

A novel nonlinear equalizer for extending the dynamic range of analog-to-digital converters

Xinfa Zhang^{a)} and Feng Yu^{b)}

Department of Instrument Engineering, Zhejiang University,

Hangzhou 310027, China

a) xinfazhangemail@gmail.com

b) osfengyu@zju.edu.cn

Abstract: Extending the dynamic range of analog-to-digital converters (ADCs) is an important issue in their application field. This paper presents a novel nonlinear equalizer based on the cube coefficient subspace (CCS) architecture to mitigate nonlinearities inherent in ADCs. The proposed equalizer is verified with a 14-bit, 200-MSPS ADC injected a two-tone signal. Simulation results show that it achieves a 16.55 dB improvement in dynamic range with a computational complexity of 110 operations per sample (OPS). The improvement using the proposed equalizer is 14.12 dB and 2.67 dB greater than those using two conventional equalizers: the memory polynomial (MP) and horizontal coordinate system (HCS) equalizers, respectively, with the same computational complexity.

Keywords: analog-to-digital converter, nonlinear equalizer, Volterra filter, cube coefficient subspace, QR decomposition

Classification: Integrated circuits

References

- [1] G. M. Raz and C. P. Chan: U.S. Patent 7609759 (2009).
- [2] P. Nikaeem and B. Murmann: *IEEE J. Sel. Topics Signal Process.* **3** (2009) 499. DOI:10.1109/JSTSP.2009.2020575
- [3] M. Schetzen: *The Volterra and Wiener Theories of Nonlinear System* (Wiley, New York, 1980) 1st ed. 8.
- [4] J. H. M. Goulart and P. M. S. Burt: *IEEE Signal Process. Lett.* **19** (2012) 135. DOI:10.1109/LSP.2011.2182342
- [5] L. Ding, G. T. Zhou, D. R. Morgan, Z. X. Ma, J. S. Kenney, J. Kim and C. R. Giardina: *IEEE Trans. Commun.* **52** (2004) 159. DOI:10.1109/TCOMM.2003.822188
- [6] B. Miller, J. Goodman and M. Herman: 42nd Asilomar Conference on Signals, Systems and Computers (2008) 593. DOI:10.1109/ACSSC.2008.5074474
- [7] J. Goodman, B. Miller, M. Herman, M. Vai and P. Monticciolo: International WD&D Conf. (2009) 224. DOI:10.1109/WDDC.2009.4800349
- [8] J. Goodman, B. Miller, M. Herman, G. M. Raz and J. Jackson: *IEEE J. Sel. Topics Signal Process.* **3** (2009) 362. DOI:10.1109/JSTSP.2009.2020243

- [9] M. Herman, B. Miller and J. Goodman: 42nd Asilomar Conference on Signals, Systems and Computers (2008) 1857. DOI:10.1109/ACSSC.2008.5074750
 [10] J. Goodman, B. Miller and M. Herman: U.S. Patent 7808315 (2010).
 [11] J. Goodman, B. Miller, J. Vian, A. Bolstad, J. Kalyanam and M. Herman: MILCOM on Waveforms and Signal Process. Tack (2010) 2113.

1 Introduction

The dynamic range of ADCs dominantly suffers from nonlinearities (e.g., harmonics, intermodulations, spurs, etc.) [1] inherent in the converter core and the sampling front-end of ADCs [2]. One possibility to diminish these nonlinearities and extend the potential dynamic range is to use a nonlinear equalizer based on Volterra filters. Volterra filters are a general technique for representing and compensating arbitrary nonlinearities with memory [3]. Nevertheless, a major drawback of this technique is that the number of coefficients in the Volterra filter grows exponentially with the order and the memory of the ADC nonlinearity behavior model [4]. To overcome the highly computational complexity, the memory polynomial (MP) [5] and horizontal coordinate system (HCS) [6, 7, 8] architectures for truncated Volterra filters have been proposed. Both MP and HCS consist of one-dimensional polynomial subkernels which are selected from the full Volterra kernel. However, compared to one-dimensional polynomial filters, multidimensional polynomial filters more easily reduce asymmetric nonlinearities, and consist of fewer subkernels [9]. The cube coefficient subspace (CCS) architecture proposed by M. Herman et al. [9] is composed of small multidimensional hypercubes within the whole Volterra kernel coefficient spaces. In this paper, a novel nonlinear equalizer based on CCS for post-corrections of ADCs is proposed. Simulation results show that the proposed equalizer yields larger dynamic range of a 14-bit, 200-MSPS ADC than the conventional equalizer based on MP or HCS with the same computational complexity.

2 Cube coefficient subspace

The CCS is a variant of the full Volterra filter to model the ADC nonlinearity behavior. Hence, it is necessary to introduce the Volterra series which is the mathematics base of Volterra filters. The Volterra series in discrete time domain describes the relation between the input $u(n)$ and output $y(n)$ as

$$y(n) = \sum_{p=1}^P y_p(n), \quad (1)$$

where

$$y_p(n) = \sum_{m_1=0}^{M-1} \cdots \sum_{m_p=0}^{M-1} h_p(m_1, \cdots, m_p) \prod_{l=1}^p u(n - m_l). \quad (2)$$

Here, p is the polynomial order, P is the maximum polynomial order, M is the memory length, $h_p(m_1, \cdots, m_p)$ are the full p th-order Volterra kernel coefficients (also known as the full p th-order Volterra kernel) [9], and m_l is the memory variable

in the l th-dimension of the full p th-order kernel $h_p(m_1, \dots, m_p)$. Each partial output $y_p(n)$ is the p -dimensional convolution of $u(n)$ with $h_p(m_1, \dots, m_p)$.

