analysis filterbank parameters are known, the discrete-time synthesis filterbank can be designed to reconstruct the desired samples. An approach for doing so relies on the assumption that the input signal is bandlimited [5], [13]. Under this assumption, these methods are able to achieve perfect reconstruction if the impulse response of the synthesis filterbank can be arbitrarily long. An arguable point of this approach is that the bandlimited assumption might not be realistic in many applications. One way to address this issue is to assume that the input signal has finite energy, and design the compensation in a minmax sense [14], [15]. In this paper we use a different criterion. We assume that the input signal is a random process and we carry out a compensation in a statistically optimal (least mean squares (LMS)) sense. A similar approach was proposed by the authors in [16], [17], to design a compensation for timing mismatch in TI-ADCs. The proposed method permits designing the synthesis filterbank so that the reconstructed samples match those that would be obtained if the input signal was passed through a prescribed anti-alias filter before sampling. This is particularly important in view of our non-bandlimited assumption on the input signal. We show that the methods in [5], [13], derived under a bandlimited assumption, are particular cases of the proposed method.

The proposed synthesis filterbank design method requires the knowledge of the power spectrum of the input signal. Since it is impractical to assume that this is known in advance, we propose a real-time method for estimating it. Nevertheless,

The authors are with the School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia. Emails: Damian.Marelli@newcastle.edu.au (Damián Marelli). Kaushik.Mahata@newcastle.edu.au (Kaushik Mahata).

Minyue Fu is with the School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia and with the Department of Control Science and Engineering, Zhejiang University, China. Email: Minyue.Fu@newcastle.edu.au (Minyue Fu).

numerical experiments suggest that an accurate knowledge of the input power spectrum is not necessary, since the reconstruction error is to some extent insensitive to input spectrum estimation errors.

Apart from being conceptually intuitive, an advantage of the TI-ADC over the HFB-ADC architecture is that, when there are neither timing nor gain mismatches, the desired samples are readily available, without the need for digital processing. However, when these mismatches are unavoidable, compensating for them is a non-trivial problem, which has been addressed in a number of works [18], [19], [20], [17]. Additionally, as in the case of HBF-ADCs, these mismatches are subject to drifts, and they need to be estimated, either off-line [21], [22] or on-line [23], [24], [25]. Notice that a TI-ADC is a particular case of a HFB-ADC, where the analysis filters are chosen as time delays. Hence, while the adaptive methods proposed in this work are intended for estimating the analog parameters and input spectrum in a HFB-ADC, these methods can also be used for estimating timing and gain mismatches, as well as the input spectrum, in a TI-ADC. Finally, we point out that, in both architectures, the complexity of the estimation task is dominant over that of the sample reconstruction task.

The rest of the paper is organized as follows. We give an overview of hybrid filterbank ADCs in Section II. In Section III we describe the proposed adaptive blind method for estimating the analysis filterbank parameters. In Section IV we describe the proposed synthesis filterbank design method, as well as the adaptive method for estimating the input power spectrum. Also in Subsection IV-C, we show that the design method derived under a bandlimited assumption is a particular case of the proposed method. Finally, some simulation results are presented in Section VI, and concluding remarks are given in Section VII. This paper is partly based on the work reported in the conference paper [26].

Discrete-time functions (i.e., signals and impulse responses) are denoted using bold letters and their continuous-time counterparts using non-bold letters. Also, time-domain functions are denoted in lowercase and their frequency-domain counterparts in uppercase. The convolution between the continuous-time signals $a(t)$ and $b(t)$ is denoted by $(a*b)(t)$. The adjoint $a^*(t)$ of $a(t)$ is defined by $a^*(t) = a^T(-t)$, and $A^*(s)$ denotes the two-sided Laplace transform of $a^*(t)$ (with s being the Laplace variable). The same notation holds for convolution and adjoint of discrete-time functions.

II. Hybrid Filterbank Analog-to-Digital Converters

The HFB-ADC scheme is depicted in Figure 1. The continuous-time signal $x(t)$ is split into M signals using an array of analog filters with transfer functions $H(s) = [H_1(s), \cdots, H_M(s)]^T$, whose outputs are then sampled at the rate of $1/DT$. In this way, the discrete-time signals $\mathbf{x}(k) = [\mathbf{x}_1(k), \cdots, \mathbf{x}_M(k)]^T$ are generated. The idea is to process $\mathbf{x}(k)$ to generate an estimate $\hat{\mathbf{y}}(k)$ of the samples $\mathbf{y}(k) = (g*x)(kT)$, collected after the anti-alias filter $G(s)$. This is typically done by upsampling the signals $\mathbf{x}(k)$ by a factor of D (i.e., $D-1$ zero valued samples are added between every two samples), then filtering each component using the array of discrete-time filters $\mathbf{F}(z) = [\mathbf{F}_1(z), \cdots, \mathbf{F}_M(z)]^T$, and finally adding together all the resulting signals.

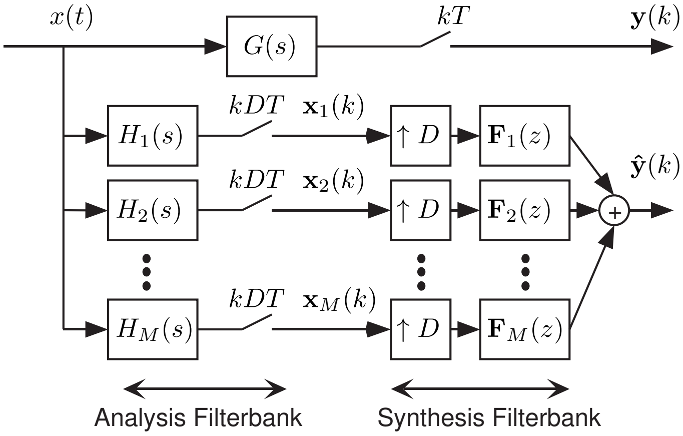

Figure 1. Slightly generalized HFB-ADC scheme considered in this work.

As mentioned in Section I, the design of a HBF-ADC comprises of two stages. The first is to estimate the continuous-time filters $H(s)$ using the samples $\mathbf{x}(k)$, and the second is to use this estimate to design the discrete-time filters $\mathbf{F}(z)$ for reconstruction. We will address these two problems in Sections III and IV below.

The scheme considered in Figure 1 is somewhat more general than the one considered in [5], [13], in that it permits placing an anti-alias filter $G(s)$ before generating the samples $\mathbf{y}(k)$ to be reconstructed (in the results below, the anti-alias filter can be removed by choosing $G(s) = 1$), as well as using oversampling (i.e., $D < M$). Notice that, when using oversampling, while the average rate of the samples $\mathbf{x}_m(k)$, $m = 1, \cdots, M$ is $M/(DT) > 1/T$, the samples $\hat{\mathbf{y}}(k)$ are still reconstructed at the desired rate $1/T$. Therefore, this form of oversampling differs from the usual form in which the samples are reconstructed at a rate higher than the desired one. When $M > D$, the choice of filters $\mathbf{F}(z)$ which produce some given samples $\hat{\mathbf{y}}(k)$ is not unique. Hence, oversampling adds flexibility in the design of $\mathbf{F}(z)$, at the expense of a higher average sampling rate.

III. Adaptive Blind Estimation of the Analysis Filterbank

In this section we propose an adaptive blind algorithm for estimating $H(s)$. We assume that the input signal is a random process with possibly non-stationary statistics. Our algorithm is derived to deal with the case when $H(s)$ is unknown and slowly time-varying.

A. Estimation criterion

Since $x(t)$ is not necessarily stationary, we define its (time-varying) autocorrelation by

$$r(t,\tau) = \mathcal{E}\{x(t+\tau)x(t)\}. \quad (1)$$

We also define its averaged autocorrelation up to time N by

$$r_N(\tau) = \frac{1}{\gamma(N)} \sum_{n=1}^{N} \lambda^{N-n} r(nDT, \tau), \tag{2}$$

where the *forgetting factor* λ is used to assign less weights to older measurements (to count for the slow time-varying nature of $H(s)$), and the scaling constant $\gamma(N) = \sum_{n=0}^{N-1} \lambda^n$ is used so that $r_N(\tau) = r(0, \tau)$ in the stationary case. Finally, the averaged autocorrelation of the samples is defined by

$$\mathbf{r}_N(k) \quad = \quad \frac{1}{\gamma(N)} \sum_{n=1}^{N} \lambda^{N-n} \mathcal{E}\{\mathbf{x}(n+k)\mathbf{x}^T(n)\}. \tag{3}$$

We then have the following result:

Lemma 1. *If $x(t)$ has uniformly bounded second moments (i.e., there exists $C > 0$ such that $\mathcal{E}\{x^2(t)\} < C$, for all $t \in \mathbb{R}$), $\int_{-\infty}^{\infty} |h_m(t)| dt < \infty$, for all $m = 1, \cdots, M$, and there exist functions $\tilde{r} \in L_1(\mathbb{R})$ (i.e., $\int_{-\infty}^{\infty} |\tilde{r}(\tau)| d\tau < \infty$) and $\kappa : \mathbb{R} \to \mathbb{R}^+$ ($\mathbb{R}^+$ denotes the set of positive real numbers) such that, for all $t, \tau, \delta \in \mathbb{R}$,*

$$|r(t, \tau) - r(t + \delta, \tau)| \le \kappa(\delta) \tilde{r}(\tau), \tag{4}$$

then

$$\mathbf{r}_N(k) \quad = \quad \mathcal{L}^{-1} \left\{ H(s) \Phi_N(s) H^*(s) \right\} (kDT) + \varepsilon(kDT), \tag{5}$$

where the averaged input power spectrum $\Phi_N(s) = \mathcal{L}\{r_N(\tau)\}$ is the two-sided Laplace transform of the averaged input autocorrelation $r_N(\tau)$. Also, for all $t \in \mathbb{R}$,

$$|\varepsilon(t)| \le (|h| * \tilde{r} * |\kappa \circ h|^*)(t), \tag{6}$$

where, $|a|$ denotes the vector/matrix formed by the absolute values of each of the entries of a and $\circ$ denotes pointwise multiplication (i.e., $\kappa \circ h(t) = \kappa(t) h(t)$).

