

# A Min-Max Optimization Framework for Designing $\Sigma\Delta$ Learners: Theory and Hardware

Amit Gore and Shantanu Chakrabartty *Member, IEEE*

**Abstract**—In this paper, we present a framework for constructing  $\Sigma\Delta$  learning algorithms and hardware that can identify and track low-dimensional manifolds embedded in a high-dimensional analog signal space. At the core of the proposed approach is a min-max stochastic optimization of a regularized cost function that combines machine learning with  $\Sigma\Delta$  modulation. As a result, the algorithm not only produces a quantized sequence of the transformed analog signals but also a quantized representation of the transform itself. The framework is generic and can be extended to higher-order  $\Sigma\Delta$  modulators and for different signal transformations. In this paper, the  $\Sigma\Delta$  learning is demonstrated for identifying linear compression manifolds which can eliminate redundant analog-to-digital conversion (ADC) paths. This improves the energy efficiency of the proposed architecture compared to a conventional multi-channel data acquisition system. Measured results from a four channel prototype fabricated in a  $0.5\mu\text{m}$  CMOS process has been used to verify the energy efficiency of the  $\Sigma\Delta$  learner and to demonstrate its real-time adaptation capabilities that are consistent with the theoretical and simulated results. One of the salient features of  $\Sigma\Delta$  learning is its self-calibration property whereby the performance remains unchanged even in the presence of computational artifacts (mismatch and non-linearities). This property makes the proposed architecture ideal for implementing practical high-dimensional analog-to-digital converters.

**Index Terms**—Manifold learning, High-dimensional signal processing,  $\Sigma\Delta$  conversion, signal de-correlation, analog-to-digital conversion, Multi-channel ADC.

## I. INTRODUCTION

Advances in miniaturization are enabling integration of an ever increasing number of recording elements within a single device. Examples of such high-density sensors range from microelectrode arrays used in biomedical applications [1], [2], [3] to microphone arrays used in acoustic sensing [4], [5]. Typically, the multi-channel analog signals acquired by these recording arrays lie in a high-dimensional space and the key challenge lies in designing adaptive analog-to-digital conversion (ADC) algorithms that exploit the topological properties of high-dimensional space. This is particularly relevant, since the conventional wisdom of three-dimensional Euclidean geometry, which is at the heart of many existing ADCs, can not be directly applied to higher dimensions. To illustrate this, consider a simple geometric example comprising of two

Copyright (c) 2009 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org. This work was performed when Amit Gore was a Ph.D. student in the Department of Electrical and Computer Engineering, Michigan State University. Shantanu Chakrabartty is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824. All correspondence regarding this paper should be directed to shantanu@egr.msu.edu

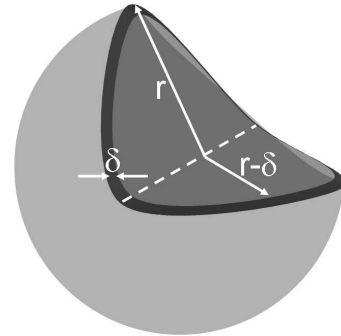


Fig. 1. Non-overlapping volume of two concentric spheres with radii  $r$  and  $(r - \delta)$

concentric hyper-spheres in  $D$  dimensions with radii  $r > 0$  and  $r - \delta > 0$  respectively. An equivalent arrangement is shown in Fig. 1 for  $D = 3$  dimensions. According to [6], the ratio of volumes of the hyper-spheres in  $D$  dimensions is given by

$$V(D) = \frac{(r - \delta)^D}{r^D} \quad (1)$$

$$= \left(1 - \frac{\delta}{r}\right)^D. \quad (2)$$

From the equation (2), it can be seen that for  $0 < \delta \leq r$ ,  $V(D) \rightarrow 0$  as the dimensionality of the space satisfies  $D \rightarrow \infty$ . Thus, this simple example illustrates that: *In higher dimensions, the volume and hence the probability of the data is typically concentrated on the surface of a hyper-sphere.* For the example shown in Fig. 1 with parameters  $\frac{\delta}{r} = 0.1$  and  $D = 32$ , the volume (or probability of occurrence) of data on the outer shell is 97%. Other geometric structures (hyper-cubes, hyper-ellipsoids, hyper-toroids etc.) also exhibit interesting data concentration properties in higher dimensions. For instance, the volume of a high-dimensional hyper-cube can be shown to be concentrated at the corners. These examples show that information (data distribution) in high-dimensional space is typically concentrated on low-dimensional manifolds, for example, hyper-surfaces or hyper-planes. The distribution becomes even more concentrated when the input signals exhibit significant degree of correlation (redundancy) which is true for analog signals recorded by high-density sensors. Therefore, an efficient high-dimensional ADC should identify the salient low-dimensional manifolds during the process of data conversion, which is contrary to many existing multi-channel data acquisition algorithms which operate independently on each of the input dimensions. In machine learning literature, an equivalent problem is known as manifold learning [7] where the objective is to determine parameters of a low-

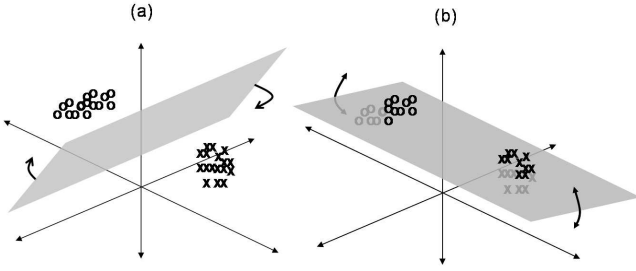


Fig. 2. Illustration of  $\Sigma\Delta$  learning using two-dimensional data: (a) initial hyperplane; (b) limit-cycle about the optimal hyperplane

dimensional manifold that can faithfully capture the geometric and statistical property of the high-dimensional input data. Some of the examples of manifold learning include principal component analysis (PCA), Kohonen maps [8], locally linear embedding (LLE) [9] and eigen-maps [10].

In this paper, we unify manifold learning and analog-to-digital conversion such that the proposed algorithm will process the input analog signals to produce not only a digitized representation of the transformed signal, but also a digitized representation of a transformation manifold. At the core of the proposed architecture and the focus of this paper is a stochastic min-max optimization procedure that yields a  $\Sigma\Delta$  modulation [11] integrated with a manifold learning step. Conceptually, the mechanism of the proposed modulation is illustrated in Fig. 2 for a simple manifold learning task. The example consists of input vectors (represented as circles and squares) which are embedded in a three dimensional space. An optimal manifold that captures the distribution of the input data is a two-dimensional hyper-plane as shown by the shaded area in Fig. 2(b). Starting from an initial estimate of the hyper-plane (shown in Fig. 2(a)) the  $\Sigma\Delta$  learning algorithm proposed in this paper will generate a sequence of approximate (quantized) hyper-planes that exhibit limit-cycle behavior about the optimal hyper-plane (see Fig. 2(b)). The statistics of the limit-cycles will then encode the parameters of the manifold at a desired resolution. The limit-cycle based learning endows the  $\Sigma\Delta$  learners with additional properties:

- The stability of learning is only dependent on the magnitude of the input signals and does not depend on the choice of hyper-parameters, for instance learning rate factors used in neural network architectures [7], [12].
- The algorithm naturally inherits the robustness properties of  $\Sigma\Delta$  modulation [11]. In addition, learning endows the proposed algorithm with a self-calibrating property where the mismatch between the sensor channels and the inherent non-linearity of analog computation are compensated. This is similar in spirit of many learning-on-silicon architectures reported in literature that can automatically compensate for hardware artifacts [13].
- When the objective of the algorithm is to determine only the parameters of slowly varying low-dimensional manifold and not the transformed data, then the  $\Sigma\Delta$  learner can operate at sampling rates lower than the Nyquist rate of the input signals. This setting is similar to *analog-to-information converters* [14], [15] except that

the basis functions are determined adaptively instead of being fixed.

The functional architecture of a  $\Sigma\Delta$  learner is shown in Fig. 3(a) where the input is a time varying analog signal  $\mathbf{x}[n] \in \mathcal{R}^D$  with  $n = 1, 2, \dots$  representing the discrete time index. The  $\Sigma\Delta$  learner also consists of a matrix-vector-multiplier (MVM) which transforms the input signal  $\mathbf{x}[n]$  according to  $\mathbf{A}[n]\mathbf{x}[n]$  where  $\mathbf{A}[n] \in \mathcal{R}^D \times \mathcal{R}^D$  denotes a linear transformation matrix. This transformed signal is then processed by an array of  $\Sigma\Delta$  modulator to produce a binary data stream  $\mathbf{d}[n] \in \{-1, +1\}^D$ . An adaptation unit uses the binary output  $\mathbf{d}[n]$  to update the matrix  $\mathbf{A}[n]$  and in the process learn the parameters of the target manifold  $\mathbf{A}_\infty$ . To illustrate the benefits of the proposed  $\Sigma\Delta$  learner over a conventional manifold learning architecture, we will compare their equivalent energy efficiency for a signal compression application. The functional architecture of a conventional manifold learning approach is shown in Fig. 3(b), where the transformation is performed after the analog-to-digital conversion step (using  $\Sigma\Delta$  modulators). For the comparison we will also make the following assumptions which is reasonable for a high-density sensor array:

- The rank of the target compression manifold  $\mathbf{A}_\infty$ , denoted by  $M$ , satisfies  $M \ll D$ . This is a valid assumption since signals acquired by high-density sensors (for example microelectrode array or microphone array) exhibit a high degree of correlation. This implies that once the parameters of  $\mathbf{A}_\infty$  has been learned,  $D - M$  redundant  $\Sigma\Delta$  modulators in Fig. 3(a) can be selectively shutdown to conserve energy.
- The desired manifold  $\mathbf{A}_\infty$  is quasi-stationary and its parameters vary much slowly compared to the input signal  $\mathbf{x}$ . This is a reasonable assumption since  $\mathbf{A}_\infty$  depends only on the physical properties of the sensor, sources and the channel. For instance, the distance between the sensor array and the sources or the variation in channel properties (dispersion or attenuation) directly affect  $\mathbf{A}_\infty$ . Thus, identifying and tracking of  $\mathbf{A}_\infty$  consumes only a fraction  $\alpha \ll 1$  of the total operational time of the  $\Sigma\Delta$  learner.

