A Polyphase Nonlinear Equalization Architecture and Semi-blind Identification Method

Benjamin Miller, Joel Goodman, Matthew Herman
{bamiller, jgoodman, mherman}@ll.mit.edu
MIT Lincoln Laboratory
244 Wood St.
Lexington, MA 02420

Abstract—In this paper, we present an architecture and semi-blind identification method for a polyphase nonlinear equalizer (pNLEQ). Such an equalizer is useful for extending the dynamic range of time-interleaved analog-to-digital converters (ADCs). Our proposed architecture is a polyphase extension to other architectures that partition the Volterra kernel into small nonlinear filters with relatively low computational complexity. Our semi-blind identification technique addresses important practical concerns in the equalizer identification process. We describe our architecture and demonstrate its performance with measured results when applied to a National Semiconductor ADC081000.

I. INTRODUCTION

Linear dynamic range is a limiting factor in wideband receivers, and is frequently dominated by nonlinear behavior in the analog front-end of the analog-to-digital converter (ADC) [1]. As bandwidth increases, high-rate ADCs are commonly implemented as several low-rate ADCs whose outputs are time-interleaved to achieve a higher effective sampling rate. In the interleaving process, additional nonlinear distortions occur that, like the distortions from the front-end, limit the receiver’s dynamic range.

To extend the dynamic range of time-interleaved ADCs (TIADCs), we have developed a polyphase nonlinear equalizer (pNLEQ); a digital signal processor that compensates for these nonlinearities. Like many methods that use polynomial models to invert nonlinear distortions, we use an interleaved polynomial model to invert interleaved nonlinear distortions. Our method is similar to other recent methods (e.g., [2], [3]) that find a small representation of the nonlinear inverse within the space of the highly complex Volterra kernel [4]. The key difference is that this architecture is polyphase, i.e., it periodically moves between different nonlinear responses. This allows us to efficiently mitigate analog non-idealities as well as those that arise due to interleaving. Unlike other methods that address interleaving issues (e.g., [5]–[7]), this approach addresses both linear and nonlinear ADC mismatches as well as polynomial distortions, and also explicitly accounts for memory effects.

This paper is organized as follows. In Section II we describe the pNLEQ architecture and explain the identification process. This includes a sequential estimation technique used to select components for the equalizer, as well as a semi-blind identification technique that necessitates a small restriction on the linear mismatch equalizer during the identification process, but one that does not affect its nonlinear response and can be used to ensure that the pNLEQ is purely nonlinear, i.e., that it does not impart an additional linear response on the ADC output. Section III demonstrates pNLEQ performance with measured data, applying a software-based pNLEQ to the output of a National Semiconductor ADC081000 TIADC. In Section IV we conclude with a brief summary.

II. POLYPHASE NONLINEAR EQUALIZATION

The objective of polyphase nonlinear equalization is to remove polynomial distortions and both linear and nonlinear mismatches between a TIADC’s constituent converters. A high-level model of our desired equalizer is shown in Figure 1, where $x(n)$ is a TIADC output stream and $y(n)$ is the equalized output. Separating the linear mismatch equalizer output $y_{L}(n)$ from the nonlinear mismatch equalizer output $y_{NL}(n)$, as shown in the figure, will be useful for analysis in Section II-A. To mitigate distortions caused by the front end and from interleaving, we use a polyphase version of the Volterra kernel, pruned to reduce complexity.

![Fig. 1. Notional high-level diagram of a polyphase nonlinear equalizer.](image)

The Volterra kernel is a polynomial extension to the linear finite impulse response (FIR) filter and is used to model nonlinear system responses using polynomials with memory. The output of a Volterra kernel system with maximum polynomial order $P$ and memory depth $M$ is given by

$$y(n) = \sum_{p=1}^{P} \sum_{m_1=0}^{M-1} \cdots \sum_{m_p=0}^{M-1} h_p(m_1, \cdots, m_p) \prod_{\ell=1}^{P} x(n - m_\ell).$$
The number of nonredundant coefficients \( h \) is

\[
\sum_{p=1}^{P} \binom{p + M - 1}{p},
\]

which grows combinatorially as the polynomial order increases, and thus the full kernel is not realistically implementable in real time for systems with significant memory and polynomial order.