Proof: See the Appendix. ■

Remark 2. The function $\kappa(\delta)$ in (4) states a bound on the rate of change of the autocorrelation of $x(t)$. Hence, for small values of δ where these statistics remain approximately constant, we have that $\kappa(\delta) \simeq 0$. If this condition holds within a time interval longer than the settling-time of the impulse response $h(t)$, then $\kappa(t)h(t) \simeq 0$. Hence, under this mild assumption, $\varepsilon(t) \simeq 0$ and therefore (5) becomes

$$\mathbf{r}_N(k) \simeq \mathcal{L}^{-1} \left\{ H(s) \Phi_N(s) H^*(s) \right\} (kDT). \tag{7}$$

Now, define the sample-average time-varying correlation by

$$\begin{aligned} \overline{\mathbf{r}}_N(k) \quad &= \quad \frac{1}{\gamma(N)} \sum_{n=1}^{N} \lambda^{N-n} \mathbf{x}(n+k) \mathbf{x}^T(n) \\ &= \quad \overline{\mathbf{r}}_{N-1}(k) + \frac{1}{\gamma(N)} \left(\mathbf{x}(N+k)\mathbf{x}^T(N) - \overline{\mathbf{r}}_{N-1}(k) \right) \end{aligned} \tag{8}$$

For a given N, we can approximate $\Phi_N(s)$ by a linear expansion $\Phi(s, \beta)$ as follows:

$$\Phi_N(s) \simeq \Phi(s, \beta) = \sum_{i=1}^{I} [\beta]_i E_i(s) \tag{9}$$

where $[\beta]_i$ denotes the i-th entry of the vector $\beta \in \mathbb{R}^I$ of expansion coefficients of $\Phi_N(s)$ on the basis $E_i(s)$, $i = 1, \cdots, I$. This approximation is realistic since any function can be approximated with an arbitrary accuracy by a linear expansion with sufficiently large number of basis elements. Since the orders of the analysis filterbank filters $H_m(s)$, $m = 1, \cdots, M$ are known, we can write a parametric version $H(s, \theta)$ of $H(s)$, where $\theta = [\theta_1, \cdots, \theta_P]$ denotes the vector of numerator and denominator coefficients of the filters $H_m(s)$, $m = 1, \cdots, M$. For a given $\overline{\mathbf{r}}_N(k)$, we can compute an estimate β_N of β up to time N as follows:

$$\beta_N(\theta) = W^{-1}(\theta) v_N(\theta), \tag{10}$$

where the the entries $[W(\theta)]_{i,j}$, $i, j = 1, \cdots, I$ and $[v_N(\theta)]_i$, $i, = 1, \cdots, I$ of the matrix $W(\theta) \in \mathbb{R}^{I \times I}$ and the vector $v_N(\theta) \in \mathbb{R}^I$, respectively, are defined by

$$\begin{aligned} [W(\theta)]_{i,j} \quad &= \quad \langle \downarrow_{DT} \{H(s,\theta) E_j(s) H^*(s,\theta)\}, \\ &\qquad \downarrow_{DT} \{H(s,\theta) E_i(s) H^*(s,\theta)\} \rangle, \qquad (11) \\ [v_N(\theta)]_i \quad &= \quad \langle \overline{\mathbf{r}}_N, \downarrow_{DT} \{H(s,\theta) E_i(s) H^*(s,\theta)\} \rangle, (12) \end{aligned}$$

with $\downarrow_{DT} \{A(s)\}(k) = \mathcal{L}^{-1} \{A(s)\} (kDT)$, for any matrix transfer function $A(s)$, and $\langle \mathbf{a}, \mathbf{b} \rangle = \mathrm{Tr}\{\mathbf{a} * \mathbf{b}^*\}(0)$ ($\mathrm{Tr}\{\cdot\}$ denotes the trace operation), for any matrix discrete-time impulse responses $\mathbf{a}(k)$ and $\mathbf{b}(k)$. We can hence define, using (7), a parametric time-varying correlation by

$$\mathbf{r}_N(k, \theta) = \downarrow_{DT} \{H(s,\theta) \Phi(s, \beta_N(\theta)) H^*(s,\theta)\} (k). \tag{13}$$

Then, the parameters θ_N up to time N can be estimated by solving the following minimization problem:

$$\begin{aligned} \theta_N \quad &= \quad \arg\min_{\tilde{\theta}} V_N(\tilde{\theta}) \qquad (14) \\ V_N(\theta) \quad &= \quad \sum_{k=0}^{K-1} \|\overline{\mathbf{r}_N}(k) - \mathbf{r}_N(k, \theta)\|_2^2 \qquad (15) \end{aligned}$$

where, for a matrix $A = [A_{i,j}]_{i,j=1}^{M}$, the norm $\|\cdot\|_2$ is defined by $\|A\|_2 = \sum_{i,j=1}^{M} |A_{i,j}|^2$.

B. Adaptive optimization algorithm

For a fixed N, the minimization problem (14)-(15) can be solved using a quasi-Newton method. These are iterative algorithms which use the parameters $\theta_{N,i}$ estimated at the i-th iteration in the following updating formula:

$$\theta_{N,i+1} = \theta_{N,i} - \mu_{N,i} T_{N,i} g_{N,i}, \tag{16}$$

where the scalar $\mu_{N,i}$ denotes the step-size at iteration i, the vector $g_{N,i}$ denotes the gradient of $V_N(\theta)$ at $\theta_{N,i}$, and the matrix $T_{N,i}$ denotes an approximation of the inverse of the Hessian of $V_N(\theta)$ at $\theta_{N,i}$. Following ideas from discrete-time system identification [27, Section 11.4], we can obtain an adaptive algorithm by carrying out one iteration of (16) for each new available sample. Hence, using the notation θ_N, μ_N, T_N and g_N for the sequence of values so obtained, we have that

$$\theta_{N+1} = \theta_N - \mu_N T_N g_N, \tag{17}$$

The p-th component $[g_N]_p$ of the gradient g_N is given by

$$[g_N]_p = -2\sum_{k=0}^{K-1}\left\langle \bar{\mathbf{r}}_N(k) - \mathbf{r}_N(k,\theta_N), \frac{\partial}{\partial\theta_p}\mathbf{r}_N(k,\theta_N)\right\rangle$$

where for matrices $A = [A_{i,j}]_{i,j=1}^M$ and $B = [B_{i,j}]_{i,j=1}^M$, the inner product $\langle\cdot,\cdot\rangle$ is defined by $\langle A, B\rangle = \sum_{i,j=1}^M A_{i,j}B_{i,j}$. The derivatives of $\mathbf{r}_N(k,\theta)$ with respect to the components θ_p of θ are given by

$$\begin{aligned}\frac{\partial}{\partial\theta_p}\mathbf{r}_N(k,\theta) &= \sum_{i=1}^{I}[\beta_N(\theta)]_i\mathbf{d}_{p,i}(k) \\ &+ \sum_{i=1}^{I}\frac{\partial}{\partial\theta_p}[\beta_N(\theta)]_i\mathbf{c}_i(k), \qquad (18)\end{aligned}$$

where

$$\begin{aligned}\mathbf{c}_i(k) &= \downarrow_{DT}\{H(s,\theta)E_i(s)H^*(s,\theta)\}(k) \qquad (19)\\ \mathbf{d}_{p,,i}(k) &= \downarrow_{DT}\left\{\frac{\partial}{\partial\theta_p}H(s,\theta)E_i(s)H^*(s,\theta)+\right.\\ &+ \left. H(s,\theta)E_i(s)\frac{\partial}{\partial\theta_p}H^*(s,\theta)\right\}(k). \qquad (20)\end{aligned}$$

Also, $\frac{\partial}{\partial\theta_p}\beta_N(\theta) = \left[\frac{\partial}{\partial\theta_p}[\beta_N(\theta)]_1,\cdots,\frac{\partial}{\partial\theta_p}[\beta_N(\theta)]_I\right]^T$, is computed by

$$\begin{aligned}\frac{\partial}{\partial\theta_p}\beta_N(\theta) &= W(\theta)^{-1}\frac{\partial v(\theta)}{\partial\theta_p} - \\ &- W(\theta)^{-1}\frac{\partial W(\theta)}{\partial\theta_p}W(\theta)^{-1}v(\theta), \qquad (21)\end{aligned}$$

with

$$\begin{aligned}\left[\frac{\partial v(\theta)}{\partial\theta_p}\right]_i &= \langle\bar{\mathbf{r}}_N, \mathbf{d}_{p,i}\rangle, \qquad (22)\\ \left[\frac{\partial W(\theta)}{\partial\theta_p}\right]_{ij} &= \langle\mathbf{d}_{p,j}, \mathbf{c}_i\rangle + \langle\mathbf{c}_j, \mathbf{d}_{p,i}\rangle. \qquad (23)\end{aligned}$$

To compute T_N we use the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula [28], which is initialized by $T_1 = I$ and proceeds as follows:

$$\begin{aligned}T_{N+1} &= T_N + \left(1 + \frac{\gamma_N^T T_N\gamma_N}{\delta_N^T\gamma_N}\right)\frac{\delta_N\delta_N^T}{\delta_N^T\gamma_N} - \\ &\quad -\frac{\delta_N\gamma_N^T T_N + T_N\gamma_N\delta_N^T}{\delta_N^T\gamma_N}, \qquad (24)\\ \delta_N &= \theta_{N+1} - \theta_N,\\ \gamma_N &= g_{N+1} - g_N.\end{aligned}$$

Notice that the BFGS formula is typically used for minimizing a cost function which does not change from one iteration to the next one. This property is not satisfied by the time-varying cost function $V_N(\theta)$ in (15). However, as we explain in Appendix B, this formula still applies when the cost function is time-varying.