The operation of a  $\Sigma\Delta$  learner would consist of two sequential phases (shown in Fig. 3(c)) which repeat for the duration for which the  $\Sigma\Delta$  learner is active. In the first phase, the  $\Sigma\Delta$  learner learns/tracks the compression manifold  $\mathbf{A}_\infty$  within a certain number of adaptation cycles (consuming a fraction  $\alpha$  of the total operational cycles). In the second phase, the adaptation unit selectively shuts down the  $D - M$  redundant modulators for the rest of the operational cycles (shown in Fig. 3(a)). If  $P_{\Sigma\Delta}$  denotes the power dissipation of a single  $\Sigma\Delta$  modulator and if  $P_{adt}$  denotes the power dissipation of a single mixed-signal multiply-addition operation, under the following condition

$$\frac{P_{adt}}{P_{\Sigma\Delta}} < \frac{1}{D(\alpha + M/D)}. \quad (3)$$

the energy efficiency of the  $\Sigma\Delta$  learner can be shown to be superior to that of the conventional approach (shown in Fig. 3(b)). The proof of the inequality (3) is presented in

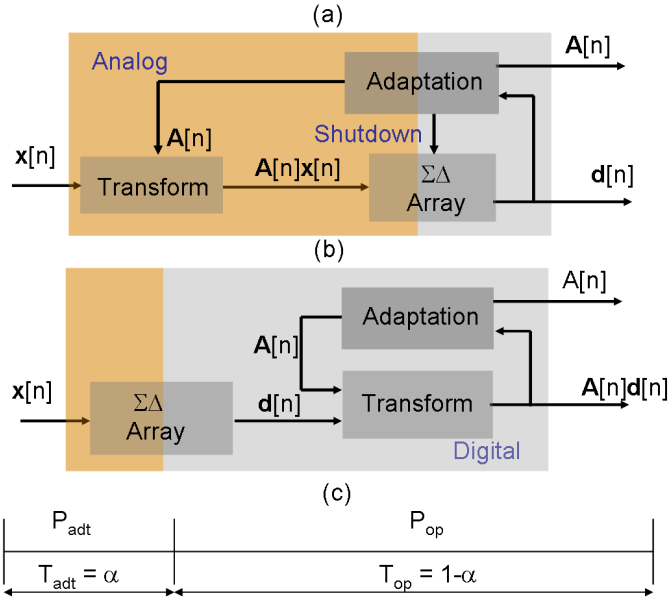


Fig. 3. Functional architectures corresponding to: (a) the proposed  $\Sigma\Delta$  learner and (b) conventional manifold learning approach. (c) Figure illustrating that when  $\Sigma\Delta$  learner is applied for a signal compression task, learning can be performed for only a fraction  $\alpha \ll 1$  of the total operational time.

Appendix I and is based on the worst case analysis where the power dissipation due to digital signal processing in conventional architecture (see Fig. 3(b)) has been ignored. The inequality (3) shows that the power savings could be significant if the fraction  $\alpha$  and the ratio  $M/D$  are simultaneously small. In the later sections will verify the inequality by presenting measured results obtained from a  $\Sigma\Delta$  learner prototyped in a  $0.5\mu$  CMOS process. Also, using Monte-Carlo analysis we will show that the parameter  $\alpha$  is inversely proportional to the input dimension  $D$  which will show that the efficiency improvement of the  $\Sigma\Delta$  learner remains invariant with the increase in the input dimensionality  $D$ .

The paper is organized as follows: section II will briefly present a summary of mathematical notations and definitions used in this paper; section III introduces the min-max optimization framework for constructing  $\Sigma\Delta$  learner; section IV presents some performance results obtained using Monte-Carlo simulations that quantifies the stability and variance of the  $\Sigma\Delta$  learner; section V describes a CMOS implementation of a four dimensional  $\Sigma\Delta$  learner; section VI presents measured results obtained using the fabricated prototype; section VII concludes the paper with some final remarks.

## II. MATHEMATICAL NOTATIONS

We will use the following notations and definitions throughout the paper:

- 1) A scalar variable will be denoted by a lower case symbol, for example  $x$ .
- 2) A column vector will be denoted by bold symbol as  $\mathbf{x}$  and its elements will be denoted by  $x_i, i = 1, 2, \dots$
- 3) The  $L_1$  norm of a vector will be denoted by  $\|\mathbf{x}\|_1$  and is given by  $\|\mathbf{x}\|_1 = \sum_i |x_i|$ . The  $L_2$  norm of a vector will be denoted by  $\|\mathbf{x}\|_2$  and is given by  $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$ . An

$L_\infty$  norm of a vector will be denoted by  $\|\mathbf{x}\|_\infty$  and is given by  $\|\mathbf{x}\|_\infty = \max_i |x_i|$ . A vector constant will be represented by a bold numeral. For example  $\mathbf{0}$  denotes a vector with all elements equal to zero.

- 4) A matrix will be denoted by an upper-case bold symbol as  $\mathbf{A}$  and its elements will be denoted by  $a_{ij}, i = 1, 2, \dots; j = 1, 2, \dots$
- 5) The  $L_\infty$  norm of a matrix will be denoted by  $\|\mathbf{A}\|_\infty$  and is given by  $\|\mathbf{A}\|_\infty = \max_{\mathbf{x}: \|\mathbf{x}\|_\infty \leq 1} \|\mathbf{A}\mathbf{x}\|_\infty$ .
- 6)  $\mathbf{w}^T$  and  $\mathbf{A}^T$  denote a transpose operation for a vector and matrix.
- 7) Discrete time sequences will be denoted by indices  $n$  as  $\mathbf{x}[n], n = 1, 2, \dots$
- 8)  $\mathcal{E}_{\mathbf{x}}\{\cdot\}$  will denote an expectation operation given by  $\mathcal{E}_{\mathbf{x}}\{\cdot\} = \int_{\mathbf{x}} (\cdot) p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}$ , where  $p_{\mathbf{x}}(\mathbf{x})$  denotes the probability density function of  $\mathbf{x}$ .
- 9) Empirical expectation will be denoted by  $\mathcal{E}_n\{\cdot\}$  and is defined as a temporal average of a discrete time sequence. For example,  $\mathcal{E}_n\{\mathbf{d}_n\} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \mathbf{d}[n]$ . A double expectation operator will then be represented by  $\mathcal{E}_n\{\mathcal{E}_n\{\mathbf{d}[n]\}\} = \lim_{N \rightarrow \infty} \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \mathbf{d}[j]$ .
- 10) Differentiation of a function  $f$  with respect to a vector or a matrix represents an element-wise operation such that  $\frac{\partial f}{\partial \mathbf{v}} = [\frac{\partial f}{\partial v_1}, \frac{\partial f}{\partial v_2}, \dots]^T$ .

## III. OPTIMIZATION FRAMEWORK FOR $\Sigma\Delta$ LEARNING

In this section, we describe a generalized form of the min-max optimization framework that will integrate learning with analog-to-digital conversion. A special case of the proposed framework was introduced in [16] which was used for neural signal compression. Given a random input vector  $\mathbf{x} \in \mathcal{R}^D$  and an internal state vector  $\mathbf{w} \in \mathcal{R}^D$ , a  $\Sigma\Delta$  learner estimates the parameters of a linear transformation matrix  $\mathbf{A} \in \mathcal{R}^D \times \mathcal{R}^D$  according to the following optimization criterion:

$$\max_{\mathbf{A} \in \mathcal{C}} (\min_{\mathbf{w}} f(\mathbf{w}, \mathbf{A})) \quad (4)$$

where

$$f(\mathbf{w}, \mathbf{A}) = \lambda \|\mathbf{w}\|_1 - \mathbf{w}^T \mathcal{E}_{\mathbf{x}}\{\mathbf{A}\mathbf{x}\}. \quad (5)$$

$\lambda > 0$  denotes a hyper-parameter and  $\mathcal{C}$  denotes a constraint space of the transformation matrix  $\mathbf{A}$ . The term  $\|\mathbf{w}\|_1$  bears similarity to the regularization which is extensively used in machine learning algorithms [17], [18]. However, the  $L_1$  norm in Eqn. (5) is an important link connecting the cost function with a single bit quantization. This is illustrated in Fig. 4 which shows an example of a one-dimensional regularization function  $\|\mathbf{w}\|_1$ . The piece-wise behavior of  $\|\mathbf{w}\|_1$  leads to discontinuous gradient  $\text{sgn}(\mathbf{w})$  (shown in Fig. 4(b)) where  $\text{sgn}(\cdot)$  denotes a signum operation equivalent to a single bit quantization. Even though the regularization framework can be extended to multi-bit quantization, in this paper we will only discuss  $\Sigma\Delta$  learners with single quantizers. The minimization step in Eqn. (4) ensures that the state vector  $\mathbf{w}$  is correlated with the transformed input signal  $\mathbf{A}\mathbf{x}$  (signal tracking step) and the maximization step in Eqn. (4) adjusts the parameters of  $\mathbf{A}$  such that it minimizes the correlation (de-correlation step)

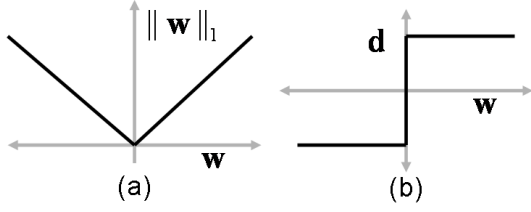


Fig. 4. Plot of (a) one dimensional  $L_1$  norm  $\|\mathbf{w}\|_1$  and (b) its derivative

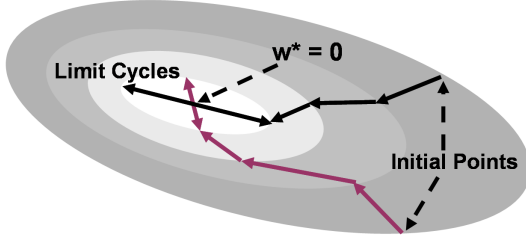


Fig. 5. Optimization contour explaining the limit cycle behavior about the minima.

between  $\mathbf{w}$  and  $\mathbf{Ax}$ . The formulation bears similarities with game-theoretic approaches [19], [20] where signal tracking and de-correlation have been formulated as conflicting objectives. However, the uniqueness of the proposed approach compared to other optimization techniques to solve Eqn. (4) is the use of bounded gradients to generate  $\Sigma\Delta$  limit-cycles. To show this we will first prove a key result:

*Lemma 1:* For the constraint set  $\{\mathcal{C} : \|\mathbf{A}\|_\infty \leq \lambda\}$ ,  $\|\mathbf{x}\|_\infty \leq 1$  and for  $f(\cdot, \cdot)$  as defined in Eqn. (5),  $\min_{\mathbf{w}} f(\mathbf{w}, \mathbf{A}) = 0$ .