The number of coefficients $h_p(m_1, \dots, m_p)$ is M^p [5], which grows exponentially with the order p . The number is so large that the full Volterra filter is unfeasible in real time systems. Therefore, most practical Volterra filters (e.g., MP, HCS, CCS, etc.) consist of several subkernels within the full Volterra kernel. If $h_p(m_1, \dots, m_p) = 0$, except along the diagonal $m_1 = m_2 = \dots = m_p$, then the full Volterra filter becomes a one-dimensional polynomial filter MP given by [5]

$$y^{MP}(n) = \sum_{p=1}^P \sum_{m=0}^{M-1} h_p(m, \dots, m) u^p(n - m). \quad (3)$$

Here, $u^p(n - m)$ is the p th power of the input $u(n)$ with the memory variable m , and $h_p(m, \dots, m)$ is the p th-order MP kernel which is a one-dimensional subkernel within the full p th-order kernel $h_p(m_1, \dots, m_p)$. The memory variables m_l for $l = 1, \dots, p$ in $h_p(m_1, \dots, m_p)$ are independent of each other, while the ones in $h_p(m, \dots, m)$ are equal to each other. As shown in Fig. 1, the yellow cube is the full 3rd-order Volterra kernel with memory length M . The purple line segment is the 3rd-order MP kernel, which lies along the main diagonal of yellow cube [9].

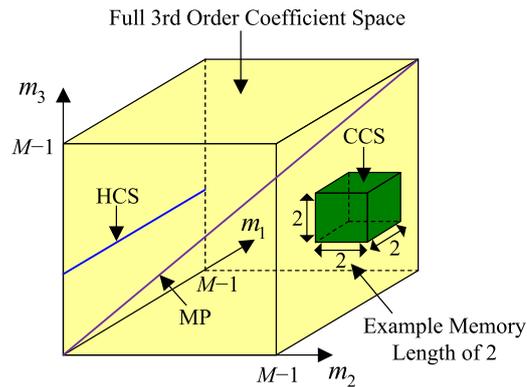


Fig. 1. Geometry of the 3rd-order MP, HCS, and CCS kernels.

Another one-dimensional polynomial filter HCS is given by [7, 8]

$$y_p(n) = \sum_{\alpha_2=i_2}^{M+i_2-1} \dots \sum_{\alpha_p=i_p}^{M+i_p-1} y_p^{HCS}(n, \alpha_2, \dots, \alpha_p) \quad (4)$$

where

$$y_p^{HCS}(n, \alpha_2, \dots, \alpha_p) = \underbrace{\sum_{m=i}^{M+i-1} h_p(m, \alpha_2, \dots, \alpha_p) u(n - m)}_{h_p(n; \alpha_2, \dots, \alpha_p) * u(n)} \prod_{l=2}^p u(n - \alpha_l). \quad (5)$$

Here, the symbol $*$ denotes the convolution operator, $h_p(m, \alpha_2, \dots, \alpha_p)$ is the p th-order HCS kernel, the i_k for $k = 1, \dots, p$ are used to center the data over the taps of the filter, and the α_l for $l = 2, \dots, p$ are fixed integer delays. The HCS is a one-dimensional convolution multiplied by the product of time-delayed values of input $u(n)$ [7, 8]. As shown in Fig. 1, the blue line segment is the 3rd-order HCS kernel.

It only picks coefficients along a single horizontal (m) paralleling to the m_1 axis, while the other dimensions of the 3rd-order kernel (α_2, α_3) stay fixed.

In addition, there is one more truncated Volterra filter called CCS, which is a multidimensional polynomial filter. The output of a p th-order, d -dimensional ($d \leq p$) CCS corresponding to a d -dimensional subkernel within the full p th-order Volterra kernel is given by [9, 10, 11]

$$y_{(p,d)}^{CCS}(n) = \sum_{m_1=0}^{M-1} \cdots \sum_{m_d=0}^{M-1} h_p(m_1, \cdots, m_d) \prod_{l=1}^d u(n - m_l - \alpha_l) \prod_{s=d+1}^p u(n - \alpha_s). \quad (6)$$

Here, $h_p(m_1, \cdots, m_d)$ is the p th-order, d -dimensional CCS kernel, and α_t for $t = 1, \cdots, p$ are fixed integer delays that determine the CCS kernel position in the full Volterra kernel space. The CCS breaks the full Volterra kernel up into small hypercubes of arbitrary dimension and polynomial order [11]. As shown in Fig. 1, the green cube is the 3rd-order, 3-dimensional CCS kernel with memory length 2.