Finally, the step-size parameter μ_N is obtained from a linear search algorithm. In this work we implement it using a backtracking procedure formed *by sub-iterations* of the *main iterations* (17), in which, starting from the initial value $\mu_{N,1} = 1$, the value of $\mu_{N,i}$ is halved at each sub-iteration until

$$V_{N+1}(\theta_N - \mu_{N,i}T_N g_N) < V_{N+1}(\theta_N), \qquad (25)$$

or a maximum number of sub-iterations is reached.

The recursive estimation method (17) requires an initialization, i.e., the choice of initial value θ_1. This can be easily obtained by choosing the nominal design values. Alternatively, a reference input can be used to obtain an initial estimate, as described in [9].

IV. Design of the Reconstruction Filters

In this section we propose an alternative to the method in [5], [13], for designing the reconstruction filters $\mathbf{f}(z)$. More precisely, we drop the bandlimited constraint on the input signal $x(t)$, and we assume instead that it has a (quasi-)stationary power spectrum $\Phi(s)$. In Section IV-A we assume that $\Phi(s)$ is known, and we design the synthesis filterbank $\mathbf{f}(z)$ using a linear LMS criterion [29], i.e., aiming at minimizing the power of the reconstruction error

$$\mathbf{e}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k). \qquad (26)$$

In Section IV-B we explain how to estimate the input spectrum $\Phi(s)$, using a variant of the estimation algorithm described in Section III. Finally, in Section IV-C we show that the design proposed in [5], [13] is a particular case of our proposed design.

A. Design assuming that the input spectrum is known

Using the polyphase representation [30], the scheme in Figure 1 can be transformed into that of Figure 2, where

$$\begin{aligned}\underline{\mathbf{y}}(k) &= [\mathbf{y}(kD), \mathbf{y}(kD-1), \cdots, \mathbf{y}(kD-D+1)]^T\\ \underline{\hat{\mathbf{y}}}(k) &= [\hat{\mathbf{y}}(kD), \hat{\mathbf{y}}(kD-1), \cdots, \hat{\mathbf{y}}(kD-D+1)]^T\end{aligned}$$

are the polyphase representations of $\mathbf{y}(k)$ and $\hat{\mathbf{y}}(k)$, respectively. Notice that we use underlined letters to denote the polyphase representation of a quantity. Also, the $D \times M$ matrix $\underline{\mathbf{F}}(z)$ is the polyphase representation of the synthesis filterbank, defined such that the impulse response $[\underline{\mathbf{f}}(k)]_{d,m}$ of its (d,m)-entry is given by:

$$[\underline{\mathbf{f}}(k)]_{d,m} = \mathbf{f}_m(kD + d - 1),$$

where $\mathbf{f}_m(k)$ denotes the impulse response of $\mathbf{F}_m(z)$.

In view of Figure 2, we can restate the problem as that of designing $\underline{\mathbf{F}}(z)$ for estimating $\underline{\mathbf{y}}(k)$ using $\mathbf{x}(k)$. If the support of the impulse response $\underline{\mathbf{f}}(k)$ of $\underline{\mathbf{F}}(z)$ is constrained so that $\underline{\mathbf{f}}(k) = 0$ if $k < a$ or $k > b$, the LMS solution can be found by solving

$$\underline{\mathbf{F}} = \arg\min_{\underline{\tilde{\mathbf{F}}}}\mathcal{E}\left\{\left\|\underline{\mathbf{y}}(0) - \sum_{l=a}^{b}\underline{\tilde{\mathbf{f}}}(l)\mathbf{x}(-l)\right\|_2^2\right\} \qquad (27)$$

where $\mathcal{E}\{\cdot\}$ denotes expected value. Now, the solution of (27) requires that the estimation error is orthogonal to the the data used in the estimation, i.e.,

$$\mathcal{E}\left\{\left(\underline{\mathbf{y}}(0) - \sum_{l=a}^{b}\underline{\mathbf{f}}(l)\mathbf{x}(-l)\right)\mathbf{x}^T(-k)\right\} = 0,$$

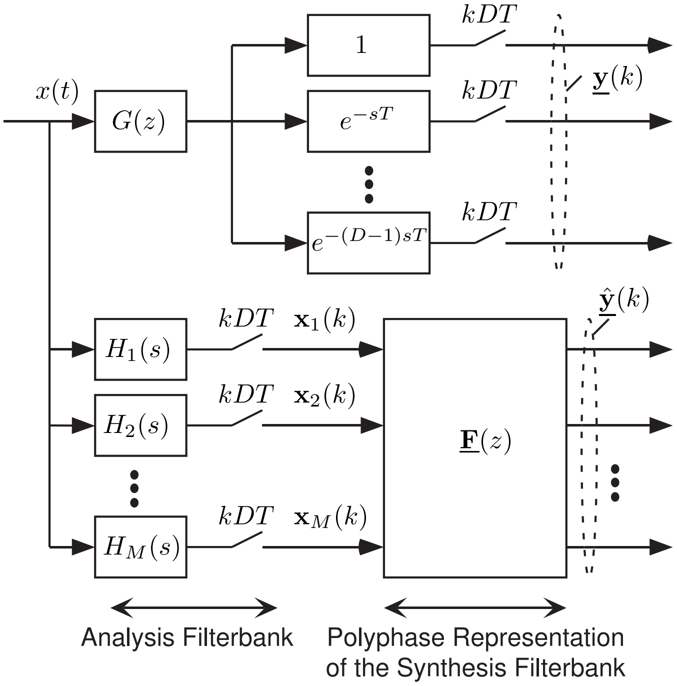

Figure 2. Transformed scheme using polyphase representation.

for all $k \in \{a, \cdots, b\}$, or equivalently,

$$\mathbf{r}_{\underline{\mathbf{y}}\mathbf{x}}(k) = \sum_{l=a}^{b} \underline{\mathbf{f}}(l)\mathbf{r}_{\mathbf{x}}(k-l), \text{ for all } k \in \{a, \cdots, b\}, \quad (28)$$

where $\mathbf{r}_{\mathbf{x}}(k)$ and $\mathbf{r}_{\underline{\mathbf{y}}\mathbf{x}}(k)$ denote the correlation matrix of $\mathbf{x}(k)$ and the cross-correlation matrix between $\underline{\mathbf{y}}(k)$ and $\mathbf{x}(k)$, respectively, i.e.,

$$\begin{aligned} \mathbf{r}_{\mathbf{x}}(k) &= \mathcal{E}\{\mathbf{x}(k)\mathbf{x}^T(0)\}, & (29) \\ \mathbf{r}_{\underline{\mathbf{y}}\mathbf{x}}(k) &= \mathcal{E}\{\underline{\mathbf{y}}(k)\mathbf{x}^T(0)\}. & (30) \end{aligned}$$

Hence, the impulse response $\underline{\mathbf{f}}(k)$ of the polyphase matrix $\underline{\mathbf{F}}(z)$ can be obtained by solving the linear problem (28). More precisely,

$$\begin{bmatrix} \underline{\mathbf{f}}(a) \\ \vdots \\ \underline{\mathbf{f}}(b) \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{\mathbf{x}}(0) & \cdots & \mathbf{r}_{\mathbf{x}}(a-b) \\ \vdots & \ddots & \vdots \\ \mathbf{r}_{\mathbf{x}}(b-a) & \cdots & \mathbf{r}_{\mathbf{x}}(0) \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{r}_{\underline{\mathbf{y}}\mathbf{x}}(a) \\ \vdots \\ \mathbf{r}_{\underline{\mathbf{y}}\mathbf{x}}(b) \end{bmatrix}, \quad (31)$$

where the superscript † denotes the (Moore-Penrose) pseudoinverse [31]. Finally, we need the expressions of $\mathbf{r}_{\mathbf{x}}(k)$ and $\mathbf{r}_{\underline{\mathbf{y}}\mathbf{x}}(k)$. It is straightforward to verify that

$$\begin{aligned} \mathbf{r}_{\mathbf{x}}(k) &= \downarrow_{DT} \{H(s)\Phi(s)H^*(s)\}(k) & (32) \\ \mathbf{r}_{\underline{\mathbf{y}}\mathbf{x}}(k) &= \downarrow_{DT} \{\Delta(s)G(s)\Phi(s)H^*(s)\}(k) & (33) \end{aligned}$$

where $\Delta(s) = [1, e^{-s}, \cdots, e^{-(M-1)s}]$.

B. Input spectrum estimation

At sample-time N, the design of the reconstruction filters, presented in Section IV-A, requires knowledge of the input spectrum $\Phi(t,s) = \mathcal{L}_\tau\{r(t,\tau)\}$ ($\mathcal{L}_\tau\{\cdot\}$ denotes the two-sided Laplace transform with respect to τ) at time $t = DTN$. Now, the criterion (14)-(15), introduced in Section III-A for estimating the input filters $H(s)$, produces as a by-product an estimate $\Phi(s, \beta_N(\theta_N))$ of the averaged input spectrum $\Phi_N(s)$, up to time $t = DTN$. This estimate is obtained as follows:

(S1) Use the estimate θ_N of θ, available at sample time N, in (10), to obtain an estimate $\beta_N(\theta_N)$ of β.

(S2) Use $\beta_N(\theta_N)$ in (9) to obtain $\Phi(s, \beta_N(\theta_N))$.

The average $\Phi_N(s)$ is obtained over a time span which is determined by the magnitude of the forgetting factor λ, and hence differs from the instantaneous input spectrum $\Phi(t,s)$, at $t = DTN$. As we show in Section VI-E, via simulation results, the accurate knowledge of the input spectrum is not critical for designing the reconstruction filters. Hence, one possibility is to simply use the estimate $\Phi(s, \beta_N(\theta_N))$ in place of $\Phi(DTN, s)$ for designing these filters. However, a problem in doing so is that, since the filters $H(s)$ change slowly with time, its estimation uses a long time span for averaging. Depending on the application, it may happen that using such a long averaging time prevents the tracking of changes on the input spectrum, if they are sufficiently fast. As a consequence of this, it may happens that $\Phi_N(s)$ is not good enough, and a better approximaiton of $\Phi(DTN, s)$ is needed.