*Proof:* First, we will show that the cost function  $f(\mathbf{x}, \mathbf{A})$  is bounded below by 0 and then show that this lower-bound lies within the feasible set defined by the constraints.

We will use topological property of norms [21] which states that for two integers  $p, q$  satisfying  $\frac{1}{p} + \frac{1}{q} = 1$ , the following relationship is valid for vectors  $\mathbf{w}$  and  $\mathbf{y}$

$$|\mathbf{w}^T \mathbf{y}| \leq \|\mathbf{w}\|_p \|\mathbf{y}\|_q \quad (6)$$

Setting  $\mathbf{y} = \mathcal{E}_x\{\mathbf{Ax}\}$  and applying Eqn. (6) the following inequality is obtained:

$$\|\mathbf{w}\|_1 \|\mathbf{y}\|_\infty \geq |\mathbf{w}^T \mathbf{y}| \geq \mathbf{w}^T \mathbf{y}. \quad (7)$$

Using the definition of the matrix norm and the given constraints, it can be easily seen  $\|\mathbf{A}\|_\infty \geq \|\mathbf{Ax}\|_\infty \geq \|\mathbf{y}\|_\infty$ . Thus,  $\|\mathbf{y}\|_\infty \leq \lambda$ . Therefore, the inequality (7) leads to

$$\lambda \|\mathbf{w}\|_1 - \mathbf{w}^T \mathcal{E}_x\{\mathbf{Ax}\} \geq 0 \quad (8)$$

which proves that the cost function  $f(\mathbf{x}, \mathbf{A})$  is bounded from below by 0. It can be seen that  $f(\mathbf{0}, \mathbf{A}) = 0$ , therefore, the lower-bound 0 lies within the constraint set and therefore is the minima of  $f(\mathbf{w}, \mathbf{A})$ .  $\square$

Lemma 1 shows that for the constraint set  $\mathcal{C}$ , the minima of  $f(\mathbf{w}, \mathbf{A})$  is already known and therefore the result of the minimization does not convey any additional information. However, in the proposed approach, the path to the final solution  $\mathbf{w}^* = \mathbf{0}$  and the limit-cycles about the solution

will be of importance. This is illustrated in Fig. 5 using a two-dimensional contour plot where starting from an initial condition, the minimization produces a trajectory towards the minima and ultimately produces a limit-cycle behavior about the minima. The path and the limit-cycles will encode the topology of the optimization manifold defined by  $f$  and hence will encode the transformation  $\mathbf{A}$ .

#### A. First-order $\Sigma\Delta$ Modulation

The link between optimization (4) and  $\Sigma\Delta$  modulation is through a stochastic gradient minimization [22] of the cost function (5). Under the condition of stationarity on the random vector  $\mathbf{x}$  and under the assumption that its probability density function of  $\mathbf{x}$  is well behaved (gradient of expectation operator is equal to expectation of the gradient operator), the stochastic gradient step with respect to  $\mathbf{w}$  yields

$$\mathbf{w}[n] = \mathbf{w}[n-1] - \left. \frac{\partial f(\mathbf{w}, \mathbf{A})}{\partial \mathbf{w}} \right|_{(n-1)} \quad (9)$$

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \mathbf{A}[n-1]\mathbf{x}[n-1] - \lambda \mathbf{d}[n] \quad (10)$$

where  $n$  signifies the discrete time index and  $\mathbf{d}[n] = \text{sgn}(\mathbf{w}[n-1])$  denotes the quantized representation according to the step function shown in Fig. 4(b). Note that the formulation (10) does not include any learning rate parameters typically used in other neural network approaches. As the recursion (10) progresses, bounded limit cycles are produced about the solution  $\mathbf{w}^*$  (see Fig. 5) whose property is characterized by the following lemma.

*Lemma 2:* For  $\mathbf{A}[n] \in \mathcal{C}$ ,  $\|\mathbf{x}\|_\infty \leq 1$  and if  $\|\mathbf{w}_0\|_\infty \leq 2\lambda$ , then  $\|\mathbf{w}_n\|_\infty \leq 2\lambda$  for  $n = 1, 2, \dots$

*Proof:* Similar to the proof for the stability of a first-order  $\Sigma\Delta$  modulation [11], we will use mathematical induction to prove the lemma. Let  $\|\mathbf{w}[n-1]\|_\infty \leq 2\lambda$ . Since  $\mathbf{d}[n] = \text{sgn}(\mathbf{w}[n-1])$ , therefore  $\|\mathbf{w}[n-1] - \lambda \mathbf{d}[n]\|_\infty \leq \lambda$  (proof for this claim is given in Appendix II). Using Eqn. (10), the following relationship holds

$$\begin{aligned} \|\mathbf{w}[n]\|_\infty &= \|\mathbf{w}[n-1] - \lambda \mathbf{d}[n] + \mathbf{A}[n-1]\mathbf{x}[n-1]\|_\infty \\ &\leq \|\mathbf{w}[n-1] - \lambda \mathbf{d}[n]\|_\infty + \|\mathbf{A}[n-1]\|_\infty \\ &\leq 2\lambda \end{aligned} \quad (11)$$

which proves the lemma using a mathematical induction.  $\square$ .

Assuming the initial condition of  $\mathbf{w}[0] = \mathbf{0}$ , the discrete time recursion (10) leads to

$$\frac{\lambda}{N} \sum_{n=1}^N \mathbf{d}[n] = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{A}[n]\mathbf{x}[n] + \frac{1}{N} \mathbf{w}[n] \quad (12)$$

Due to the bounded property of  $\mathbf{w}[n]$  (according to lemma 2), equation (12) leads to the following asymptotic property:

$$\mathcal{E}_n\{\mathbf{d}[n]\} = \frac{1}{\lambda} \mathcal{E}_n\{\mathbf{A}[n]\mathbf{x}[n]\} \quad (13)$$

where  $\mathcal{E}_n\{\cdot\}$  denotes an empirical expectation with respect to time index  $n$ . Thus, the recursion (10) produces a quantized sequence whose mean asymptotically encodes the transformed

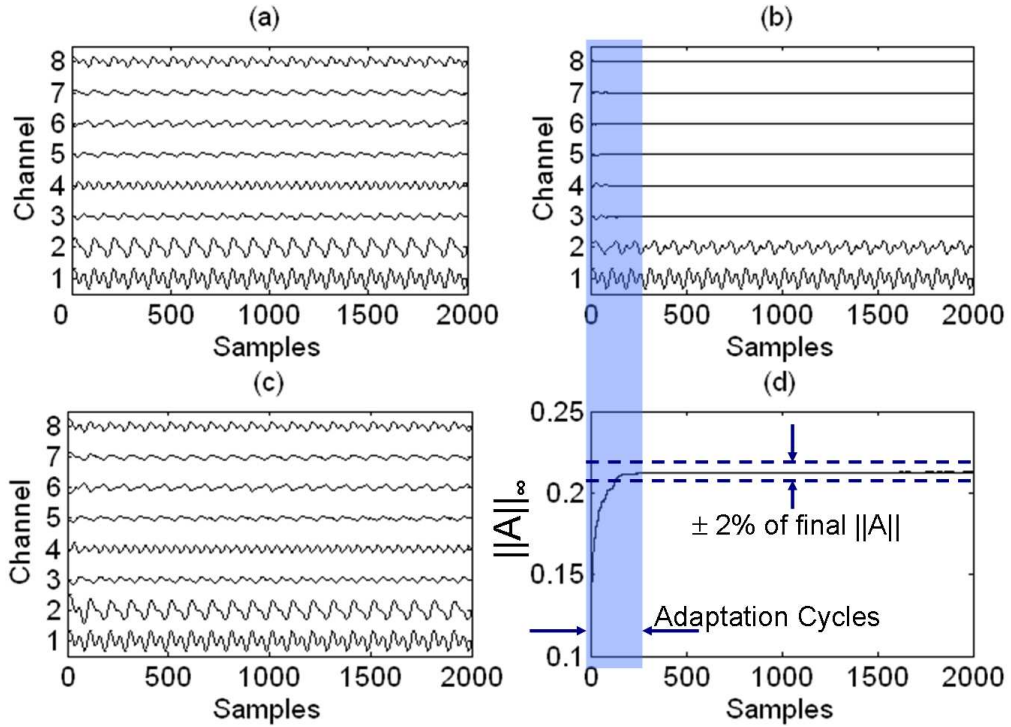


Fig. 6. Functional verification of a  $\Sigma\Delta$  learner using synthetic multi-channel data: (a) analog signals presented as input to the learner; (b) output signals produced by the learner; (c) reconstructed signals using the output of learner and the learned transformation  $\mathbf{A}$ ; (d) convergence of the  $\|\mathbf{A}\|_\infty$

input at infinite resolution. For a stationary input sources (sources whose statistical properties are fixed),  $\mathbf{A}[n]$  converges to an asymptotic value  $\mathbf{A}_\infty$ , which then can be used for reconstruction according to

$$\mathcal{E}_n\{\hat{\mathbf{x}}[n]\} \approx \lambda \mathbf{A}_\infty^{-1} \mathcal{E}_n\{\mathbf{d}[n]\} \quad (14)$$

where  $\mathbf{A}_\infty^{-1}$  is the inverse transform of  $\mathbf{A}_\infty$ . Since for high-density sensing, the rank of the input space is less than its dimension, it is important to choose a constraint space  $\mathcal{C}$  where the non-redundant  $M$  dimensional space can be easily identified. One such constraint space is represented by upper and lower triangular matrices with their diagonal elements fixed. Also for such a class of transforms, the inverse  $\mathbf{A}_\infty^{-1}$  can be easily computed using back-substitution techniques.