It is frequently sufficient, however, to use a small subset of the coefficients. Recent nonlinear equalization techniques divide the Volterra kernel into small components and address only small subspaces of the full coefficient space. For example, the horizontal coordinate system (HCS) divides the \( p \)-th order Volterra kernel into components with output given by

\[
y_{\text{HCS}}(n) = \sum_{k=0}^{M-1} h_p(k, \alpha_1, \ldots, \alpha_{p-1}) x(n - k) \prod_{\ell=1}^{p-1} x(n - \alpha_{\ell})
\]

for an input \( x(n) \). With HCS, each component is a linear FIR filter whose output is multiplied by time-delayed values of the equalizer input, as shown in Figure 2(a). This takes advantage of the ability to perform fast one-dimensional convolutions.

We present the polyphase analog of HCS, although the principle can be easily applied to other architectures such as those described in [2], [3], [8]. Using a TIADC, polynomial architectures such as HCS can only equalize distortions common to all constituent ADCs. However, if we replace the FIR filter with a polyphase filter whose output is decimated by a factor of \( N_f \), the number of interleaved ADCs, as shown in Figure 2(b), we can equalize one ADC without regard to the others.

In the figure, \( h_{p,\rho} \) is the \( p \)-th order Volterra kernel for phase \( \rho \), and is split into \( N_f \) separate kernels \( h_{p,\rho,1}, \ldots, h_{p,\rho,N_f} \) for efficient real-time implementation. Using a linear polyphase filter (polynomial order \( p = 1 \)), we can also remove the linear response of this ADC, equalizing nonlinear distortions caused by mismatches in the ADCs’ linear responses. We can therefore create an equalizer for each ADC that accepts data at the TIADC sampling rate, but outputs data at the rate of a constituent ADC. Thus a polyphase HCS component for phase \( \rho \) (out of \( N_f \) total phases) that is analogous to the full-rate HCS component with output given by (1) has output given by

\[
y_{\text{HCS}}(n) = \sum_{k=0}^{M-1} h_{\rho}(k, \alpha_1, \ldots, \alpha_{p-1}) x(n - k) \prod_{\ell=1}^{p-1} x(n - \alpha_{\ell}),
\]

where \( \delta_{N_f}(n) \) is equal to 1 if \( n \equiv 0 \mod N_f \) and is equal to 0 otherwise. By interleaving the outputs of \( N_f \) of these equalizers, one for each ADC, a nonlinear equalizer for polynomial, linear mismatch and polynomial mismatch distortions in the TIADC can be created.

A recent linear mismatch equalization technique uses filters whose outputs are multiplied by sine waves, in effect up-converting the output to the image frequencies [9]. This is useful in that it clearly separates the linear and nonlinear effects of the linear mismatches in the ADC. It is, however, more computationally expensive, since the outputs of \( N_f - 1 \) filters must be computed for each output sample. Using polyphase HCS for linear mismatch equalization (i.e., using interleaved linear decimating filters) requires computing the output of only one filter for each output sample, and it is also possible to remove any linear response (as will be discussed in Section II-A).

To excite the TIADC, we use high-fidelity analog tone generators followed by filters to remove harmonics. This gives us confidence that all nonlinear distortions at the output of the ADC are created by the ADC itself. However, this leaves us semi-blind in that we do not know the true phase and amplitude of the excitation tones. Like in [5], we transform our ADC output and the nonlinear combinations thereof into the frequency domain and remove the frequency locations that contain either noise or the original excitation tones. This makes our training data purely nonlinear and allows us to ignore the linear response of the system.