The computational requirements for the p th-order MP, HCS and CCS with memory length M are shown in Table I. It seems that the amount of CCS computational loads is larger than MP or HCS as the dimension d increases. However, for addressing the localization of nonlinearities, using several multi-dimensional CCS kernels can adapt coefficients in arbitrary dimensions more easily [10].

Table I. The computational requirements for the p th-order MP, HCS and CCS

Architecture	Number of coefficients	Number of multipliers	Number of adders
MP	M	$[M(\lfloor \log_2 p \rfloor + 1), M(2\lfloor \log_2 p \rfloor + 1)]^*$	$M - 1$
HCS	M	$M + p - 1$	$M - 1$
CCS	M^d	$p - d + \sum_{l=1}^d M^l$	$M^d - 1$

*The symbol $\lfloor \cdot \rfloor$ denotes the floor operation.

3 Constructing the proposed equalizer

In this section, the method of constructing the proposed nonlinear equalizer based on CCS is discussed firstly. Then we present the algorithm of selecting an optimal set of CCS kernels, which is a key factor in constructing an efficient equalizer.

3.1 Constructing method

3.1.1 Recasting the CCS representation in matrix-vector form

To begin our discussion of construction method, we first recast Eq. (2) in vector form as

$$y_p(n) = \mathbf{u}_p^T(n) \mathbf{h}_p, \quad (7)$$

where $(\cdot)^T$ denotes the transposition operator. The p th-order input products $\mathbf{u}_p(n)$ and the full p th-order kernel coefficients \mathbf{h}_p both are $M^p \times 1$ vectors,

$$\mathbf{u}_p(n) = [u^p(n), u(n-1)u^{p-1}(n), \dots, u^p(n-M+1)]^T, \quad (8)$$

and

$$\mathbf{h}_p = [h_p(0, 0, \dots, 0), h_p(1, 0, \dots, 0), \dots, h_p(M-1, M-1, \dots, M-1)]^T. \quad (9)$$

Then, we recast Eq. (1) in vector form as

$$y(n) = \mathbf{u}^T(n)\mathbf{h} \quad (10)$$

Here, all input products $\mathbf{u}(n)$ from 1 to P order and the corresponding coefficients \mathbf{h} both are $(\sum_{p=1}^P M^p) \times 1$ vectors,

$$\mathbf{u}(n) = [\mathbf{u}_1^T(n), \mathbf{u}_2^T(n), \dots, \mathbf{u}_P^T(n)]^T, \quad (11)$$

and

$$\mathbf{h} = [\mathbf{h}_1^T, \mathbf{h}_2^T, \dots, \mathbf{h}_P^T]^T. \quad (12)$$

Similarly, the CCS expressed as Eq. (6) can also be recast in vector form as

$$y_{(p,d)}^{CCS}(n) = [\mathbf{u}_{(p,d)}^{CCS}(n)]^T \mathbf{h}_{(p,d)}^{CCS}, \quad (13)$$

where the $M^d \times 1$ vector $\mathbf{u}_{(p,d)}^{CCS}(n)$ can be represented by

$$\mathbf{u}_{(p,d)}^{CCS}(n) = \left[\prod_{l=1}^d u(n - \alpha_l) \prod_{s=d+1}^p u(n - \alpha_s), \dots, \prod_{l=1}^d u(n - M + 1 - \alpha_l) \prod_{s=d+1}^p u(n - \alpha_s) \right]^T \quad (14)$$

and the $M^d \times 1$ vector $\mathbf{h}_{(p,d)}^{CCS}$ contains the corresponding coefficients. Using N samples of $u(n)$, we define the proposed equalizer input $u(n)$ in the $N \times 1$ vector form as

$$\tilde{\mathbf{u}}(n) = [u(n), u(n-1), \dots, u(n-N+1)]^T. \quad (15)$$

Then, Eq. (13) can be rewritten in matrix-vector form as

$$\mathbf{y}_{(p,d)}^{CCS} = \mathbf{U}_{(p,d)}^{CCS} \mathbf{h}_{(p,d)}^{CCS} \quad (16)$$

with

$$\mathbf{y}_{(p,d)}^{CCS} = [y_{(p,d)}^{CCS}(n), y_{(p,d)}^{CCS}(n-1), \dots, y_{(p,d)}^{CCS}(n-N+1)]^T \quad (17)$$

and

$$\mathbf{U}_{(p,d)}^{CCS} = [\mathbf{u}_{(p,d)}^{CCS}(n), \mathbf{u}_{(p,d)}^{CCS}(n-1), \dots, \mathbf{u}_{(p,d)}^{CCS}(n-N+1)]^T, \quad (18)$$

where $\mathbf{y}_{(p,d)}^{CCS}$ is an $N \times 1$ vector and $\mathbf{U}_{(p,d)}^{CCS}$ is an $N \times M^d$ matrix. According to the Eqs. (14), (15) and (18), we can recast the matrix $\mathbf{U}_{(p,d)}^{CCS}$ as