### B. $\Sigma\Delta$ learning

The maximization step (de-correlation) in Eqn. (4) yields updates for matrix  $\mathbf{A}$  according to a gradient ascent procedure as

$$\mathbf{A}[n] = \mathbf{A}[n-1] + \xi \left. \frac{\partial f(\mathbf{w}, \mathbf{A})}{\partial \mathbf{A}} \right|_{n-1} \quad (15)$$

which can be written as

$$\mathbf{A}[n] = \mathbf{A}[n-1] - \xi \mathbf{w}[n-1] \mathbf{x}[n-1]^T; \mathbf{A}[n] \in \mathcal{C}. \quad (16)$$

The parameter  $\xi$  controls the learning rate of the update (16). Since we are interested in obtaining a digitized representation of the matrix  $\mathbf{A}$  the parameter  $\xi$  in update (16) can be replaced by its binary form as  $\xi = 2^{-P}$  and the variables  $\mathbf{w}[n-1]$ ,  $\mathbf{x}[n]$  can be replaced by its signed forms as:

$$\mathbf{A}[n] = \mathbf{A}[n-1] - 2^{-P} \mathbf{d}[n] \text{sgn}(\mathbf{x}[n-1])^T; \mathbf{A}[n] \in \mathcal{C} \quad (17)$$

where we have used the relationship that  $\mathbf{d}[n] = \text{sgn}(\mathbf{w}[n-1])$ . The updates in Eqn. (17) can be implemented using an up-down counter which also acts as a storage for the binary representation of the matrix  $\mathbf{A}$ . The parameter  $P$  will be referred to as the transform resolution as it determines the precision by which the target manifold  $\mathbf{A}_\infty$  can be determined. It is important to point out that unlike conventional neural network algorithms where the choice of the learning rate parameter is important to guarantee stability of the algorithm, the parameter  $P$  only affects the performance and not the stability of the  $\Sigma\Delta$  learner. This has been verified using Monte-Carlo simulations presented in section IV.

To ensure that the matrix  $\mathbf{A}$  always lies within the constraint space  $\mathcal{C}$  (space of lower triangular matrices), the updates in equation (17) are only applied to the off-diagonal lower triangular matrix elements. The values of other elements are fixed according to  $a_{ij} = 0, \forall i < j$ , and  $a_{ii} = 1$ . Thus, if  $\|\mathbf{A}\|_\infty \leq \lambda$  is satisfied, then the recursion (17) will asymptotically lead to

$$\mathcal{E}_n\{\mathbf{d}[n] \text{sgn}(\mathbf{x}[n])^T\} \rightarrow 0 \quad (18)$$

for  $\mathbf{A}_\infty \in \mathcal{C}$ . Thus, the proposed  $\Sigma\Delta$  learning algorithm produces quantized sequences  $\mathbf{d}[n]$  that are uncorrelated to the signum function of the input signal. We will first illustrate using a synthetic example that the condition (18) is sufficient to identify redundant  $\Sigma\Delta$  modulation paths. For this example two sinusoidal signals with different frequencies were first chosen. These signals were then mixed together in linear proportions to generate eight synthetic signals as shown in Fig. 6(a). These signals were then presented as a input to the  $\Sigma\Delta$  learner. Thus,

even though the dimensionality of the input signal space for this example is eight, its rank is two. Fig. 6(b) shows the output produced by the  $\Sigma\Delta$  learning algorithm where the binary stream  $\mathbf{d}[n]$  has been low-pass filtered and decimated [11]. The results demonstrate that the  $\Sigma\Delta$  learner correctly reduces the dimensionality of the output signals, where only two of the eight channels contain significant energy, where as the energy in the rest of the channels (called *residual energy*) diminishes to zero. The convergence of the learning algorithm can be visualized for this example using Fig. 6(d), which shows that the  $\|\mathbf{A}\|_\infty$  stabilizes after a certain number of iterations. We will quantify the convergence behavior of the  $\Sigma\Delta$  learner using *adaptation cycles*, which is defined as the number of learning iterations required before  $\|\mathbf{A}\|_\infty$  converges to  $\pm 2\%$  of its stabilized value. The adaptation cycles also determines the parameter  $\alpha$  which is the fraction of the total operation period when learning is enabled. The converged value of  $\mathbf{A}_\infty$  is used to reconstruct the input signals using the output according to Eqn. (14) which is shown in Fig. 6(c). In the numerical experiments presented later in Section IV, the quality of reconstruction will be quantified using the *reconstruction error* which is a mean-square error between the reconstructed output and the input signal. This simple experiment demonstrates the functionality of  $\Sigma\Delta$  learning in identifying redundant (null-space) signal processing paths. This response can also be seen in the frequency domain as shown in Fig. 7 where the FFT of one of the redundant channels (channel 8) is illustrated for a first-order  $\Sigma\Delta$  modulator (a) with and (b) without learning. Since, for this example the input signal consisted of two fundamental sinusoids the frequency domain response consists of two distinct impulses. Also, it can be seen from Fig. 7, that in addition to the familiar noise-shaping characteristics of a first-order modulator [11], the  $\Sigma\Delta$  learner suppresses the sinusoidal signals (redundant signals) after learning. Thus, the learner acts as a band reject filter with respect to signal present in channel 1 and 2. Moreover, any common mode signals, for instance dc offsets, which are associated with all channels are eliminated.

### C. Higher-order $\Sigma\Delta$ Modulation

The formulation of  $\Sigma\Delta$  learning could also be extended to higher-order modulation. This can be achieved by incorporating momentum terms in Eqn. (10) to obtain

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \left. \frac{\partial f(\mathbf{w}, \mathbf{A})}{\partial \mathbf{w}} \right|_{n-1} + \mathbf{w}[n-1] - \mathbf{w}[n-2] \quad (19)$$

$$\begin{aligned} \mathbf{w}[n] = & \mathbf{w}[n-1] \\ & + (\mathbf{A}[n-1]\mathbf{x}[n-1] - \lambda\mathbf{d}[n]) \\ & + \mathbf{w}[n-1] - \mathbf{w}[n-2] \end{aligned} \quad (20)$$

Momentum terms have extensively been used in optimization theory and neural networks for improving the performance of learning algorithms [22], [23] and has been used in  $\Sigma\Delta$  learning to improve its convergence speed. Even though Hessian-based formulations have also been proposed

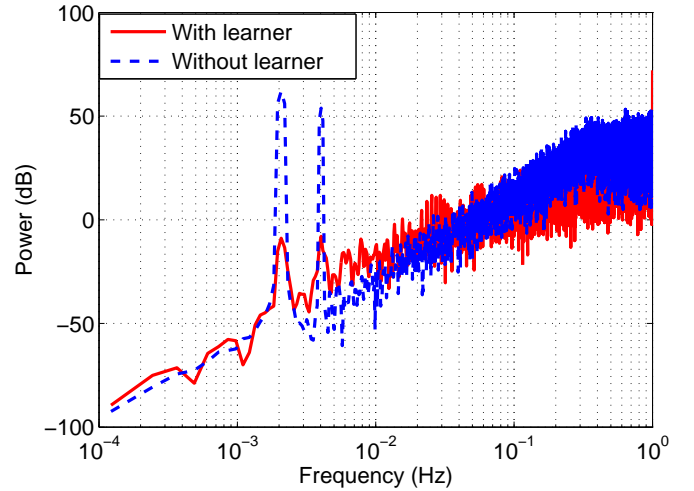


Fig. 7. FFT response of the first-order  $\Sigma\Delta$  learner for the synthetic example in Fig. 6, showing that the 8<sup>th</sup> channel rejects frequencies present in the lower dimensional channels

for improving the convergence speed of neural network algorithms, they are not suitable for optimizing piece-wise cost function (4). Recursion (20) will generate quantized vector sequences  $\mathbf{d}[n]$  whose first order expectation as well as second order expectation converge asymptotically according to

$$\mathcal{E}_n\{\mathbf{d}[n]\} = \frac{1}{\lambda}\mathcal{E}_n\{\mathbf{A}[n]\mathbf{x}[n]\} \quad (21)$$

$$\mathcal{E}_n\{\mathcal{E}_n\{\mathbf{d}[n]\}\} = \frac{1}{\lambda}\mathcal{E}_n\{\mathbf{A}[n]\mathbf{x}[n]\}. \quad (22)$$

The proof of convergence for the expressions (21 - 22) is given in Appendix III. The update in Eqn. (20) is equivalent to a second-order  $\Sigma\Delta$  modulator [11], thus linking the momentum based gradient descent rule to a second-order  $\Sigma\Delta$  modulation. Similar to Eqn. (14), the reconstruction formula based on Eqn. (22) can be expressed in terms of asymptotic value of the linear transform  $\mathbf{A}_\infty$  as

$$\mathcal{E}_n\{\hat{\mathbf{x}}[n]\} \approx \lambda\mathbf{A}_\infty^{-1}\mathcal{E}_n\{\mathcal{E}_n\{\mathbf{d}[n]\}\} \quad (23)$$

Eqn. (19) can also be generalized to incorporate  $L^{\text{th}}$  order momentum terms according to