To identify the coefficient values of the polyphase filters in the HCS components, we solve

\[
\hat{h} = \arg \min_{h} \| W_{p} x + W_{p} X_{h} \|_{2}^{2},
\]

where \( x \) is a column of ADC outputs (the stimulus of our pNLEQ), \( W_{p} \) is a pruned DFT matrix containing only the rows associated with frequencies of interest (those containing nonlinear distortions), and \( X = [X_{1} \cdots X_{J}] \), where \( J \) is the number of polyphase HCS components, is a matrix whose columns are the interleaved polynomial terms associated with each component we use (\( X_{J} \) being the columns associated with the \( J \)-th component). If the \( J \)-th component has polynomial order \( p_{J} \), delays \( \alpha_{1,j}, \ldots, \alpha_{p_{J} - 1,j} \), and \( M_{J} \) taps in its polyphase filter; and equalizes distortions from the \( p \)-th constituent ADC.

A. Semi-Blind Identification

Fig. 2. Full-rate and polyphase NLEQ components.
(of a total of \( N_I \)), then letting
\[
x_j(i) = \left[ x(i) \prod_{k=1}^{p_j-1} x(i-\alpha_{k,j}), \ldots, x(i-M_j+1) \prod_{k=1}^{p_j-1} x(i-\alpha_{k,j}) \right],
\]
the \( r \)th row of \( X_j \) will be \( x_j(r) \) if \( r \equiv \rho_j \mod N_I \) and a row of zeros otherwise. Thus we have
\[
X_j = \begin{bmatrix}
\vdots \\
x_j(iN_I + \rho_j) \\
0 \\
x_j((i+1)N_I + \rho_j) \\
0 \\
\vdots 
\end{bmatrix},
\]
where \( 0 \in \mathbb{R}^{(N_I-1) \times M_j} \) is a matrix of zeros. The solution \( \hat{h} \) to (2) then contains the coefficients of the polyphase filters in our pNLEQ components, and the sum of the outputs of these components with the distortion components of the input to the equalizer (the TIADC output) will be minimized, reducing spurs in the frequency spectrum.

Ignoring the tones is useful, as it removes the need for synchronization as well as any uncertainty about the phase and amplitude of the signal. A problem arises, however, when it is possible to create a purely linear response with the pNLEQ. If it is possible for the pNLEQ to adapt its coefficients such that the aggregate response \( y_{\text{L}}(n) = -x(n) \), then this will solve (2) with no residual error. Specifically, since \( \hat{h} \) is identified with respect to only distortions and not the tones, if it simply multiplies the input by \(-1\) and adds this to the original signal, the distortions will be removed, and the fact that the tones are removed as well is not a concern to the adaptation algorithm.

To remove nonlinear distortions in non-interleaved ADCs, only polynomials with \( p > 1 \) are required, so it is impossible to create the additive inverse of the original signal. Using interleaved polynomials to correct distortions in TIADCs, however, requires using interleaved linear components. If there is a linear component for each phase, this can be adapted to output the additive inverse of the input. For each phase \( \rho \) there is a decimating filter \( h_{\rho} \) for linear mismatch equalization. If for all \( \rho \), \( h_{1,\rho}(0) = -1 \) and \( h_{1,\rho}(k) = 0 \) for \( k \neq 0 \), then the output will be the additive inverse of the input.

It is possible, however, to use the same architecture to remove spurs in the spectrum and use semi-blind identification. Using linear polyphase filters for only \( N_I - 1 \) phases, we can address the same nonlinear behavior of the ADC in a way that does not allow a purely linear response to be created. Consider the linear mismatch equalizer in Figure 3(a). Assuming that this equalizer removes all linear mismatches imparted by the TIADC, then by adding another full-rate (non-polyphase) filter in parallel with this equalizer, no additional nonlinear behavior is imparted. That is, if
\[
y_{\text{L}}(n) = \sum_{\rho=1}^{N_I} \sum_{m=0}^{M-1} h_{1,\rho}(m)x(n-m)\delta_{N_I}(n-\rho),
\]
then \( y_{\text{L}}(n) + \sum_{k=0}^{M-1} g(k)x(n-k) \) will have the same nonlinear response as \( y_{\text{L}}(n) \) since the mismatches are the same.