$$\mathbf{U}_{(p,d)}^{CCS} = [\tilde{\mathbf{u}}(n - \alpha_1) \circ \dots \circ \tilde{\mathbf{u}}(n - \alpha_d) \circ \tilde{\mathbf{u}}(n - \alpha_{d+1}) \circ \dots \circ \tilde{\mathbf{u}}(n - \alpha_p), \dots, \tilde{\mathbf{u}}(n - M + 1 - \alpha_1) \circ \dots \circ \tilde{\mathbf{u}}(n - M + 1 - \alpha_d) \circ \tilde{\mathbf{u}}(n - \alpha_{d+1}) \circ \dots \circ \tilde{\mathbf{u}}(n - \alpha_p)]^T, \quad (19)$$

where the symbol \circ denotes the Hadamard product. We define the matrix $\mathbf{U}_{(p,d)}^{CCS}$ as a CCS processing element (PE). In Eq. (19), the order p and dimension d fulfill $1 \leq d \leq p \leq P$, and delay values $\alpha_t (t = 1, \dots, p)$ range from G to K . The values

of G and K are determined by the computational complexity and the effect on mitigating the nonlinearities inherent in ADCs. To obtain better effect on mitigating the nonlinearities, the choice of G and K should be larger at the cost of more computational complexity. Generally, the values of G and K could be 0 and $\lfloor M/2 \rfloor$ to make the balance. Regard to their differences in p , d and α_t , CCS PEs compose a user-defined candidate PE set ψ , as shown in Table II. The number of PEs can be adjusted according to the computational complexity.

Table II. The user-defined candidate processing element set ψ

p	d	PEs	The CCS PE expression in matrix form*
1	1	$\mathbf{U}_{(1,1)}^{CCS}$	$[\tilde{\mathbf{u}}(n-0-\alpha_1), \tilde{\mathbf{u}}(n-1-\alpha_1), \dots, \tilde{\mathbf{u}}(n-M+1-\alpha_1)]_{N \times M}$
2	1	$\mathbf{U}_{(2,1)}^{CCS}$	$[\tilde{\mathbf{u}}(n-0-\alpha_1) \circ \tilde{\mathbf{u}}(n-\alpha_2), \dots, \tilde{\mathbf{u}}(n-M+1-\alpha_1) \circ \tilde{\mathbf{u}}(n-\alpha_2)]_{N \times M}$
	2	$\mathbf{U}_{(2,2)}^{CCS}$	$[\tilde{\mathbf{u}}(n-0-\alpha_1) \circ \tilde{\mathbf{u}}(n-0-\alpha_2), \dots, \tilde{\mathbf{u}}(n-M+1-\alpha_1) \circ \tilde{\mathbf{u}}(n-M+1-\alpha_2)]_{N \times M^2}$
3	1	$\mathbf{U}_{(3,1)}^{CCS}$	$[\tilde{\mathbf{u}}(n-0-\alpha_1) \circ \tilde{\mathbf{u}}(n-\alpha_2) \circ \tilde{\mathbf{u}}(n-\alpha_3), \dots, \tilde{\mathbf{u}}(n-M+1-\alpha_1) \circ \tilde{\mathbf{u}}(n-\alpha_2) \circ \tilde{\mathbf{u}}(n-\alpha_3)]_{N \times M}$
	2	$\mathbf{U}_{(3,2)}^{CCS}$	$[\tilde{\mathbf{u}}(n-0-\alpha_1) \circ \tilde{\mathbf{u}}(n-0-\alpha_2) \circ \tilde{\mathbf{u}}(n-\alpha_3), \dots, \tilde{\mathbf{u}}(n-M+1-\alpha_1) \circ \tilde{\mathbf{u}}(n-M+1-\alpha_2) \circ \tilde{\mathbf{u}}(n-\alpha_3)]_{N \times M^2}$
	3	$\mathbf{U}_{(3,3)}^{CCS}$	$[\tilde{\mathbf{u}}(n-0-\alpha_1) \circ \tilde{\mathbf{u}}(n-0-\alpha_2) \circ \tilde{\mathbf{u}}(n-0-\alpha_3), \dots, \tilde{\mathbf{u}}(n-M+1-\alpha_1) \circ \tilde{\mathbf{u}}(n-M+1-\alpha_2) \circ \tilde{\mathbf{u}}(n-M+1-\alpha_3)]_{N \times M^3}$
...

* $1 \leq d \leq p \leq P$, $G \leq \alpha_t \leq K$ ($t = 1, \dots, p$).

3.1.2 Deriving the equalizer coefficients

We inject a two-tone signal into the ADC, which is combined with two sine-waves whose frequencies are f_1 and f_2 . Then the ADC output $\tilde{\mathbf{u}}$ is transformed as the frequency domain representation $\tilde{\mathbf{y}}_{FFT}$ by the fast Fourier transform (FFT): $\tilde{\mathbf{u}} \xrightarrow{FFT} \tilde{\mathbf{y}}_{FFT}$. The amplitudes of $\tilde{\mathbf{y}}_{FFT}$ at f_1 and f_2 are both changed to the minimum value among all frequencies, and the modification of the vector $\tilde{\mathbf{y}}_{FFT}$ is written as $\hat{\mathbf{y}}_{FFT}$. And then $\hat{\mathbf{y}}_{FFT}$ is reformed as the time domain representation $\hat{\mathbf{y}}$ by the inverse FFT (IFFT): $\hat{\mathbf{y}}_{FFT} \xrightarrow{IFFT} \hat{\mathbf{y}}$. We obtain the approximate linear and nonlinear modality $\hat{\mathbf{y}}$ in the ADC to derive the equalizer coefficients.