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \left. \frac{\partial f(\mathbf{w}, \mathbf{A})}{\partial \mathbf{w}} \right|_{n-1} + \Delta_n^L(\mathbf{w}[n-1]) \quad (24)$$

where  $\Delta_n^L(\cdot)$  denotes an  $L^{\text{th}}$  order difference. In case of  $L = 1$  Eqn. (24) is equivalent to a second order modulation according to Eqn. (19). Higher-order momentum terms have also been used in neural networks [24] to accelerate the dynamics of gradient descent iterations especially where optimization contours are flat. However, without any loss of generality, in this paper we will only investigate properties of first and second order modulators only. The optimization approach using momentum terms can also be used to visualize and understand the dynamics of  $\Sigma\Delta$  modulators. Fig. 8(a)-(d) illustrates this using a one-dimensional cost function  $f(w) = |w| - wx$  with  $|x| < 1$ . Fig. 8(a) corresponds to  $x = 0$ , therefore the stochastic gradient iterations corresponding to the first-order  $\Sigma\Delta$

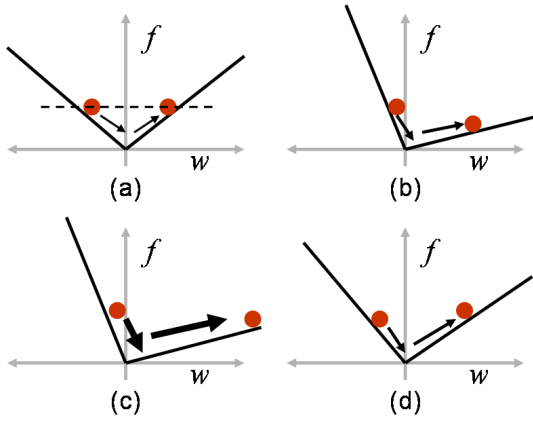


Fig. 8. Illustration of  $\Sigma\Delta$  dynamics for (a) first-order modulator with  $x = 0$ , (b) modulator with  $x > 0$ ; (c) higher-order modulators (bold lines indicate higher velocity); (d) effect of reducing the magnitude of  $x$

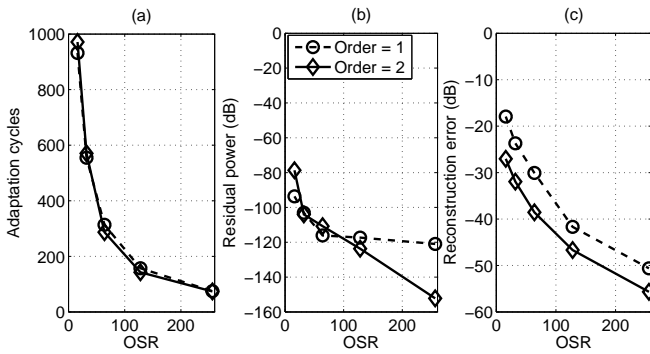


Fig. 9. The effect of OSR of the  $\Sigma\Delta$  converter array on the performance of the  $\Sigma\Delta$  learner.

modulation exhibits limit cycles symmetrical about the minima  $w = 0$ . Fig. 8(b) shows the equivalent dynamics corresponding to  $x > 0$ , therefore the resulting limit cycles spend more time in the region  $w > 0$ . Fig. 8(c) shows the dynamics of a higher-order modulator, where the momentum factor accelerates the marker towards the minima. The overshoot beyond the minima is proportional to the net velocity at the minima. For higher-order modulators, the acceleration and hence the velocity of approach could be large enough so that the magnitude of limit cycles can become unbounded. Therefore, the stability of  $\Sigma\Delta$  modulation can be improved either by reducing the velocity of approach towards the minima (by modifying the shape of the optimization manifold) or by constraining the magnitude of the input  $x$ . These approaches are similar to stabilization techniques already used for designing higher-order  $\Sigma\Delta$  modulators [11].

#### IV. MONTE-CARLO ANALYSIS OF $\Sigma\Delta$ LEARNERS

In this section we analyze the performance of the  $\Sigma\Delta$  learning algorithm using Monte-Carlo simulations. The system parameters that were included in this study are: (a) Oversampling ratio (OSR), which is defined as the ratio of frequency of  $\Sigma\Delta$  learning updates to the bandwidth of the input signal; (b) Sparsity, which is the ratio of the number of input channels  $D$  and the rank of the input signal space  $M$  and (c) Transform

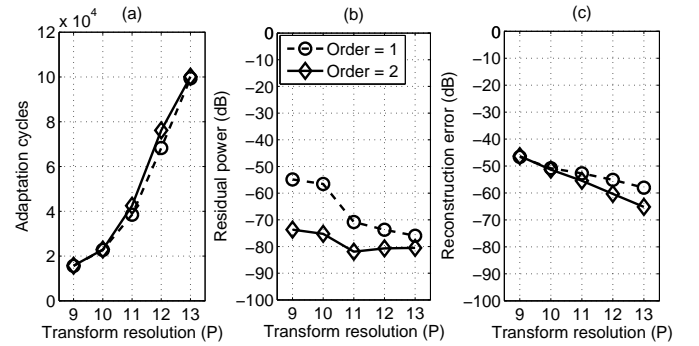


Fig. 10. The effect of resolution of a signal transformation matrix  $\mathbf{A}$  on the performance of the  $\Sigma\Delta$  learner.

resolution ( $P$ ), which quantifies the precision of the updates in the Eqn. (17). To understand the effect of these system parameters on the performance of the learning algorithm, a controlled simulation experiment was found to be better suited than using real-life data. For all experiments presented in this section, we used a similar setting as described in Fig. 6 which consists of two sinusoidal signals (with different normalized frequencies) mixed in random proportions. The rank of the eight dimensional input space was fixed to two. The performance of the first-order and the second-order  $\Sigma\Delta$  learning algorithm were then quantified using system parameters as described in section III-C. These parameters are:

- 1) *Adaptation cycles*: The minimum learning iterations before the  $\|\mathbf{A}\|_\infty$  converged to  $\pm 2\%$  of the stabilized value.
- 2) *Residual power*: Quantifies the performance of the algorithm to achieve signal compression. This can be calculated using the signal at the output of the  $i^{\text{th}}$  channel according to  $R_i = \mathcal{E}_n\{(\mathcal{E}_n\{d_i[n]\})^2\}$ .
- 3) *Reconstruction error*: Quantifies the accuracy of the learning algorithm in identifying the compression transform. For the numerical experiments presented in this section, the reconstruction error is calculated by computing the mean-square error between the input signal and signal reconstructed according to Eqn. (23) as  $R_e = \mathcal{E}_n\{\|\mathbf{x}[n] - \hat{\mathbf{x}}[n]\|_2^2\}$ .

All numerical results presented in this section were obtained after averaging the results over 100 runs, where for each run, two sinusoidal signals (see Fig. 6) were mixed in random proportions to produce eight input signals.

Fig. 9 shows that the adaptation cycles required for  $\Sigma\Delta$  learning decreased with increase in OSR (for both the first and the second order modulation). This can be attributed to the increased precision in tracking of the stationary manifold with an increase in the OSR. As expected, higher OSR also improved the reconstruction and the suppression of residual error which is shown in Fig. 9(b) and (c). Fig. 10 summarizes the performance of the  $\Sigma\Delta$  learner when the resolution parameter  $P$  was varied. It can be seen that the number of adaptation cycles increases with the increase in parameter  $P$ . This is because the learning algorithm has to span the signal space at finer increments. Also, from Fig. 10, it can be seen

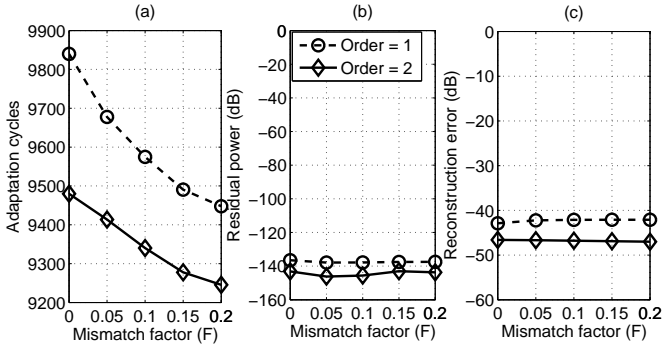


Fig. 11. The effect of mismatch on the performance of the  $\Sigma\Delta$  learner.

that the reconstruction error and the residual error decreased with increased resolution. This can be attributed to a more precise determination of the transformation parameters with an increase in  $P$ .

The next set of experiments were used to evaluate the robustness of the proposed  $\Sigma\Delta$  learner to different forms of computational artifacts. In literature, the effect of computational artifacts on the performance of  $\Sigma\Delta$  modulation has been extensively studied [25], [26] and several compensation methods have been proposed [27], [28], [29]. Three kinds of artifacts have been considered in this paper which are consistent with other reported numerical study in literature. These include: (a) mismatch and random offset errors introduced while computing the transform; (b) leakage in the  $\Sigma\Delta$  recursion and (c) the non-linearity of analog computation.

Mismatch and random offset errors were introduced in  $\Sigma\Delta$  learning by adding a random error matrix  $\epsilon$  with  $\|\epsilon\|_\infty \leq F$  to the transformation matrix  $\mathbf{A}$ . The parameter  $F$  denotes the mismatch factor and was used to quantify the system performance as shown in Fig. 11. It can be seen from Fig. 11(a) that the number of adaptation cycles decreases with increased in the mismatch factor. This was consistent with several results reported in machine learning literature where randomization aids the convergence of the learning algorithm [17], [34]. The residual power and the reconstruction error, however, remain unchanged demonstrating the robustness of  $\Sigma\Delta$  learning to mismatch artifacts. The residual power and the reconstruction error of the second order system showed variations up to 2% and 1% compared to 1.5% and 1.7% of the first order system.