If this is the case, we can do so using a second algorithm for estimating $\Phi(DTN, s)$, which runs in parallel to the one used for estimating $H(s)$. This second algorithm uses the estimate θ_N of the analysis filter parameters, available at time $t = DTN$, to obtain an estimate $\breve{\Phi}(s, \breve{\beta}_N(\theta_N))$ of the input spectrum $\Phi(DTN, s)$, using (9)-(10). To track fast changes in the input spectrum, the sample-average autocorrelation $\breve{\bar{\mathbf{r}}}_N(k)$, used for this second algorithm, is built using a forgetting factor $\breve{\lambda}$ smaller than the one (λ) used for estimating $H(s)$. More precisely, choosing a suitable forgetting factor $\breve{\lambda} < \lambda$, we compute

$$\breve{\bar{\mathbf{r}}}_N(k) = \frac{1}{\breve{\gamma}(N)} \sum_{n=1}^{N} \breve{\lambda}^{N-n} \mathbf{x}(n+k)\mathbf{x}^T(n)$$

where $\breve{\gamma}(N) = \sum_{n=0}^{N-1} \breve{\lambda}^n$. Then we compute $\breve{v}_N(\theta_N)$ using

$$\breve{v}_N(\theta_N) = \left\langle \breve{\bar{\mathbf{r}}}_N(k), \downarrow_{DT} \{H(s,\theta_N)E_i(s)H^*(s,\theta_N)\} \right\rangle .$$

Subsequently, we compute the estimate $\breve{\beta}_N(\theta_N)$ of β using

$$\breve{\beta}(\theta_N) = W^{-1}(\theta_N)\breve{v}_N(\theta_N),$$

where $W(\theta_N)$ is given by (11). Finally, $\breve{\Phi}(s, \breve{\beta}_N(\theta_N))$ is given by

$$\breve{\Phi}(s, \breve{\beta}_N(\theta_N)) = \sum_{i=1}^{I} [\breve{\beta}_N(\theta_N)]_i E_i(s).$$

Notice, that the speed of change of the input spectrum that can be tracked is limited by the property stated in Remark 2. More precisely, changes on the input spectrum need to be slow enough so that it can be considered quasi-stationary over a time span equal to the impulse response length of the analisys filters, which is a mild requirement.

C. Comparison with the approach in [13] and [5]

An approach for designing the reconstruction filters $\mathbf{F}(z)$ was proposed in [5], and improved in [13]. This method assumes that $M = D$, $G(s) = 1$ and the signal $x(t)$ is bandlimited to $1/2T$. Under this assumption, $\mathbf{F}(z)$ is designed as follows:

$$\mathbf{F} = \arg\min_{\tilde{\mathbf{F}}} \int_{-\pi}^{\pi} \left| \frac{1}{M} \sum_{m=0}^{M-1} \left(U_m - \mathbf{F}^T(e^{j\omega}) \mathbf{H}\left(e^{j(\omega - \frac{2\pi m}{M})} \right) \right) \right|^2 \tag{34}$$

where $U_m = 1$ for $m = 0$ and 0 otherwise, and $\mathbf{H}(z)$ is a discrete-time equivalent of the analysis filterbank $H(s)$, whose frequency response is given by

$$\mathbf{H}(e^{j\omega}) = H\left(j\frac{\omega}{T} \right), \, \omega \in [-\pi, \pi]. \tag{35}$$

Moreover, perfect reconstruction can be achieved (i.e., $\hat{\mathbf{y}}(k) = \mathbf{y}(k)$) if the impulse response $\mathbf{f}(k)$ can be arbitrary large (i.e., if $\mathbf{f}(k) \neq 0$ can hold for all $k \in \mathbb{Z}$).

In this section we show that the synthesis filterbank design (34) is equivalent to our proposed design (28)-(33) when $M = D$, $G(s) = 1$ and the input power spectrum is given by

$$\Phi(j\omega) = \begin{cases} 1, & |\omega| < \frac{\pi}{T} \\ 0, & \text{otherwise} \end{cases}. \tag{36}$$

Under these assumptions, it holds that $x(t) = \sum_{k=-\infty}^{\infty} \mathbf{y}(k) \operatorname{sinc}\left(\frac{t}{T} - k \right)$, and therefore

$$\mathbf{x}(k) \;=\; \sum_{l=-\infty}^{\infty} \mathbf{h}(l) \mathbf{y}(k-l), \tag{37}$$

where $\mathbf{h}(k) = \mathcal{Z}^{-1}\{\mathbf{H}(z)\}$ is the impulse response the discrete-time equivalent analysis filterbank (35). Let $\mathbf{y}'(k)$, $\hat{\mathbf{y}}'(k)$ and $\mathbf{e}'(k)$ denote truncated realizations of $\mathbf{y}(k)$, $\hat{\mathbf{y}}(k)$ and $\mathbf{e}(k)$, respectively[1]. Then, using the alias representation [30], we can write

$$\hat{\mathbf{Y}}'(z) = \frac{1}{M} \mathbf{F}^T(z) \mathbf{H}_A(z) \mathbf{Y}'_A(z)$$

where $\mathbf{H}_A(z)$ and $\mathbf{Y}'_A(z)$ denote the alias representations of $\mathbf{H}(z)$ and $\mathbf{Y}'(z)$, respectively, which are given by

$$\mathbf{H}_A(z) \;=\; [\mathbf{H}(z), \mathbf{H}(e^{j\frac{2\pi}{D}} z), \cdots, \mathbf{H}(e^{j\frac{2\pi(D-1)}{D}} z)] \tag{38}$$

$$\mathbf{Y}'_A(z) \;=\; [\mathbf{Y}'(z), \mathbf{Y}'(e^{j\frac{2\pi}{D}} z), \cdots, \mathbf{Y}'(e^{j\frac{2\pi(D-1)}{D}} z)]^T \tag{39}$$

Now, letting $\mathbf{u} = [1, 0, \cdots, 0]$, we can write $\mathbf{Y}'(z) = \mathbf{u}\mathbf{Y}'_A(z)$ and $\mathbf{E}'(z) = \left(\frac{1}{M} \mathbf{F}^T(z) \mathbf{H}_A(z) - \mathbf{u} \right) \mathbf{Y}'_A(z)$. Now, (36) implies that $\mathbf{y}'_A(k)$ is a white vector random process, then the LMS criterion for designing $\mathbf{F}(z)$ becomes

$$\mathbf{F} \;=\; \arg\min_{\tilde{\mathbf{F}}} \int_{-\pi}^{\pi} \left\| \frac{1}{M} \tilde{\mathbf{F}}^T(e^{j\omega}) \mathbf{H}_A(e^{j\omega}) - \mathbf{u} \right\|_2^2 d\omega$$

which, in view of (38), is equivalent to (34).

Equation	Complexity [mult./DT sec.]
$v(\theta_N)$	$\Psi_v = PM^2K$
$W(\theta_N)$	$\Psi_W = P^2M^2K$
$\frac{\partial}{\partial \theta_p} \mathbf{r}_N(k, \theta_N), \forall k, p$	$\Psi_{\nabla r} = 2PKI$
g_N	$\Psi_g = PM^2K + \Psi_{\nabla r}$
T_N	$\Psi_T = 3P^2 + 7P + 2$
$\mu_N T_N g_N$	$P(1+P) + \Psi_g + \Psi_T$

Table II
COMPLEXITY OF INTERMEDIATE TERMS NEEDED FOR THE BLIND ESTIMATION ALGORITHM.

V. COMPLEXITY AND IMPLEMENTATION

In this section we analyze the numerical complexity of the proposed algorithms. To this end we use the number of multiplications as the complexity measure. In particular, solving a positive definite linear system of n equations requires $\Psi_{\text{cho}}(n) = n^3/3$ multiplications [32, Sec. 4.2]. Also, computing the impulse response of a continuous system, at n given sample times, requires the computation of a residue-pole decomposition, plus n times that of the exponential function. To estimate the associated complexity, we assume that the system's poles and zeros are readily available. This assumption is valid since system orders are relatively low, and the complexity associated with computing their poles and zeros is negligible compared to the overall complexity. Then, the computation of a residue of a system of m zeros and n poles requires $\Psi_{\text{res}}(m, n) = m + n - 3$ multiplications. Also, computing the exponential function requires $\Psi_{\exp} = 20$ multiplications, because we consider 20 terms in the expansion $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$, which guarantees that the residual error is smaller than 10^{-6}, for $x \in [-6, 0]$.

Using the above, we state the complexity of each proposed algorithm. Since these algorithms are recursively computed once per sample time, we express their complexity in multiplications per DT seconds. We assume that the order of the numerator and the denominator of $E_i(s)$ are the same, for all $i = 1, \cdots, I$, and that the same condition holds for $H_m(s)$, for all $m = 1, \cdots, M$. We then denote by n_E, d_E, n_H, d_H, n_G and d_G, the number of roots of the numerator of $E_i(s)$, the denominator of $E_i(s)$, the numerator of $H_m(s)$, the denominator of $H_m(s)$, the numerator of $G(s)$ and the denominator of $G(s)$, respectively. We also define $n_1 = n_E + 2n_H$, $d_1 = d_E + 2d_H$, $n_2 = n_E + 2n_H$, $d_2 = d_E + 3d_H$, $n_3 = n_E + n_H + n_G$ and $d_3 = d_E + d_H + d_G$.

In Table I we state the complexity of a number of terms used in different parts of the blind estimation algorithm presented in Section III. They need to be computed only once per sample time.