Next, we select CCS PEs from ψ and derive the corresponding coefficients. The algorithm of selecting an optimal CCS PE set η is illustrated in section 3.2. We assume that $J-1$ PEs ($\mathbf{U}_{(p_1, d_1)}^{CCS_1}, \dots, \mathbf{U}_{(p_{J-1}, d_{J-1})}^{CCS_{J-1}}$) have been selected and identified as the optimal PEs. And then, the PE $\mathbf{U}_{(p_J, d_J)}^{CCS_J}$ is selected. We combine $\mathbf{U}_{(p_J, d_J)}^{CCS_J}$ with the previous $J-1$ PEs to identify whether it is optimal. The method of identification is based on comparing the residues of the least squares solution, which is detailed in section 3.1.3. Using these CCS PEs, the modality $\hat{\mathbf{y}}$ is given by

$$\hat{\mathbf{y}} = \hat{\mathbf{U}} \hat{\mathbf{h}}. \quad (20)$$

Here, $\hat{\mathbf{U}}$ is an $N \times \sum_{j=1}^J M^{d_j}$ matrix,

$$\hat{\mathbf{U}} = [\mathbf{U}_{(p_1, d_1)}^{CCS_1}, \mathbf{U}_{(p_2, d_2)}^{CCS_2}, \dots, \mathbf{U}_{(p_J, d_J)}^{CCS_J}], \quad (21)$$

and $\hat{\mathbf{h}}$ is a $(\sum_{j=1}^J M^{d_j}) \times 1$ vector,

$$\hat{\mathbf{h}} = \left[\left(\mathbf{h}_{(p_1, d_1)}^{CCS_1} \right)^T, \left(\mathbf{h}_{(p_2, d_2)}^{CCS_2} \right)^T, \dots, \left(\mathbf{h}_{(p_J, d_J)}^{CCS_J} \right)^T \right]^T. \quad (22)$$

The number of coefficients $\hat{\mathbf{h}}$ is $\sum_{j=1}^J M^{d_j}$, which is smaller than N . The method of calculating the coefficients $\hat{\mathbf{h}}$ is the least-squares solution. In order to avoid the ill-conditioned problem of normal equation, we use QR decomposition to approximate the least squares solution. The matrix $\hat{\mathbf{U}}$ is decomposed as

$$\hat{\mathbf{U}} = \mathbf{Q}\mathbf{R}, \quad (23)$$

where \mathbf{Q} is an orthogonal matrix and

$$\mathbf{Q}^{-1} = \mathbf{Q}^T. \quad (24)$$

The combination of Eqs. (20), (23) and (24) is given by

$$\mathbf{R}\hat{\mathbf{h}} = \mathbf{Q}^T\hat{\mathbf{y}}. \quad (25)$$

In Eq. (25), we can easily obtain the least squares solution $\hat{\mathbf{h}}_{least-squares}$ of $\hat{\mathbf{h}}$.

3.1.3 Identifying the equalizer architecture

Although we get the least squares solution $\hat{\mathbf{h}}_{least-squares}$, we cannot arbitrarily infer that the recently selected PE $\mathbf{U}_{(p_j, d_j)}^{CCS_j}$ is optimal. We must identify which CCS PE selected in section 3.1.2 is optimal. We first present the residue of the solution $\hat{\mathbf{h}}_{least-squares}$ as

$$\mathbf{r} = \hat{\mathbf{y}} - \hat{\mathbf{U}}\hat{\mathbf{h}}_{least-squares}. \quad (26)$$

The identification standard is that the CCS PE whose $\|\mathbf{r}\|_2$ is minimum among all candidate CCS PEs $\mathbf{U}_{(p_j, d_j)}^{CCS_j}$ (of heterogeneous delay values α_i), and smaller than the previous minimum $\|\mathbf{r}\|_2$ among all candidate CCS PEs $\mathbf{U}_{(p_{j-1}, d_{j-1})}^{CCS_{j-1}}$ (of heterogeneous delay values α_i) is optimal. We repeat the CCS PE selection in section 3.1.2 until we find the optimal CCS PE set η . After obtaining the set η , the associated matrix $\hat{\mathbf{U}}_{optimal}$ and coefficients $\hat{\mathbf{h}}_{optimal}$ based on the selection algorithm, we identify $\hat{\mathbf{U}}_{optimal}\hat{\mathbf{h}}_{optimal}$ as the main body of the equalizer. It is the estimation of the ADC linearity and nonlinearity behavior model. The equalizer output \mathbf{y}_{eq} is given by

$$\mathbf{y}_{eq} = \tilde{\mathbf{y}} - \hat{\mathbf{U}}_{optimal}\hat{\mathbf{h}}_{optimal}, \quad (27)$$

where $\tilde{\mathbf{y}}$ is an appropriately delayed version of the ADC output $\tilde{\mathbf{u}}$, as shown in section 3.1.2. In the end, we get the equalization result \mathbf{y}_{eq} of the ADC.