Instead of adding a full-rate filter in parallel, we can add the same impulse response to each phase’s linear mismatch filter i.e., \( y_{\text{L}}(n) + \sum_{k=0}^{M-1} g(k)x(n-k) \) is equivalent to
\[
\tilde{y}_{\text{L}}(n) = \sum_{\rho=1}^{N_I} \sum_{m=0}^{M-1} [h_{1,\rho}(m) + g(m)]x(n-m)\delta_{N_I}(n-\rho).
\]

Letting \( g = -h_{1,N_I} \), the linear response at phase \( N_I \) becomes 0, resulting in the system shown in Figure 3(b). Since the mismatches are the same, this equalizer has the same nonlinear behavior (i.e., it creates the same spurs) as the one in Figure 3(a), but since it outputs 0 every \( N_I \) samples, it is impossible to create a purely linear response. The response of this equalizer will, however, have a linear component.

After identifying the mismatches, and therefore the nonlinear behavior of the system, the linear response of the system...
can be removed with an additional filter. First note that the linear response of interleaved linear impulse responses is the average of the individual linear responses. The output of the system is given by \( y_n(n) \) in (3). Since \( \delta_{k,n} \) is periodic, it can be represented as the sum of complex exponentials, and we can rewrite (3) as

\[
y_n(n) = \frac{1}{N_f} \sum_{\rho=1}^{N_f} \sum_{m=0}^{M-1} h_{1,\rho}(m) x(n-m) + \sum_{k=0}^{N_f-1} e^{2\pi i k(n-\rho)/N_f}.
\]

Separating out the case of \( k = 0 \), we have

\[
y_{k}(n) = \frac{1}{N_f} \sum_{\rho=1}^{N_f} \sum_{m=0}^{M-1} h_{1,\rho}(m) x(n-m) + \sum_{k=1}^{N_f-1} e^{2\pi i k(n-\rho)/N_f},
\]

(4)

The exponential in (4) all have nonzero digital frequencies, meaning that while the left-hand summand is now the linear combination of time-delayed values of the input \( x(n) \), and therefore represents a linear response, the right-hand summand is the linear combination of time-delayed and frequency-shifted values of \( x(n) \), which are not linearly related to the original signal. Therefore, the linear impulse response of the interleaved system is given by \( h_{\text{avg}}(n) = \sum_{\rho=1}^{N_f} h_{1,\rho}(n)/N_f \), the average of the individual linear responses.

This can be taken into account in the adaptation process to remove the linear response of the linear mismatch equalizer and make the pNLEQ purely nonlinear. Consider the columns of the matrix \( X \) in (2) that correspond to coefficients of the linear mismatch filters, and separate these into groups of columns where each group is associated with a different phase. Each group then forms a matrix \( X_{L,\rho} \), where the \( r \)th row is given by \( [x(r), x(r-1), \ldots, x(r-M+1)] \) if \( r \equiv \rho \) mod \( N_f \) and is a row of zeros otherwise. Without loss of generality, assume that phase \( N_f \) has its linear impulse response forced to zero, so \( X \) does not contain the columns of \( X_{L,N_f} \). We now have a matrix \( [X_{L,1}, \ldots, X_{L,N_f-1}] \) that contains the columns corresponding to the coefficients of the linear mismatch equalization filters.

Now consider a matrix \( X_L \) where the \( r \)th row is equal to \( [x(r), x(r-1), \ldots, x(r-M+1)] \) for all rows. Since the linear response of an interleaved linear system is the average linear response of all phases, this matrix can be used in the adaptation process to remove the linear response of the equalizer. Let the coefficients of the linear mismatch equalization filter for phase \( \rho \) be given by the entries in the vector \( h_{L,\rho} \), and the average of these vectors across all phases be given by \( h_{\text{avg}} = \sum_{\rho=1}^{N_f} h_{L,\rho}/N_f \) (in this case \( h_{L,N_f} \) is a zero vector). Note that

\[
[X_{L,1}, \ldots, X_{L,N_f}] [h_{L,1}^T, \ldots, h_{L,N_f}^T]^T = X_L h_{\text{avg}} = [X_{L,1}, \ldots, X_{L,N_f-1}] [h_{L,1}, \ldots, h_{L,N_f-1}]^T.
\]

Thus by setting the columns of \( X \) represented by \( X_{L,\rho} \) to \( X_L, \rho - X_L \) when identifying the coefficients, we can explicitly take into account the fact that we will remove the linear response after identifying the equalizer with \( N_f - 1 \) linear mismatch filters. In the final equalizer, the coefficients \( h_{L,\rho} \) will be replaced with \( h_{L,\rho} - h_{\text{avg}} \) (including phase \( N_f \)), and the linear mismatch equalizer will have no linear response.