Using the terms shown in Table I, we can compute those whose complexity is given in Table II.

In addition to the terms shown in Tables I and II, the linear search algorithm (25) requires the evaluation of $V_N(\theta)$ at different values of θ. Each evaluation requires $\Psi_V = M^2K + \Psi_1$ multiplications. Then, denoting by Q the number of linear search steps in (25), Table III shows the complexity of each task involved in the blind estimation algorithm, as well as the sample reconstruction algorithm.

[1] So that their z-transforms $\mathbf{Y}'(z)$, $\hat{\mathbf{Y}}'(z)$ and $\mathbf{E}'(z)$ are well defined on the unit circle.

Equation	Complexity [mult./DT sec.]
$\downarrow_{DT} \{H(s,\theta_N)E_i(s)H^*(s,\theta_N)\}(k)$, $\forall k, i$	$\Psi_1 = IM^2 \frac{d_1}{4}(\Psi_{\text{res}}(n_1,d_1) + K\Psi_{\text{exp}})$
$\downarrow_{DT} \left\{\frac{\partial}{\partial\theta_p} H(s,\theta_N)E_i(s)H^*(s,\theta_N)\right\}(k)$, $\forall k, i, p$	$\Psi_2 = IPM \frac{d_2}{2}(\Psi_{\text{res}}(n_2,d_2) + K\Psi_{\text{exp}})$
$\downarrow_{DT} \{\Delta(s)G(s)E_i(s)H^*(s,\theta_N)\}(k)$, $\forall k, i$	$\Psi_3 = IM \frac{d_3}{2}(\Psi_{\text{res}}(n_3,d_3) + K\Psi_{\text{exp}})$
$\bar{\mathbf{r}}_N(k)$, $\forall k$	$\Psi_{\bar{\mathbf{r}}} = \frac{M(M+1)}{2}K$

Table I
COMPLEXITY OF BASIC TERMS NEEDED FOR THE BLIND ESTIMATION ALGORITHM.

Algorithm	Complexity [mult./DT sec.]
Input spectrum estimation	$\Psi_{\text{cho}}(P) + \Psi_v + \Psi_W$
Filterbank estimation	$P(1+P) + \Psi_g + \Psi_T + Q\Psi_V$
Computing reconstruction filters	$M\Psi_{\text{cho}}(KM)$
Sample reconstruction	$M^2(b-a+1)$

Table III
COMPLEXITY OF EACH TASK.

In a practical implementation, the digital processing algorithm needs first to compute the terms listed in Table I. These terms are then used to compute the tasks listed in Table III, whose components are detailed in Table II. Now, it can be computationally unaffordable to repeat these steps at the arrival of each new sample. However, notice that from all these computations, only the sample reconstruction task needs to be strictly carried out once per sample. The remaining computations are related to the estimation task. Hence, their computation can be carried out asynchronously to sample arrivals, to accommodate computational power limitations.

To explain this point in more detail, we provide below a sketch of the digital processing algorithm. The algorithm is divided in two routines. The *synchronous routine* is executed each time a new sample arrives, and carries out the sample reconstruction task. On the other hand, the *asynchronous routine* is continuously executed in the background and carries out the estimation task. In the sketch below we assume that the input spectrum is jointly estimated with the analysis filterbank parameters, as described in (S1)-(S2) in Section IV-B. If instead, a second algorithm is used for estimating the input spectrum with a smaller forgetting factor, this algorithm needs to be added to the asynchronous routine.

Digital processing algorithm:

- **Synchronous routine:** Whenever a new (vector) sample $\mathbf{x}(t)$ arrives (i.e., once every DT seconds),
  1) add $\mathbf{x}(t)$ to a *temporary buffer*;
  2) reconstruct the samples $\hat{\mathbf{y}}(kD), \hat{\mathbf{y}}(kD-1), \cdots, \hat{\mathbf{y}}(kD-D+1)$ using the available reconstruction filters $\mathbf{f}_m(k)$, $m = 1, \cdots, M$.
- **Asynchronous routine:** Continuously iterate over the following steps,
  1) update the available cost function $V_N(\theta)$ to $V_{N+n}(\theta)$, where n is the number of samples accumulated in the temporary buffer during the last iteration, and empty the buffer;
  2) execute a quasi-Newton iteration (17), to obtain a new estimate θ_{N+n} of the parameters θ, and hence a new estimate of the analysis filterbank $H(z)$;
  3) use θ to build a new estimate of the input spectrum $\Phi(s)$ (using (S1)-(S2) in Section IV-B);
  4) use the new estimates of $H(z)$ and $\Phi(s)$ to compute the reconstruction filters $\mathbf{f}_m(k)$, $m = 1, \cdots, M$, using (31)-(33).

VI. NUMERICAL EXPERIMENTS

In this section we present numerical experiments to illustrate the performance of the blind estimation method presented in Section III, as well as the sample reconstruction method presented in Section IV. To this end, following [13], we consider an eight-channel HFB-ADC, where for simplicity, we use the sampling period $T = 1$. The analysis filterbank is composed of Butterworth second-order bandpass filters of bandwidth $1/16$ Hz, except for the first one which is a first-order lowpass filter of the same bandwidth. The bandwidths are contiguously allocated so that they cover the whole frequency range from 0 Hz to 0.5 Hz. The output of each filter is sampled at $1/8$ Hz (i.e., the upsampling factor D in Figure 1 equals the number M of channels).

A. Performance of the proposed sample reconstruction method

In order to evaluate the proposed sample reconstruction method, we compare its performance with that of the method (34) (derived under a bandlimited assumption on the input signal), which we denote by (BL). The comparison is done in terms of the signal-to-distortion ratio (SDR) of the reconstructed samples, which is defined by

$$SDR = 10\log_{10}\left(\frac{\sum_{k=1}^{N}|\mathbf{y}(k)|^2}{\sum_{k=1}^{N}|\mathbf{y}(k) - \hat{\mathbf{y}}(k)|^2}\right).$$

For both methods we use $a = -3$ and $b = 4$ in (27), which results in the discrete-time filters $\mathbf{F}_m(z)$, $m = 1, \cdots, M$ having 64 taps with 31 non-causal taps. From Table III, the resulting reconstruction scheme requires 512 multiplications each DT seconds.

In the first simulation we use $G(s) = 1$. We generate the input signal as filtered white noise using a Butterworth lowpass filter $L(s)$ of 20-th order and varying cutoff frequency f_c. We call $L(s)$ the *input filter*, and its relation to the spectrum $\Phi(s)$ of the input signal $x(t)$ is given by $\Phi(s) = |L(s)|^2$. The frequency responses of several such filters with different values of f_c are shown in Figure 3. We compare the performances of the BL method and the proposed LMS method for several values of f_c. The result is shown in Figure 4. We see how the LMS method clearly outperforms the BL method, especially for low cutoff frequency values. The difference in performance is caused by the assumption (36) on the input spectrum, under which the BL method is designed. As pointed out in [17], this assumption results in reconstruction filters $\mathbf{F}_m(z)$, $m = 1, \cdots, M$ having very long impulse responses. Then, when theses filters are truncated to 64 taps, as described above, the reconstruction performance is significantly impaired.

Figure 4 also shows that the SDR in both methods decreases as the cutoff frequency f_c increases. This is a consequence of the generalized sampling theorem, which states that a signal which is bandlimited to $M/2DT$, can be reconstructed from the samples obtained after filtering it using M filters and at $1/D$-th of the Nyquist rate [33]. This implies that it is possible to perfectly reconstruct the input signal $x(t)$, at any time t (including $t = kT$, $k \in \mathbb{Z}$ as is our goal), only if it is bandlimited to 0.5 Hz. Hence, the reduced performance for high values of f_c is due to the leakage energy above 0.5 Hz.

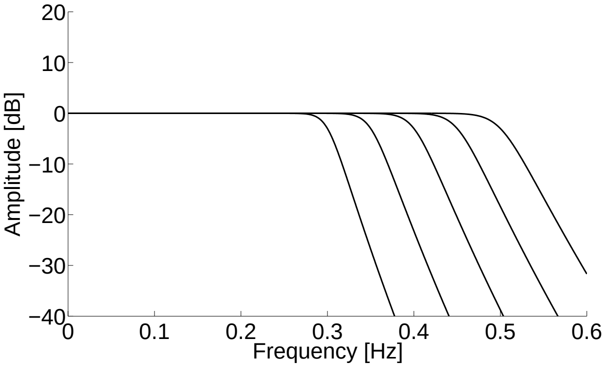

Figure 3. Frequency response of a family of input filters $L(s)$ with different cutoff frequency values.

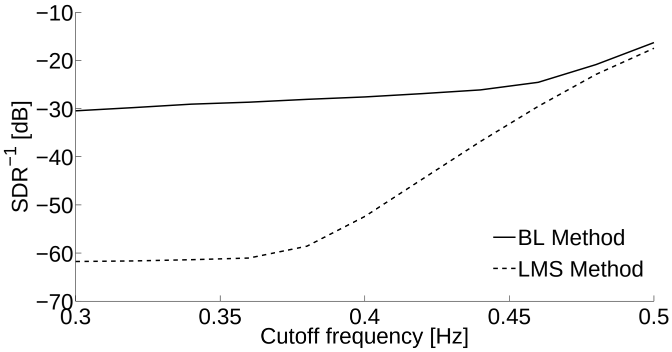

Figure 4. Performance comparison of the BL and LMS methods for different values of the input filter cutoff frequency.

B. Sample reconstruction using an anti-alias filter

In the second simulation we evaluate the performance of the proposed method when reconstructing the samples that would be obtained after filtering the input signal using a prescribed anti-alias filter $G(s)$. For the input filter $L(s)$ we use a Butterworth lowpass filter of 5-th order and varying cutoff frequency. Also, for the anti-alias filter $G(s)$ we use a Butterworth lowpass filter of 20-th order and fixed cutoff frequency at 0.4 Hz. The obtained SDR values for different cutoff frequencies are shown in Figure 5. As expected, the performance of the LMS method improves with the use of the anti-alias filter.