3.2 Selecting algorithm

Like most articles [6, 8], the computational complexity is defined as operations per sample (OPS). The total number of operations is the sum of the number of adders and multipliers. We set the ceiling number of the OPS of the equalizer complexity as W . The objective is to select an optimal CCS PEs set η from the candidate PEs set ψ which is used to construct an equalizer achieving the highest performance with a given computational complexity. As described in algorithm, we start with an empty set of PEs. Then we select one candidate PE from the set ψ in each selection and calculate the each 2-norm of the corresponding residue. The PE with one-dimension and one-order is selected initially. And then we successively select PEs with

one-dimension, whose order increases by one at a time. As the order of the selected PE exceeds P , we turn to select PEs with two-dimension and two-order. And so on, we repeat the previous selection until the optimal PEs set η is selected. With regard to heterogeneous delay values, PEs with the same dimension and order are different from each other. We select the PE with the minimum 2-norm of the residue from them and add it into the set ζ which is already selected. We terminate the algorithm when the number of the OPS of the equalizer complexity exceeds W .

Algorithm: The selecting algorithm for the optimal CCS PE set η

Require: $1 \leq d_j \leq p_j \leq P (j = 1, \dots, J-1, J, \dots), G \leq \alpha_t \leq K (t = 1, \dots, P)$

$$\sum M^{d_j} < N$$

```

1:  $\eta \leftarrow \emptyset, \zeta \leftarrow \emptyset, j \leftarrow 1, d_j \leftarrow 1, p_j \leftarrow 1$ ;
2: while the number of OPS is smaller than  $W$  do
3:    $\psi \leftarrow \psi \setminus \{\mathbf{U}_{(p_j, d_j)}^{CCS_j}\}$ ;
4:   for  $\alpha_t = G$  to  $K$  ( $t = 1, \dots, p_j$ ) do
5:      $\zeta \leftarrow \zeta \cup \{\mathbf{U}_{(p_j, d_j, \alpha_t)}^{CCS_j}\}$ ;
6:     derive the least squares solution  $\hat{\mathbf{h}}_{least-squares}$  of the coefficients  $\hat{\mathbf{h}}$ 
       and the 2-norm of the corresponding residue  $\mathbf{r}(\alpha_t)$ ;
7:      $\zeta \leftarrow \zeta \setminus \{\mathbf{U}_{(p_j, d_j, \alpha_t)}^{CCS_j}\}$ ;
8:   end for
9:   find the minimum  $\|\mathbf{r}(\alpha_t)\|_2$  and the corresponding CCS PE
        $\mathbf{U}_{(p_j, d_j, \alpha_t)}^{CCS_j-optimal}$ , which is a candidate optimal PE;
10:  if the current minimum  $\|\mathbf{r}(\alpha_t)\|_2$  is smaller than the previous then
11:     $\zeta \leftarrow \zeta \cup \{\mathbf{U}_{(p_j, d_j, \alpha_t)}^{CCS_j-optimal}\}, j \leftarrow j + 1$ ;
12:  end if
13:   $p_j \leftarrow p_j + 1$ ;
14:  if  $p_j > P$  then
15:     $d_j \leftarrow d_j + 1, p_j \leftarrow d_j$ ;
16:  end if
17: end while
18:  $\eta \leftarrow \zeta$ 

```

As we known, both linearities and nonlinearities exist in the output of the ADC. Linearities can be easily mitigated by one dimension CCS PE and should be mitigated firstly. Otherwise, they can mix with nonlinearities and block our path to mitigate nonlinearities when we let them go unchecked. If we inverse the sequence of selection, the computational burden becomes heavier for the same performance.

4 Simulation results

4.1 Simulation environments

We applied the proposed equalizer to a 14-bit 200-MSPS ADC in order to evaluate its performance. As shown in Fig. 2, the analog outputs of two vector signal generators were combined and injected into the ADC. A field programmable gate array (FPGA) captured and stored data from the ADC. The ChipScope Pro Analyzer tool transferred these data from the FPGA to the disk of a computer.

Then we constructed the equalizer with the MATLAB, and injected data into the equalizer for simulation. Specifically, the evaluation of the equalizer is performed offline.

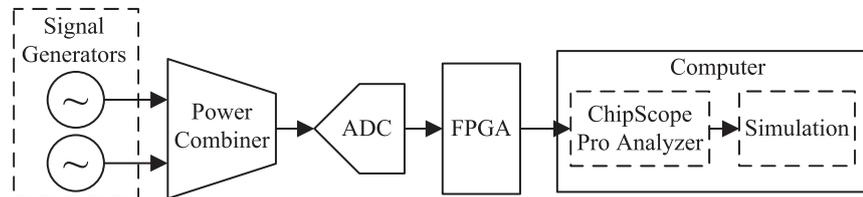


Fig. 2. Block diagram of the ADC data acquisition

4.2 Performance comparison

We injected a two-tone signal into a 14-bit ADC sampling at f_s (200 MSPS). The frequency of one tone f_1 is 80.00 MHz and another tone f_2 is 82.00 MHz. As shown in Fig. 3(a), the dynamic range of the ADC without equalization is -59.57 dBFS. While in Fig. 3(b), after equalization using the proposed equalizer based on CCS, the dynamic range is -76.12 dBFS. It illustrates that the proposed equalizer achieves a 16.55 dB improvement. Furthermore, as described in section 1, non-