The non-linearity of matrix-vector multiplier was modeled using a compressive response in the  $\Sigma\Delta$  update according to

$$\mathbf{w}[n] = \mathbf{w}[n-1] + \mathbf{g}(\mathbf{A}[n-1], \mathbf{x}[n-1]) - \mathbf{d}[n]. \quad (25)$$

where

$$\mathbf{g}(\mathbf{A}[n], \mathbf{x}[n]) = 1/(1 + \exp(-\beta \mathbf{A}[n] \mathbf{x}[n])) \quad (26)$$

with  $\beta$  being a hyper-parameter that controlled the shape of non-linearity (shown in Fig. 13(a)). Fig. 12(a) shows that the number of adaptation cycles decreased with increase in the parameter  $\beta$ . This can be attributed to higher gradient at the origin which led to faster convergence speed and hence smaller number of adaptation cycles. For this experiment, the

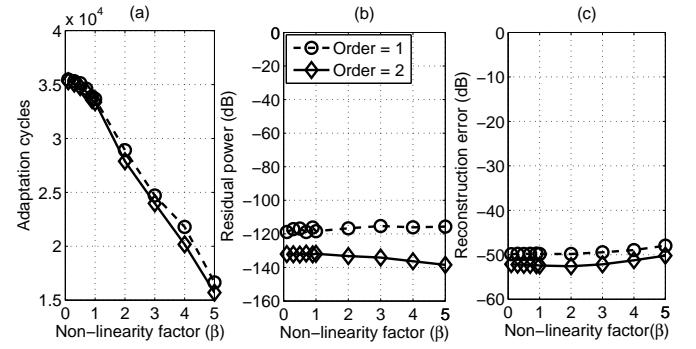


Fig. 12. The effect of computational non-linearity on the performance of the  $\Sigma\Delta$  learner.

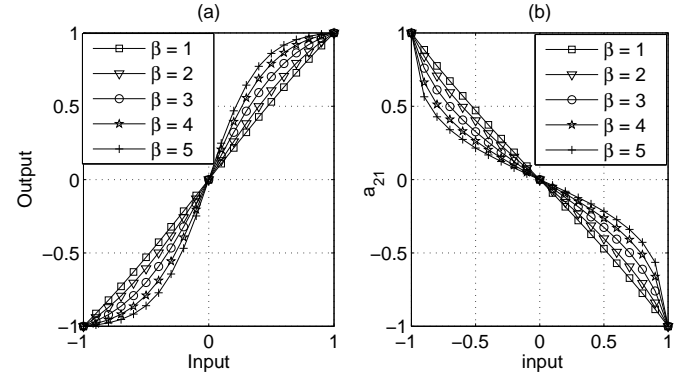


Fig. 13. Illustration of the self-calibration in a  $\Sigma\Delta$  learner: (a) The non-linearity of the transform as modeled by a sigmoidal function; (b) learned value of the transformation parameter  $a_{21}$  which tracks the inverse of the non-linear function.

maximum variation in the residual power and the reconstruction error was found to be 2.9 % and 3.2 % respectively, demonstrating the low sensitivity of the  $\Sigma\Delta$  learner to non-linear response. To understand the robustness property of the  $\Sigma\Delta$  learner to computational non-linearities consider the adapted value of the matrix element  $a_{21}$  as shown in Fig. 13(b). The value of  $a_{21}$  were obtained subsequent to convergence of the  $\Sigma\Delta$  learning for different values of inputs. It can be easily verified from Fig. 13(b) that  $a_{21} \approx g^{-1}(x_2)$ , thus compensating the non-linear effect of Eqn. (26).

The final experiment evaluated the effect of signal sparsity on the  $\Sigma\Delta$  learning performance. For this setup, the number of channels (dimension of the input vectors) was increased, while keeping the number of independent sinusoidal signals fixed. Thus, the rank of input signal space was always fixed to two, similar to setting described in Fig. 6. Fig. 14(a) shows that as the sparsity of the input space increases (while the rank is fixed), the number of adaptation cycles reduces. The numerical result illustrate that when the input channels show large degree of correlation (as in high-density sensing), increasing the number of input channels improves the convergence rate of learning. This shows that the learning algorithm can exploit more spatial information to successfully eliminate redundancy in the  $\Sigma\Delta$  modulator output with unchanged residual power. For this experiment, the worst case variation in the recon-

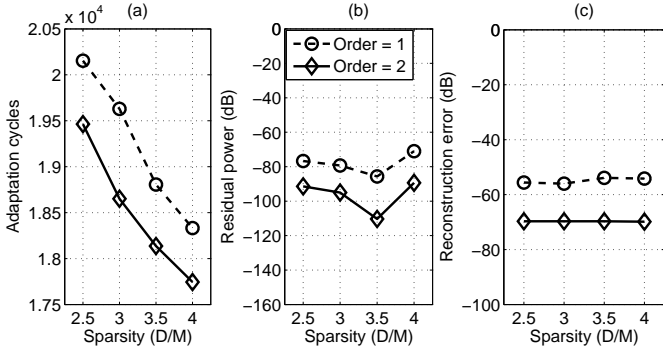


Fig. 14. The effect of input sparsity on the performance of the  $\Sigma\Delta$  learner.

struction error increased to 3.5%. Therefore, when the sparsity or the ratio  $D/M$  increased, the adaptation cycles and hence the fraction  $\alpha$  in Eqn. (3) decreased as  $O(1/D)$ . Thus the inequality 3 can be expressed as  $P_{adt}/P_{\Sigma\Delta} < O(1)$ , implying that the energy efficiency improvement of the  $\Sigma\Delta$  learner does not degrade with increase in signal dimensionality. In the next section we quantify the parameters  $P_{adt}$  and  $P_{\Sigma\Delta}$  through measurement results obtained from a prototype  $\Sigma\Delta$  learner fabricated in a  $0.5\mu\text{m}$  CMOS process.

## V. CMOS IMPLEMENTATION OF $\Sigma\Delta$ LEARNER

For the prototype implementation, the input dimension was chosen to be  $D = 4$  due to constraints on the silicon area. The system level architecture of the  $\Sigma\Delta$  learner is shown in Fig. 15. It consists of an array of analog processing units (APUs) which implements the following matrix-vector operation

$$\begin{pmatrix} y_1[n] \\ y_2[n] \\ y_3[n] \\ y_4[n] \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ a_{21}[n] & 1 & 0 & 0 \\ a_{31}[n] & a_{32}[n] & 1 & 0 \\ a_{41}[n] & a_{42}[n] & a_{43}[n] & 1 \end{pmatrix} \begin{pmatrix} x_1[n] \\ x_2[n] \\ x_3[n] \\ x_4[n] \end{pmatrix}$$

Each APU implements a single multiply-accumulate operation between the stored digitized representation of  $a_{ij}$  and the input signal  $x_j$ . The APUs also adapt the stored parameter  $a_{ij}$  according to the equation (17). Note that the APU array is organized in a lower-triangular form which ensures that the constraint  $C$  is satisfied. Also, the APUs on the diagonal of the array do not adapt, hence denoted by a different symbol. Each APU consists of a transconductor  $T_{ij}$  (see Fig. 15 inset) whose bias current is proportional to the matrix element  $a_{ij}$ . The bias current modulates the transconductance and hence the output current  $I_{OUT}$  is proportional to the product of  $a_{ij}$  and the input signal  $x_j$ . Additions in the matrix-vector-multiplication were implemented using Kirchoff's current summation principle at the common node  $y_i[n]$  where all outputs of the APUs are connected. The nodes  $y_i[n]$  are maintained at a virtual ground (by  $\Sigma\Delta$  modulators) which ensures that the current summation is accurate.

Updates of the parameter  $a_{ij}$  (according to Eqn. 16) are implemented using a digitally programmable current DAC (see Fig. 15 inset). An 11-bit up-down counter stores a digital representation of  $a_{ij}$  which is processed by the DAC to

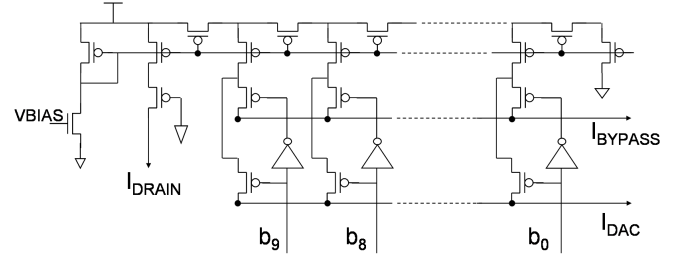


Fig. 16. Schematic of a binary 10-bit current mode DAC.

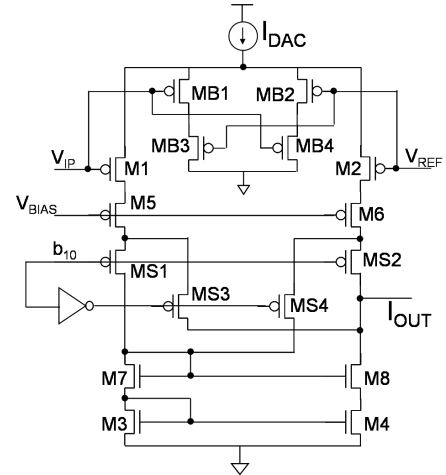


Fig. 17. Schematic of the source degenerated transconductor.

produce the bias current of the transconductor. Because the updates in Eqn. (17) is binary, the multiplication operator in (17) is implemented using an XNOR gate (see Fig. 15 inset). The output of the XNOR gate drives the up-down control signal of the counter. The design of the up-down counter is based on a network of D flip-flops and has been optimized in this work for area and power dissipation. The counter also incorporates a shift capability where the contents of the counter can be initialized and accessed using a serial-chain interface. Note that for the diagonal APUs the parameters  $a_{ii}$  are non-adaptive and hence does not contain any counter and the current-DAC.