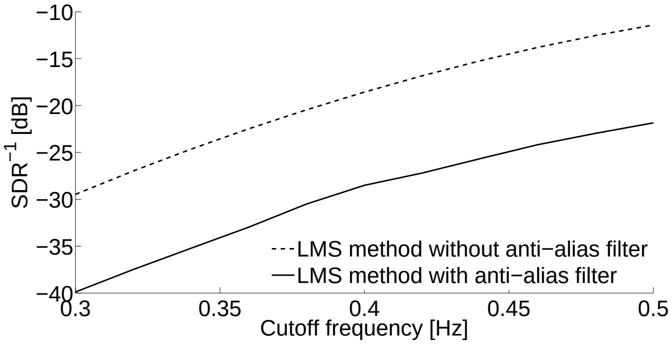

Figure 5. Performance of the LMS method, with and without anti-alias filter using a 5-th order Butterworth lowpass input filter.

C. Effect of Quantization

The effect of quantization in HFB-ADCs, under the bandlimited assumption described in Section IV-C, has been analyzed in [34]. In this section we study the effect of quantization on the proposed sample reconstruction method, via numerical simulations. We use $G(s) = 1$ and for $L(s)$ we use a Butterworth lowpass filter of 5-th order and cutoff frequency $f_c = 0.3$ Hz. In Figure 6 we compare the SDR obtained after quantizing the ideal samples $\mathbf{y}(k)$, with that obtained after quantizing the HFB-ADC samples $\mathbf{x}_m(k)$, $m = 1, \cdots, M$. For comparison purposes, we use the same number of bits per sample in both schemes[2]. Since the HFB-ADC has $M = 8$ channels, each of which is sampled $D = 8$ times slower than the ideal samples, the ADC on each channel uses the same number of bits than that used for quantizing the ideal samples. In Figure 6 we show the SDR obtained using different quantization bits. We see that the consequence of quantization is similar in both schemes.

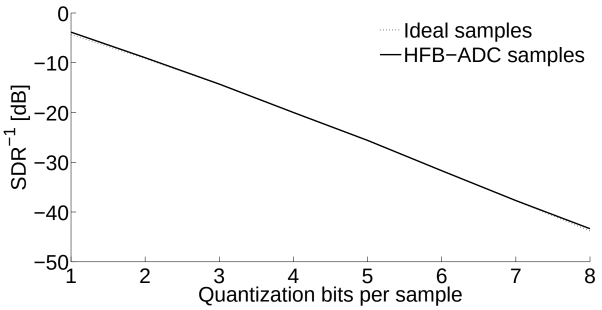

Figure 6. SRD obtained after quantization of ideal and HFB-ADC samples.

[2] Keeping constant the number of bits per sample is natural, as this is the fundamental constraint in commercial ADCs. If the proposed scheme is implemented on a standard embedded platform like a DSP, then it is typical that the number of available bits per channel depends on the number of channels used, keeping constant the total number of bits. See for example [35].

D. Performance of the proposed blind estimation method

In this section we evaluate the performance of the blind estimation method proposed in Section III. We use an analysis filterbank which is obtained by perturbing the *nominal analysis filterbank* used in the previous simulation. The perturbation is done by multiplying the real and imaginary components of each pole by $1+\epsilon$, with ϵ being a Gaussian random variable with standard deviation $\sigma_\epsilon = 0.2$. For the adaptive blind estimation algorithm we use a forgetting factor of $\lambda = 1-10^{-5}$, so that measurements that are older than 10^5 samples are included in the criterion with a weight that is at most $e^{-1} = 0.3679$ the weight of the most recent measurement. We then run the algorithm over 10^6 samples.

For the linear expansion (9), we use 12 basis elements $E_i(s)$, $i = 1, \cdots, 12$, chosen as

$$\begin{aligned} E_1(s) &= \frac{1}{(s-a_1)(s-\bar{a}_1)(s+a_1)(s+\bar{a}_1)}, \\ E_i(s) &= \frac{s^2}{(s-a_i)(s-\bar{a}_i)(s+a_i)(s+\bar{a}_i)}, \; i = 2, \cdots, 12, \end{aligned}$$

with $a_1 = 0.1714$ and $a_i = 0.1714\,(1 + j(2i-1))$, $i = 2, \cdots, 12$, and the overbar denoting complex conjugation. Their frequency response is shown in Figure 7.

In view of Table III, the joint estimation of the input spectrum and the filterbank requires 1.322×10^6 multiplications, each DT seconds, plus 137×10^3 multiplications for each linear search iteration. Then, the computation of the reconstruction filters requires 87.38×10^3 multiplications.

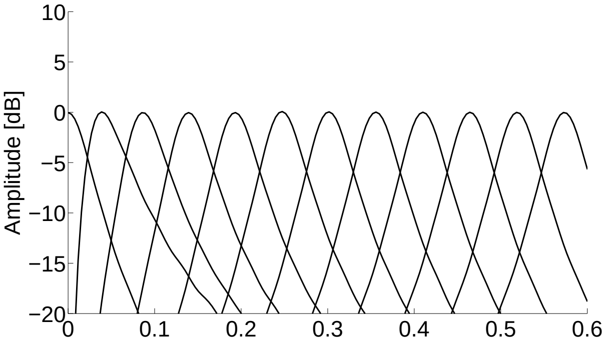

Figure 7. Basis elements $E_i(s)$, $i = 1, \cdots, 12$ used for approximating the input power spectrum.

We evaluate the performance of the blind estimation method using two scenarios:

(a) In the first simulation we generate an input signal with a time-varying power spectrum $\Phi(t,s)$, $t \in \mathbb{R}$. We do so using a 3-th order time-varying input filter $L(t,s)$, so that $\Phi(t,s) = |L(t,s)|^2$, $t \in \mathbb{R}$. The input filter $L(t,s)$ is designed so that it has no zeros, and poles at -3.142 and $-0.4854 \pm j1.494 + 0.5 \times \sin(2\pi t/5000)$. Hence, the imaginary component of the complex poles oscillate so that 1600 cycles are included within 10^5 samples. Also, each cycle is 100 times longer than the impulse response length of the analysis filters $H(s)$, so that the quasi-stationary requirement in Remark 2 is satisfied. In Figs. 8 and 9 we show the estimated input spectrum and analysis filters, respectively. We see that both, the input spectrum and the analysis filters are accurately estimated up to a threshold frequency of about 0.4 Hz. This is due to the low power level available at high frequencies.

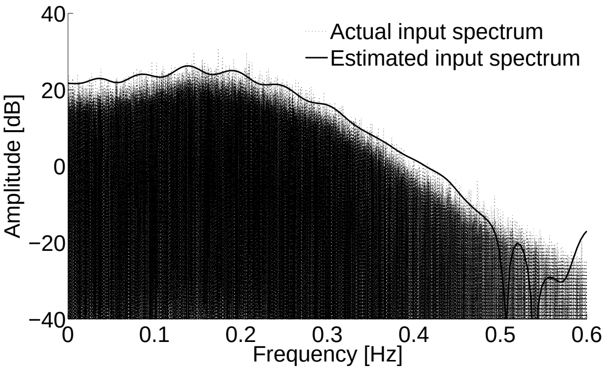

Figure 8. Actual and estimated input power spectra, when low power is available at high frequencies.

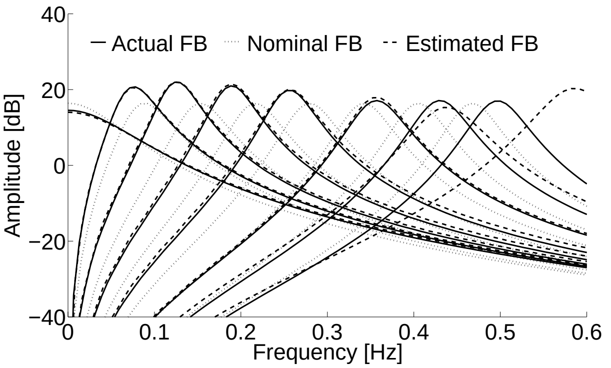

Figure 9. Frequency responses of the actual, nominal and estimated analysis filterbanks, when low power is available at high frequencies.

(b) In the second simulation we repeat the same experiment, but we modify the input filter $L(t,s)$ so that a more significant power level is available in the high frequency range. To this end, we choose $L(t,s)$ having no zeros, and poles at -3.142 and $-0.6796 \pm j2.092 + 0.5 \times \sin(2\pi t/5000)$. The results are shown in Figures 10 and 11. In this case, both the input spectrum and the analysis filters are properly estimated.

In Table IV we show the SDR values obtained using the actual, the nominal and the estimated filterbanks. We do so considering the filterbanks estimated in both scenarios, namely: (a) when low power is available at high frequencies, and (b) with a significant power level at all frequencies. For this comparison we generate the input signal as described in scenario (a).

We conclude that an accurate estimate of the analysis filterbank is relevant in the HFB-ADC design. Also, while analysis filters having a low power level in their passbands are not accuratley estimated, this does not seriously undermine the reconstruction performance.

Actual filterbank	Nominal filterbank	Estimated filterbank (a)	Estimated filterbank (b)
-19.82 dB	-3.21 dB	-16.64 dB	-18.48 dB

Table IV
COMPARISON OF SDR^{-1} OBTAINED USING THE ACTUAL, THE NOMINAL AND THE ESTIMATED ANALYSIS FILTERBANKS.