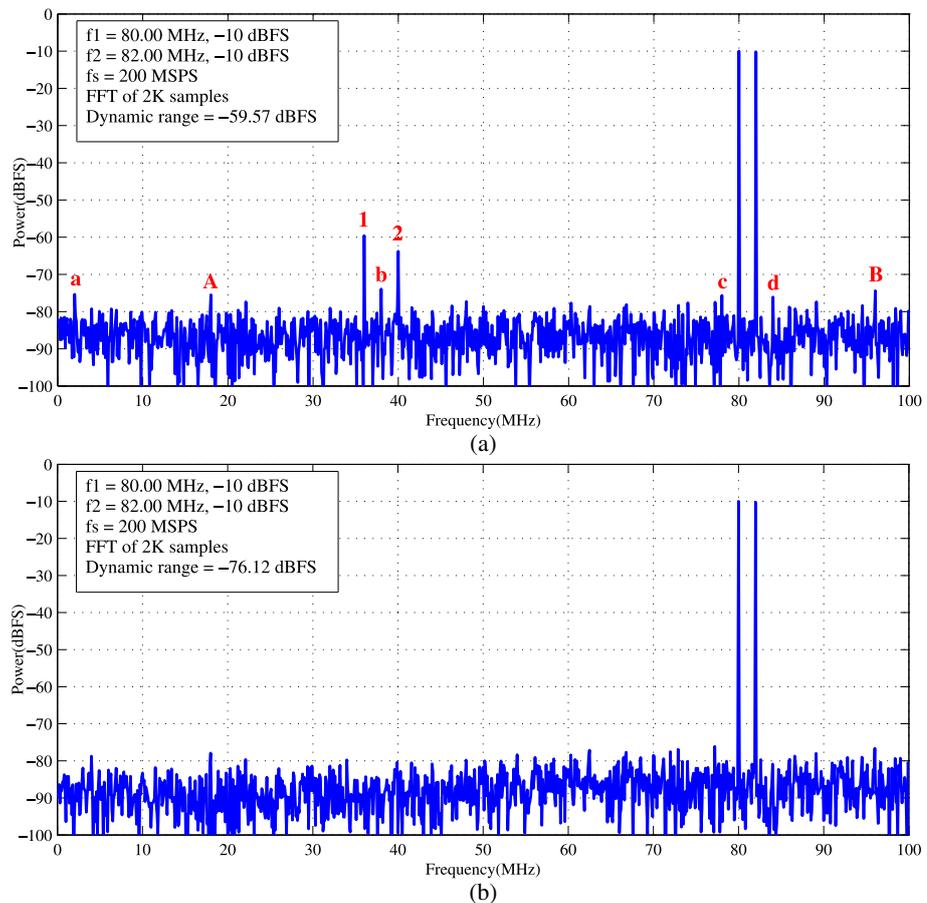


Fig. 3. (a) Before equalization. (b) After equalization using the proposed equalizer based on CCS.

linearities of an ADC include harmonics, intermodulations and spurs. As shown in Fig. 3(a), all kinds of nonlinearities can be clearly distinguished by their locations in the frequency spectrum. Table III is the summary of nonlinearities in the ADC. Fig. 3 demonstrates that the proposed equalizer can efficiently mitigate harmonics, intermodulations and spurs.

Table III. Nonlinearities summary

Type	Location in the frequency spectrum (MHz)	
Harmonics	Label 1: $f_s - 2f_2 = 36.00$	Label 2: $f_s - 2f_1 = 40.00$
Intermodulations	Label a: $f_2 - f_1 = 2.00$	Label b: $f_s - (f_1 + f_2) = 38.00$
	Label c: $2f_1 - f_2 = 78.00$	Label d: $2f_2 - f_1 = 84.00$
Spurs	Label A: $0.5f_s - f_2 = 18.00$	Label B: $-0.75f_s + 3f_2 = 96.00$

For comparison, we also equalized the ADC using two conventional equalizers: the MP and HCS equalizers. As shown in Fig. 4, after equalization using the MP and HCS equalizers, respectively, the dynamic ranges of the ADC are -62.00 dBFS and -73.45 dBFS. Two conventional equalizers only achieve 2.43 dB and 13.88 dB improvements, respectively.