The 10 least significant bits of the Up-Down counter  $b_0, \dots, b_9$  drive a 10-bit current DAC which is implemented using a standard MOS resistive network [30]. The output current  $I_{DAC}$  then modulates the bias current of a transconductor whose schematic is shown in Fig. 17. The most significant bit  $b_{10}$  of the Up-Down counter controls the sign (direction) of the output current, and thus is used for implementing a four-quadrant multiplier. The transconductor consists of a p-MOS input stage which drives a cascoded output stage. The transconductor uses a bump circuit [31] (transistors  $MB1 - MB4$ ) for source degeneration and hence for increasing the input voltage range (or reducing the transconductance). The bump circuit operates by steering the output current such that the transistor pair implements an equivalent resistor. The direction of the output current is controlled by the  $b_{10}$  which

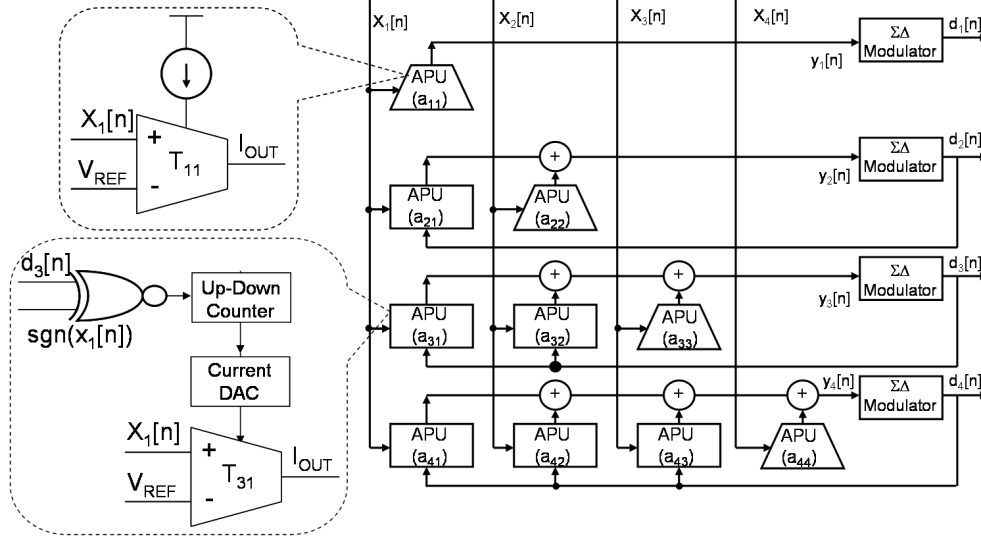


Fig. 15. Architecture of the  $\Sigma\Delta$  learner with  $D = 4$ .

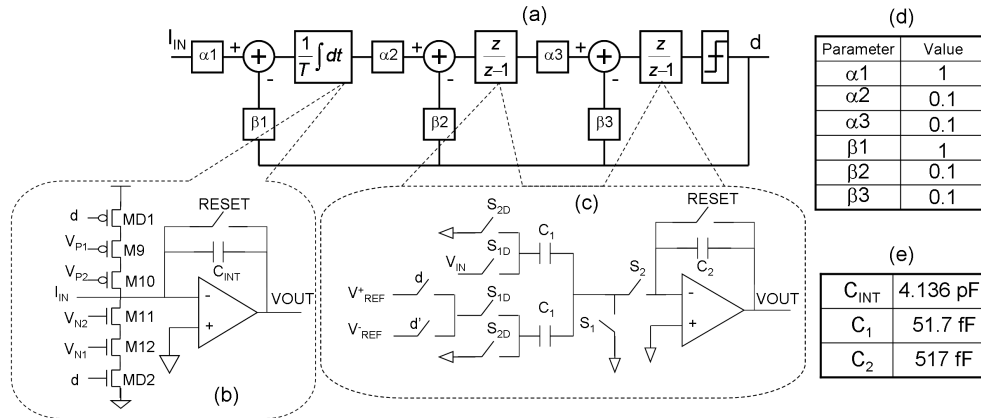


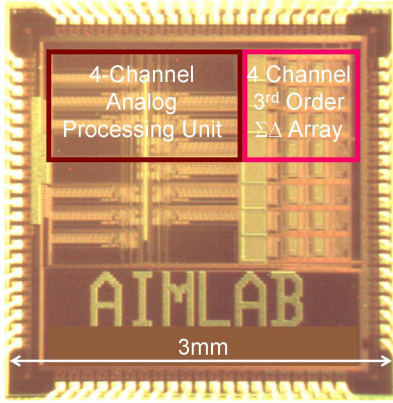
Fig. 18. Architecture of the third-order single loop single bit  $\Sigma\Delta$  modulator.

switches the currents using transistors  $MS1 - MS4$ . Compared to a voltage mode design for the APUs, the proposed current mode transconductor network significantly reduces the required silicon area. We estimate that the current mode transconductor reduces the area by a factor of 50 compared to its voltage mode counterparts designed with  $20fF$  capacitive network.

For the implementation of the  $\Sigma\Delta$  learner, a third-order modulator was chosen. Compared to a single-order modulator the third-order modulator can achieve a higher conversion rate and hence achieve higher energy efficiency (for a fixed OSR) [11]. Since the output of the APU array are currents, the first stage of the modulator uses a current-mode continuous time integrator. The subsequent modulator stages were implemented using voltage mode circuits (switched capacitor integrators) to avoid intermediate voltage to current conversion stages. Such hybrid  $\Sigma\Delta$  modulators [32] have been shown to relax design constraints on amplifier unity gain frequency, power budgets of continuous time modulators and the scal-

ability of discrete time modulators. Fig. 18(d) shows the gain parameters of the 3<sup>rd</sup> order hybrid (mixed mode)  $\Sigma\Delta$  modulator that were chosen to maintain the stability of the transconductor network.

The first stage of the modulator is a continuous time current mode integrator as shown in Fig. 18(b) where reference currents through transistor devices  $M9 - M12$  were biased to avoid  $\Sigma\Delta$  overload condition. As shown in Fig. 18(b), the multiplication between the digital bit  $d$  and the reference current is implemented by switching (on/off) the cascoded current source (sink). Switching at the source as opposed to switching at the drain has several advantages [33] as it reduces the channel charge injection [33] and clock feed-through at the integration node. The size of the integrating capacitor ( $C_{INT}$ ) in the first stage modulator was chosen to avoid integrator saturation as well as to limit the integrator swing within the accepted input range of the second stage modulator. If the reference current of the first stage of the modulator is denoted by  $I_{ref}$ , the clock period be denoted by  $T_{CLK}$  and the input


 Fig. 19. Microphotograph of the fabricated  $\Sigma\Delta$  learner.

voltage range of the second stage be denoted by  $V_{lim}$ , then the size of the integrating capacitor is chosen according to

$$C_{INT} = \frac{I_{ref} * T_{CLK}}{V_{lim}} \quad (27)$$

The second and the third stage modulators are switched capacitor modulators whose single ended version is shown in Fig. 18(c). The loop gain of the third order hybrid  $\Sigma\Delta$  modulator with capacitor sizes are shown in Fig. 18(e). For all integrators, a standard folded cascode op-amp is used which provides an open loop DC gain of 60dB. This is in conjunction with the minimum required gain greater than twice the oversampling ratio of the  $\Sigma\Delta$  converter [11] to reduce the effects of signal leakage due to a non-ideal integrator.

## VI. MEASUREMENT RESULTS

### A. Circuit characterization

Fig. 19 shows the micrograph of a prototype  $\Sigma\Delta$  learner which was fabricated in a  $0.5\mu\text{m}$  CMOS process. Table I summarizes the measured specifications of the  $\Sigma\Delta$  learner prototype. The  $SNR$ ,  $SNDR$  and the  $SFDR$  metrics are reported for a single  $\Sigma\Delta$  modulator whose input is driven by a single APU, while the other APUs are disabled. For all the experiments reported in this section, the input voltage swing was limited to 300mV which was determined by the linear operating range of the source degenerated transistor. The power dissipation of each of the components of the  $\Sigma\Delta$  learner are summarized in table II. The power dissipation of a single  $\Sigma\Delta$  modulator was measured to be  $P_{\Sigma\Delta} = 231\mu\text{W}$  where as the power dissipation of a single APU was measured to be  $P_{adt} = 15.68\mu\text{W}$ . Based on these measured quantities and the discussion presented in section I, it can be seen that the energy efficiency of the  $\Sigma\Delta$  learner supersedes that of a conventional multi-channel data converter. In fact, we estimate that power savings of more than 60% can be achieved using the proposed  $\Sigma\Delta$  learning architecture.

The power dissipation metrics of independent channels are summarized in Table II, where the metric for the channel  $j$  is calculated as

$$P_{Mod} + P_{DC} + (j - 1)(P_{NDC} + P_{CNT})\mu\text{W} \quad (28)$$

 TABLE I  
Measured specification of the 4-dimensional  $\Sigma\Delta$  learner

Parameters	Values
Technology	0.5 $\mu\text{m}$ 2P3M CMOS
Die Size	3 mm $\times$ 3 mm ( 4 channels )
Supply	3.3 V
Channels	2 $\times$ 4
Input Range	300mV
Input Frequency Bandwidth	4 kHz
Sampling Frequency	250 kHz
Over-sampling ratio (OSR)	32
SNR	53.81 dB
SNDR	52.68 dB
SFDR	60.72 dB
Total Power dissipation	1.5 mW at 250KHz ( 4 channel )
Active area of the 3 <sup>rd</sup> order $\Sigma\Delta$ modulator	850 $\mu\text{m}$ $\times$ 208 $\mu\text{m}$
Active area of the 4 dimensional system	2483 $\mu\text{m}$ $\times$ 876 $\mu\text{m}$

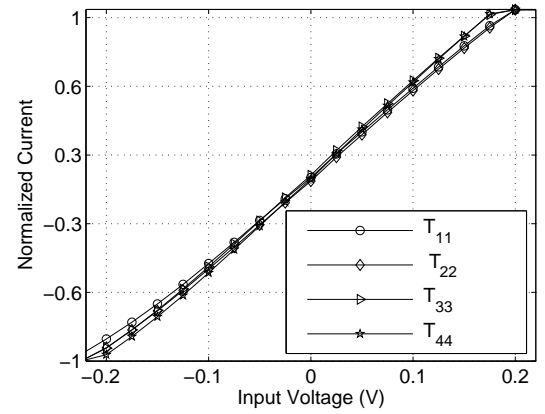


Fig. 20. Measured response of the source degenerated transconductors. The mismatch between the transconductors was determined to be less than 4%.

with  $P_{Mod}$  representing the power dissipation due to the modulator,  $P_{DC}$  represents the static power dissipation,  $P_{NDC}$  represents the power dissipation of a single transistor and  $P_{CNT}$  represents the power dissipation of the counter and current DACs.