### B. Sequential Estimation

Unlike many polynomial signal processing architectures, HCS does not cover a fixed subset of Volterra kernel coefficients. For example, a memory polynomial [8] considers only coefficients along the main diagonal, i.e., for a \( p \)th-order Volterra kernel \( h_p(m_1, \ldots, m_p) \), the only nonzero coefficients are those where \( m_i = m_j \) for all \( 1 \leq i, j \leq p \). An HCS component, on the other hand, can occupy any horizontal swath of the kernel where \( m_i \) is fixed (at any value) for \( 1 < i \leq p \). It is necessary, therefore, to determine which swaths to use. To identify the HCS components to use in the equalizer, we use an iterative sequential estimation technique similar to orthogonal matching pursuit [10]. A full Volterra kernel is divided into HCS components, and at each iteration the algorithm selects the component that most improves performance, i.e., reduces error in (2) in conjunction with the components already selected. The algorithm continues until either a performance objective or a computational complexity limit is met.

### III. RESULTS

We applied our pNLEQ architecture to the National Semiconductor ADC081000, using the sequential estimation algorithm described in Section II-B to choose polyphase HCS components. Figures 4(a) and 4(b) show the frequency spectrum of the output of each constituent ADC with the same 3-tone stimulus at the input (2 tones are very close in frequency), with power specified in decibels with respect to full scale (dBFS).

Since each ADC samples at a sub-Nyquist rate and there is an ambiguity caused by undersampling, frequency is specified by the discrete bin. Note that the nonlinear distortions differ significantly in each ADC, some distortions being more than 10 dB higher in Channel 1 than in Channel 2. When the two channels are interleaved, the data is sampled at the Nyquist rate and this results in the spectrum shown in Figure 4(c). The frequency axis is now specified in megahertz (MHz) since these now correspond to specific analog frequencies. Note that there are visible nonlinear distortions due to polynomial nonlinearities as well as linear and nonlinear mismatches between the ADCs.

Figure 4(d) demonstrates pNLEQ performance when equalizing this ADC; using linear as well as second-, third- and fourth-order polyphase HCS components. All three types of distortions are significantly reduced by pNLEQ. The equalizer reduces distortions in the TIADC to the same level as full-rate NLEQ can reduce distortions for the constituent ADCs, as shown in Figure 5. Evaluating pNLEQ performance with over 2000 3-tone data sets in a 345 MHz band, we typically achieve an improvement in intermodulation-free dynamic range (IFDR) of over 13 dB, similar to what is shown in the Figure 4(d), while an interleaved memoryless polynomial of the same order achieves an improvement of only 3 dB. Polynomial nonlinearities are strongest, so equalizing linear mismatches alone provides only negligible improvement.

### IV. SUMMARY

In this paper we describe an architecture for mitigating three kinds of nonlinear distortions found in TIADCs: polynomial
distortions, linear mismatch distortions and polynomial mismatch distortions. Our polyphase nonlinear equalizer takes advantage of the polyphase nature of the TIADC response by using (nonlinear) polyphase decimation filters in order to efficiently equalize distortions that vary from sample to sample due to inconsistencies between the TIADC’s constituent converters. To enable semi-blind identification, we have presented an identification method that forces a linear impulse response of 0 on one of the phases of the equalizer during training, but maintains equivalent nonlinear behavior, and thus removes the same spurs. We present measured results using the National Semiconductor ADC081000 showing an IFDR improvement of an order of magnitude greater than an interleaved memoryless polynomial equalizer. Removing linear mismatch distortions alone does not improve IFDR, since polynomial distortions are the limiting factor.

REFERENCES