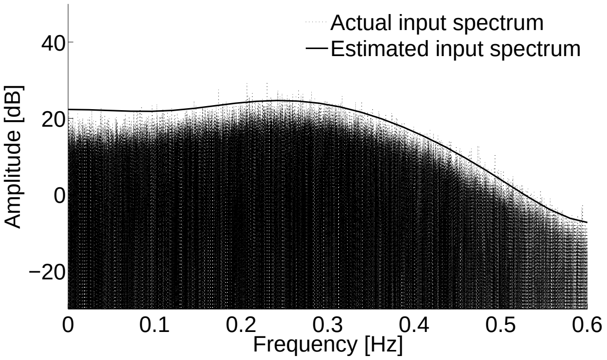

Figure 10. Actual and estimated input power spectra, with significant power level at all frequencies.

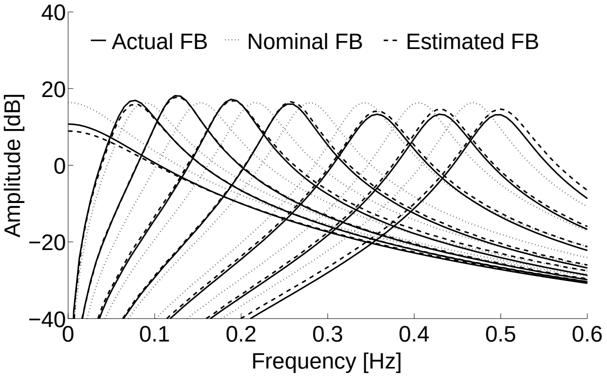

Figure 11. Frequency responses of the actual, nominal and estimated analysis filterbanks, with significant power level at all frequencies.

E. Irrelevance of an accurate input spectrum estimate

In this last simulation we illustrate that an accurate estimation of the input spectrum is not critical to the reconstruction error. To this end, we evaluate the performance degradation of the proposed LMS design method when there is perfect knowledge of the analysis filters $H(s)$ but imperfect knowledge of the input power spectrum. The actual input signal power spectrum is determined by the *actual input filter*, while the available input power spectrum used to design the LMS compensator is determined by an *available input filter*. We design the available input filter as a Butterworth lowpass filter of 20-th order and varying cutoff frequency f_c. For the actual input filter we use a Butterworth lowpass filter of 10-th order and varying cutoff frequency f_c in cascade with a second order filter with poles in $-0.0782 f_c \pm j0.4938 f_c$. The frequency responses of the available and the actual input filters are shown in Figure 12, and the simulation result is shown in Figure 13. We see that, while not being optimal, the performance of the LMS method does not deteriorate significantly.

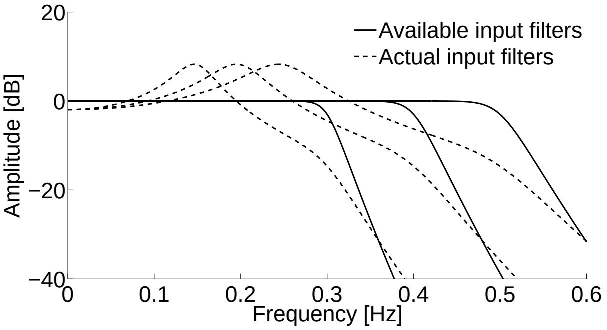

Figure 12. Frequency responses of available and actual input filters for different cutoff frequency values.

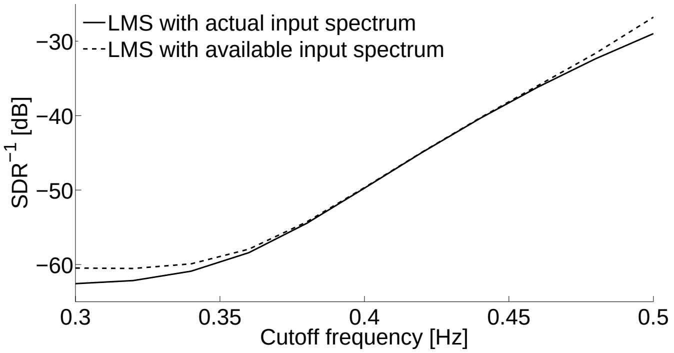

Figure 13. Performance degradation of the LMS method in the presence of input power spectrum mismatch.

VII. CONCLUSION

We have proposed an adaptive blind method for estimating the analysis filterbank parameters in a hybrid filterbank analog-to-digital converter. This estimation method is able to cope with non-stationary input signals. We have also presented a design method for the sample reconstruction stage, by using the estimated analog parameters. The reconstruction is done by minimizing the power of the reconstruction error in the samples. To this end, the spectrum of the input signal is adaptively estimated. We have shown that existing approaches based on a bandlimited assumption on the input signal are particular cases of our proposed design. We have presented numerical experiments showing the improved performance of the blind estimation method, and the clear advantage of the proposed LMS design.

Appendix A
Proofs

Proof of Lemma 1: We have that $\mathbf{x}(n) = \int_{-\infty}^{\infty} h(t)x(nDT-t)\,dt$. Then, from (3) we obtain

$$\begin{aligned}\mathbf{r}_N(k) &= \frac{1}{\gamma(N)}\sum_{n=1}^{N}\lambda^{N-n}\times \\ &\quad\times\mathcal{E}\left\{\int_{-\infty}^{\infty} h(t)x((n+k)DT-t)\,dt\times\right. \\ &\quad\left.\times\int_{-\infty}^{\infty} x(nDT-\delta)h^T(\delta)\,d\delta\right\}.\end{aligned}$$

Now, since $\int_{-\infty}^{\infty}|h_m(t)|dt < \infty, m = 1,\cdots,M$ and $\mathcal{E}\{x^2(t)\} < C$ for all $t\in\mathbb{R}$, in view of Fubini's theorem, we can exchange the expectation with the integrations. By doing so we obtain

$$\begin{aligned}\mathbf{r}_N(k) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t)\frac{1}{\gamma(N)}\sum_{n=1}^{N}\lambda^{N-n}\times \\ &\quad\times\mathcal{E}\left\{x((n+k)DT-t)x(nDT-\delta)\right\}h^T(\delta)\,dtd\delta \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t)\frac{1}{\gamma(N)}\sum_{n=1}^{N}\lambda^{N-n}\times \\ &\quad\times r(nDT-\delta,kDT+\delta-t)h^T(\delta)\,dtd\delta\end{aligned}$$

Now, using (1) and defining

$$\begin{aligned}\varepsilon(\xi) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t)\frac{1}{\gamma(N)}\sum_{n=1}^{N}\lambda^{N-n}\times \\ &\quad\times\left(r(nDT-\delta,\xi+\delta-t)-\right. \\ &\quad\left.-r(nDT,\xi+\delta-t)\right)h^T(\delta)\,dtd\delta,\end{aligned}$$

we have that

$$\begin{aligned}\mathbf{r}_N(k) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t)\frac{1}{\gamma(N)}\sum_{n=1}^{N}\lambda^{N-n}\times \\ &\quad\times r(nDT,kDT+\delta-t)h^T(\delta)\,dtd\delta+\varepsilon(kDT) \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(t)r_N(kDT+\delta-t)h^T(\delta)\,dtd\delta+ \\ &\quad+\varepsilon(kDT) \\ &= (h*r_N*h^*)(kDT)+\varepsilon(kDT) \\ &= \mathcal{L}^{-1}\left\{H(s)\Phi^N(s)H^*(s)\right\}(kDT)+\varepsilon(kDT).\end{aligned}$$

Finally, to show 6, from 4 we have that

$$\begin{aligned}|\varepsilon(\xi)| &\leq \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|h(t)|\frac{1}{\gamma(N)}\sum_{n=1}^{N}\lambda^{N-n}\times \\ &\quad\times\Downarrow r(nDT-\delta,\xi+\delta-t)- \\ &\quad-r(nDT,\xi+\delta-t)|\,|h^T(\delta)|\,dtd\delta \\ &\leq \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}|h(t)|\frac{1}{\gamma(N)}\sum_{n=1}^{N}\lambda^{N-n}\kappa(\delta)\times \\ &\quad\times\tilde{r}(\xi+\delta-t)|h^T(\delta)|\,dtd\delta \\ &= \left(|h|*\tilde{r}*|\kappa\circ h^*|\right)(\xi).\end{aligned}$$

■

Proof of (18)-(23): From (13), (9) we have that

$$\begin{aligned}&\frac{\partial}{\partial\theta_p}\mathbf{r}_N(k,\theta) = \\ &= \frac{\partial}{\partial\theta_p}\downarrow_{DT}\left\{H(s,\theta)\Phi_N(s,\beta_N(\theta))H^*(s,\theta)\right\}(k) \\ &= \sum_{i=1}^{I}\frac{\partial}{\partial\theta_p}\left([\beta_N(\theta)]_i\downarrow_{DT}\left\{H(s,\theta)E_i(s)H^*(s,\theta)\right\}(k)\right) \\ &= \sum_{i=1}^{I}[\beta_N(\theta)]_i\frac{\partial}{\partial\theta_p}\downarrow_{DT}\left\{H(s,\theta)E_i(s)H^*(s,\theta)\right\}(k) \\ &+\sum_{i=1}^{I}\frac{\partial}{\partial\theta_p}[\beta_N(\theta)]_i\downarrow_{DT}\left\{H(s,\theta)E_i(s)H^*(s,\theta)\right\}(k) \\ &= \sum_{i=1}^{I}[\beta_N(\theta)]_i\mathbf{d}_{p,i}(k) \\ &+\sum_{i=1}^{I}\frac{\partial}{\partial\theta_p}[\beta_N(\theta)]_i\mathbf{c}_i(k).\end{aligned}$$

Now, equation (21) follows from (10) and the following property:

$$\begin{aligned}&M(\theta)M^{-1}(\theta) = I \\ \Leftrightarrow&\frac{\partial M(\theta)}{\partial\theta_p}M(\theta)^{-1}+M(\theta)\frac{\partial M(\theta)^{-1}}{\partial\theta_p} = \mathbf{0} \\ \Leftrightarrow&\frac{\partial M(\theta)^{-1}}{\partial\theta_p} = M(\theta)^{-1}\frac{\partial M(\theta)}{\partial\theta_p}M(\theta)^{-1}.\end{aligned}$$