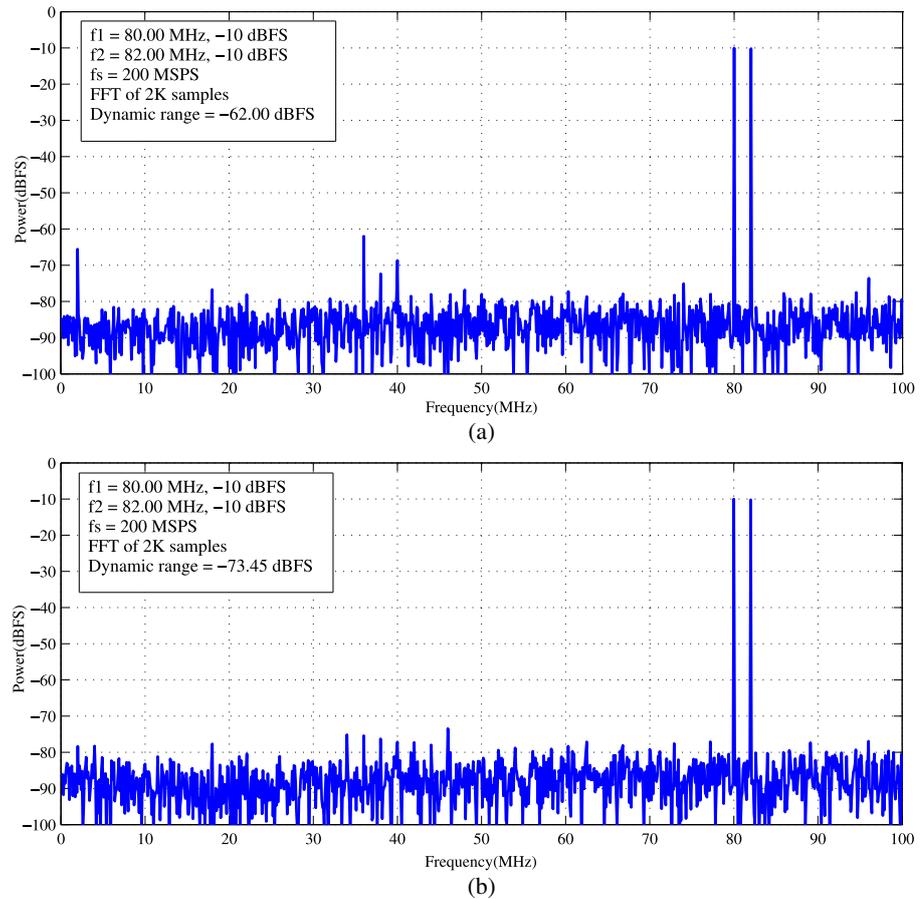


Fig. 4. (a) After equalization using the conventional equalizer based on MP. (b) After equalization using the conventional equalizer based on HCS.

The comparison of performance and computational complexity among three equalizers is shown in Table IV. The optimal PE set of the CCS equalizer includes 4 one-dimension CCS PEs and 2 two-dimension CCS PEs. The polynomial order of PEs ranges from 1 to 4. Each PE has memory length 4 in each dimension and its delay values range from 0 to 2. Thus, the values of G and K are 0 and 2, respectively. The MP equalizer consists of 8 first- through fourth-order PEs and each PE has memory length 4. The HCS equalizer consists of 11 first- through fourth-order PEs, their delay values vary from 0 to 2 and each PE has memory length 4. Table IV shows that the dynamic range improvement using the CCS equalizer is 14.12 dB and 2.67 dB greater than those using MP and HCS equalizers, respectively, with the same computational complexity.

Table IV. The comparison of performance and computational complexity

Equalizer type	Dynamic range improvement (dB)	Number of coefficients	Computational complexity		
			Number of multipliers	Number of adders	Operations per sample
MP	2.43	32	80	31	111
HCS	13.88	44	67	43	110
CCS	16.55	48	63	47	110

The values of the proposed equalizer coefficients are shown in Fig. 5.

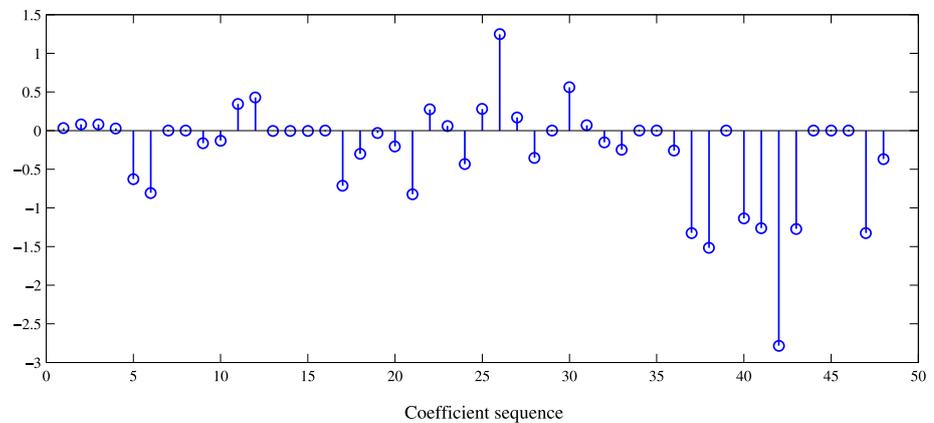


Fig. 5. The values of the proposed equalizer coefficients

5 Conclusion

In this paper, we proposed a novel nonlinear equalizer based on CCS to extend the dynamic range of ADCs. Our equalizer is verified with a 14-bit, 200-MSPS ADC injected a two-tone signal. Simulation results show that it can efficiently mitigate nonlinearities (harmonics, intermodulations and spurs) inherent in the ADC. It achieves a 16.55 dB improvement in dynamic range with a computational complexity of 110 OPS. The improvement using our equalizer is 14.12 dB and 2.67 dB greater than those using two conventional equalizers: the MP and HCS equalizers, respectively, with the same computational complexity.