Also, equation (22) follows from (12) and (20), and equation (23) follows from (11), (19) and (20). ■

Appendix B
BFGS Formula for Adaptive Optimization

For a given $n\in\mathbb{N}$, let $g_n(\theta)$ and $H_n(\theta)$ denote the gradient and the Hessian matrix of $V_n(\theta)$, respectively. Using a Taylor expansion we have that

$$g_n(\theta_{n+1}) = g_n(\theta_n)+H_n(\theta_n)\delta_n+\epsilon_n,$$

where θ_{n+1} is obtained from (17), $\delta_n = \theta_{n+1}-\theta_n$ and $\epsilon_n = O\left(\|\delta_n\|_2^2\right)$. Also, let ρ_n be defined so that

$$g_{n+1}(\theta_{n+1}) = g_n(\theta_{n+1})+\rho_n.$$

Then,

$$g_{n+1}(\theta_{n+1}) = g_n(\theta_n)+H_n(\theta_n)\delta_n+\epsilon_n+\rho_n. \qquad (40)$$

Now, suppose that $V_{n+1}(\theta) = V_n(\theta)$, for all $\theta\in\mathbb{R}^P$ (i.e., the cost function is stationary). We then have that $\rho_n = 0$, for all $n\in\mathbb{N}$. Also, if $V_n(\theta)$ is a quadratic function, we have that $\epsilon_n = 0$ and $H_n(\theta_n) = H$ (i.e., the Hessian matrix is independent of n and θ), for all $n\in\mathbb{N}$. Hence, (40) becomes

$$\gamma_n = H\delta_n, \qquad (41)$$

where $\gamma_n = g_{n+1}(\theta_{n+1})-g_n(\theta_n)$. In this case, for a given $N\in\mathbb{N}$, we can find H by solving (41) for $n = 1,\cdots,N$, provided the matrix $\Delta_N = [\delta_1,\cdots,\delta_N]$ has full row rank.

In practice, neither $V_{N+1}(\theta) = V_N(\theta)$ nor $V_N(\theta)$ is quadratic. Then, assuming that we know an approximation S_N of $H_{N-1}(\theta_{N-1})$, in view of (41), we can build S_{N+1} by adding an update to S_N so that

$$\gamma_n = S_{N+1}\delta_n. \tag{42}$$

This is the starting point of the derivation of the BFGS method for a stationary cost function. The details on how to obtain (24) from (42) can be found in [28, Sec. 3.2]. The only difference between the stationary and non-stationary cases is that in the former, the updates on S_N are done to account for the fact that $V_N(\theta)$ is non-quadratic, while in the latter, these updates also account for the fact that $V_N(\theta)$ is non-stationary. The validity of this approach is justified as follows. When $V_N(\theta)$ is stationary (i.e., $V_N(\theta) = V(\theta)$), it follows from [28, Th. 3.4.1] that if there exists N_0 such that the sequence θ_N, $N \geq N_0$, belongs to a set where $V(\theta)$ is quadratic, then the BFGS formula converges to the minimum of $V(\theta)$ in at most p steps, provided the line searches in (25) are exact. In the non-stationary case, the same condition holds if additionally, $V_N(\theta)$ becomes stationary for $N \geq N_0$.

References

[1] W. C. J. Black and D. A. Hodges, "Time interleaved converter arrays," *IEEE Journal of Solid-State Circuits*, vol. SC-15, no. 6, pp. 1022 – 1029, 1980.

[2] C. Vogel, "The impact of combined channel mismatch effects in time-interleaved ADCs," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 1, pp. 415–427, 2005.

[3] S.-W. Sin, U.-F. Chio, U. Seng-Pan, and R. Martins, "Statistical spectra and distortion analysis of time-interleaved sampling bandwidth mismatch," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 55, no. 7, pp. 648–652, July 2008.

[4] A. Petraglia and S. Mitra, "High-speed A/D conversion incorporating a QMF bank," *IEEE Transactions on Instrumentation and Measurement*, vol. 41, no. 3, pp. 427–431, 1992.

[5] S. Velazquez, T. Nguyen, S. Broadstone, V. Co, and R. Beach, "Design of hybrid filter banks for analog/digital conversion," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 956–967, 1998.

[6] S. Mazlouman and S. Mirabbasi, "A frequency-translating hybrid architecture for wide-band analog-to-digital converters," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 7, pp. 576–580, 2007.

[7] C. Lelandais-Perrault, T. Petrescu, D. Poulton, P. Duhamel, and J. Oksman, "Wideband, bandpass, and versatile hybrid filter bank A/D conversion for software radio," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 8, pp. 1772–1782, 2009.

[8] S. H. Zhao and S. C. Chan, "Design and multiplierless realization of digital synthesis filters for hybrid-filter-bank A/D converters," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 10, pp. 2221–2233, 2009.

[9] Y. Sanada and M. Ikehara, "Decorrelating compensation scheme for coefficient errors of a filter bank parallel A/D converter," *IEEE transactions on wireless communications*, vol. 3, no. 2, pp. 341–348, 2004.

[10] Z. Song, C. Lelandais-Perrault, D. Poulton, and P. Bénabes, "Adaptive equalization for calibration of subband hybrid filter banks A/D converters," in *European Conference on Circuit Theory and Design (ECCTD)*, 2009, pp. 45–48.

[11] Z. Song, C. Lelandais-Perrault, and P. Bénabes, "Synthesis of complex subband hybrid filter banks A/D converters using adaptive filters," in *16th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*, 2009, pp. 399–402.

[12] Z. Song, C. Lelandais-Perrault, D. Poulton, and P. Bénabes, "Synthesis of subband hybrid filter banks ADCs with finite word-length coefficients using adaptive equalization," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 577–580.

[13] D. Asemani, J. Oksman, and P. Duhamel, "Subband Architecture for Hybrid Filter Bank A/D Converters," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 2, pp. 191–201, 2008.

[14] H. Shu, T. Chen, and B. Francis, "Minimax design of hybrid multirate filter banks," *IEEE Trans. Circuits Syst. II*, vol. 44, no. 2, pp. 120–128, 1997.

[15] H. Nguyen and M. Do, "Hybrid Filter Banks With Fractional Delays: Minimax Design and Application to Multichannel Sampling," *IEEE Transactions on Signal Processing*, vol. 56, no. 7 Part 2, pp. 3180–3190, 2008.

[16] D. Marelli, K. Mahata, and M. Fu, "Linear LMS compensation for timing mismatch in time-interleaved ADCs," in *34th Annual Conference of IEEE Industrial Electronics, 2008. IECON 2008.*, nov. 2008, pp. 1901 –1906.

[17] ——, "Linear LMS compensation for timing mismatch in time-interleaved ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 11, pp. 2476 –2486, nov. 2009.

[18] H. Johansson and P. Lowenborg, "Reconstruction of nonuniformly sampled bandlimited signals by means of digital fractional delay filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 11, pp. 2757 – 2767, 2002.

[19] R. S. Prendergast, B. C. Levy, and P. J. Hurst, "Reconstruction of band-limited periodic nonuniformly sampled signals through multirate filter banks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 8, pp. 1612 – 1622, 2004.

[20] C.-Y. Wang and J.-T. Wu, "A background timing-skew calibration technique for time-interleaved analog-to-digital converters," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 53, no. 4, pp. 299–303, April 2006.

[21] H. Jin and E. Lee, "A digital-background calibration technique for minimizing timing-error effects in time-interleaved adcs," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 7, pp. 603–613, Jul 2000.

[22] S. Jamal, D. Fu, M. Singh, P. Hurst, and S. Lewis, "Calibration of sample-time error in a two-channel time-interleaved analog-to-digital converter," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 1, pp. 130–139, Jan. 2004.

[23] J. Elbornsson, F. Gustafsson, and J.-E. Eklund, "Blind adaptive equalization of mismatch errors in a time-interleaved a/d converter system," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 1, pp. 151–158, Jan. 2004.

[24] S. Huang and B. Levy, "Blind calibration of timing offsets for four-channel time-interleaved adcs," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 54, no. 4, pp. 863–876, April 2007.

[25] A. Haftbaradaran and K. Martin, "A background sample-time error calibration technique using random data for wide-band high-resolution time-interleaved adcs," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 55, no. 3, pp. 234–238, March 2008.

[26] D. Marelli, K. Mahata, and M. Fu, "Optimal design of hybrid filterbank analog-to-digital converters using input statistics," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, apr. 2009, pp. 3377 –3380.

[27] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 1999.

[28] R. Fletcher, *Practical methods of optimization*, 2nd ed., ser. A Wiley-Interscience Publication. Chichester: John Wiley & Sons Ltd., 1987.

[29] T. Kailath, A. Sayed, and B. Hassibi, *Linear Estimation*, ser. Prentice Hall Information and System Sciences Series. Upper Saddle River, New Jersey: Prentice Hall, 2000.

[30] P. Vaidyanathan, *Multirate Systems and Filterbanks*. Englewood Cliffs, N.J.: Prentice Hall, 1993.

[31] A. Ben-Israel and T. N. E. Greville, *Generalized inverses*, 2nd ed., ser. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC, 15. New York: Springer-Verlag, 2003, theory and applications.

[32] G. H. Golub and C. F. V. Loan, *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*, 3rd ed. The Johns Hopkins University Press, 10 1996.

[33] A. Papoulis, "Generalized sampling expansion," *IEEE Transactions on Circuits and Systems*, vol. 24, no. 11, pp. 652–654, November 1977.

[34] P. Lowenborg and H. Johansson, "Quantization noise in filter bank analog-to-digital converters," in *IEEE International Symposium on Circuits and Systems*, 2001.

[35] *C8051F120/1/2/3/4/5/6/7 High-Speed Mixed-Signal ISP MCU Family*, Silicon Laboratories, 2003.