 TABLE II  
Measured power dissipation of the  $\Sigma\Delta$  learner

(a) System Component Power	
3 <sup>rd</sup> order mixed mode Modulator ( $P_{Mod}$ )	231 $\mu\text{W}$
Diagonal cell ( $P_{DC}$ )	14.44 $\mu\text{W}$
Non-diagonal Cell ( $P_{NDC}$ )	16.5 $\mu\text{W}$
10-bit counter/shifter ( $P_{CNT}$ )	55 $\mu\text{W}$

(b) System Channel Power	
Channel 1	245.44 $\mu\text{W}$
Channel 2	331.38 $\mu\text{W}$
Channel 3	417.31 $\mu\text{W}$
Channel 4	503.25 $\mu\text{W}$

Fig. 20 shows the measured response of the transconductors used in the APUs. The currents produced by the transconductors were measured after decimating the output of the 3rd order modulator using a fourth-order digital sinc filter. The measured response show a linear operating range of 300mV

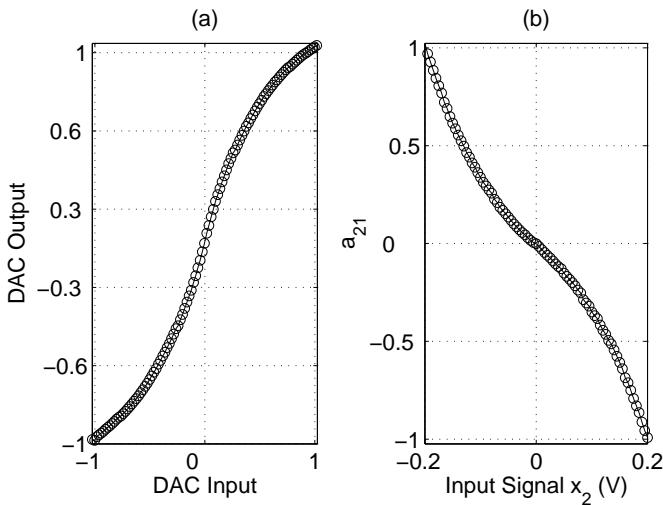


Fig. 21. (a) The non-linearity of the current DAC (b) value of  $a_{21}$  after learning.

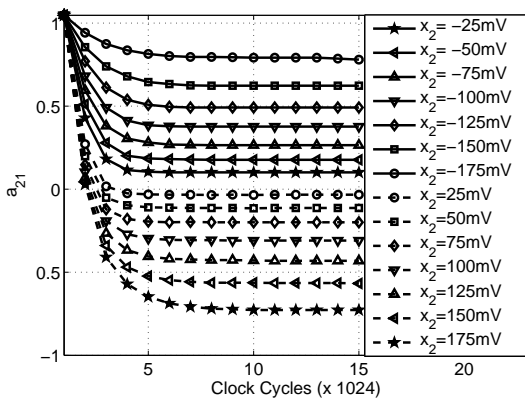


Fig. 22. Measured convergence of the parameter  $a_{21}$  for different magnitudes of  $x_2$  and a fixed magnitude of  $x_1$ .

with a worst case mismatch of 4%. Similar mismatch has been observed in the transconductors of the non-diagonal cells. In addition to mismatch, the non-diagonal cells also exhibit a non-linear response due to the current DACs as shown in Fig. 21(a). However, we have already shown using Monte-Carlo simulations that the mismatch and non-linear response can be compensated by the  $\Sigma\Delta$  learning algorithm. The self-calibrating and compensating ability of the  $\Sigma\Delta$  learner was verified using the following experimental set up. A DC signal of magnitude  $200mV$  was applied to the first channel  $x_1$  and the DC signal applied to the second channel  $x_2$  was varied in steps of  $5mV$  from  $-200mV$  to  $+200mV$ . For each of these values, the counters were first initialized to the maximum value of 2047 ( $+1023$ ) after which the  $\Sigma\Delta$  learner was run for 20000 clock cycles.

Fig. 22 shows the value of the parameter  $a_{21}$  obtained after each clock cycle showing a similar convergence response as observed in numerical simulations. Fig. 22 shows that the system is stable under the overload condition of input signals ( $x_1, x_2 > \pm 150mV$ ) exceeding the input range of the system.

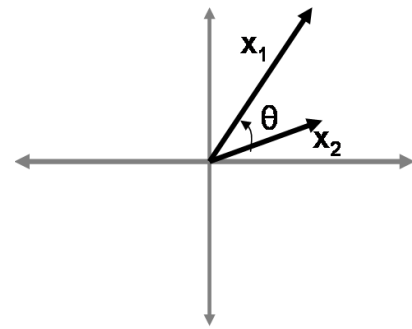


Fig. 23. A phasor diagram showing the two sinusoidal signals represented by vectors  $x_1, x_2$  differing by a phase  $\theta$ .

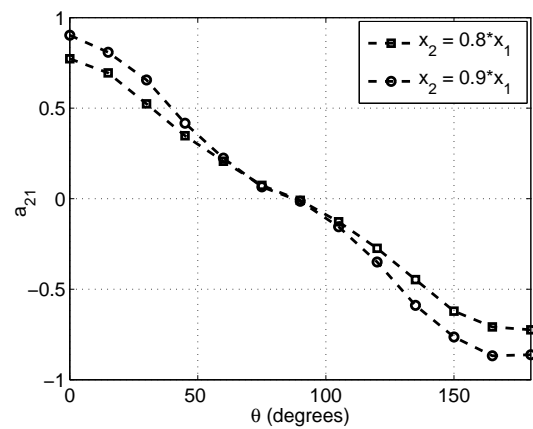


Fig. 24. The adapted value of the parameter  $a_{21}$  for different magnitude and the phase difference between the sinusoidal signals.

The stabilized value of the transform parameter  $a_{21}$  after 10000 adaptation cycles for all DC inputs of  $x_2$  are plotted in Fig. 21(b). The response is the inverse function of the DAC non-linearity showing that the  $\Sigma\Delta$  learner is able to compensate for the non-linearity of the DAC similar to the numerical experiments reported in Fig. 12.

The next set of experiments were used to verify the response of the  $\Sigma\Delta$  learner to identify redundant  $\Sigma\Delta$  modulator paths. For this experiment, two sinusoidal inputs of the same

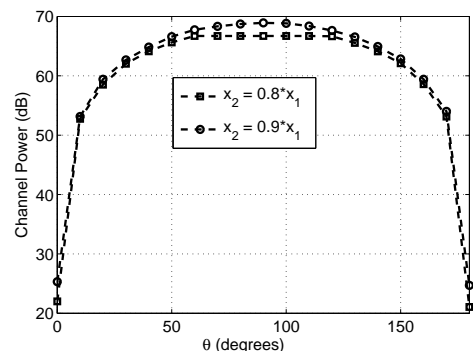


Fig. 25. The residual power on the second channel as the phase difference between the two signals is varied from  $0^\circ$  to  $180^\circ$

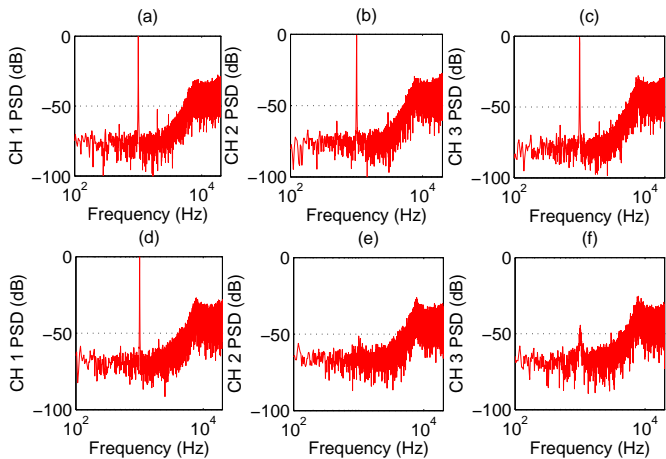


Fig. 26. Signal power in each channel before and after adaptation

frequency but different phase was presented as inputs to the  $\Sigma\Delta$  learner. For the sake of convenience the two sinusoidal signal are shown using a phasor diagram as shown in Fig. 23. The magnitude of the sinusoid is represented by the length of the vector and the phase difference between the sinusoids is represented by  $\theta$ . The phase and the magnitude of the second sinusoidal signal  $x_2$  was varied with respect to the first  $x_1$ . Fig. 24 shows the adapted value of  $a_{21}$  as the phase difference was varied from  $0^\circ$  to  $180^\circ$ . It can be clearly seen that the parameter  $a_{21}$  tracks the phase difference of the two signal and is minimum (zero) when the phase difference is  $90^\circ$ . The non-linear response of the  $a_{21}$  is due to the non-linearity of the DAC. The residual power at the output of the second modulator is shown in Fig. 25. It can be seen that the output is attenuated by more than  $50dB$  when the phase difference is  $0^\circ$  and  $180^\circ$ . This demonstrates that the  $\Sigma\Delta$  learner can identify the non-redundant subspace (whose rank is unity when the phasors are aligned with respect to each other) when  $\theta = 0^\circ$  and  $\theta = 180^\circ$ .

Fig. 26 shows the frequency domain response of the  $\Sigma\Delta$  learner when the adaptation is disabled and when the adaptation is enabled. The inputs of the system were applied with a  $1kHz$  sinusoidal signal and the  $FFT$  response of the  $\Sigma\Delta$  modulator outputs was observed. Plots (a),(b) and (c) of Fig. 26 show the power spectral density of 3 channels when the learning is disabled. Fig. 26(d), (e) and (f) show the  $FFT$  response of the  $\Sigma\Delta$  learner when the learning is enabled. In addition to showing the noise-shaping characteristics the plot shows that the  $\Sigma\Delta$  learner indeed suppresses the cross-channel redundancy. It must be noted that the signal transformation operation is a linear operation which is seen by observing no harmonics in the uncorrelated signal plots of Fig. 26(d), (e) and (f).

To validate that the output of the  $\Sigma\Delta$  learner can be used for reconstructing the input, the input signals were reconstructed using the modulator output and the digitized representation of the transform matrix  $\mathbf{A}$ . The reconstruction of the  $2^{nd}$  and the  $3^{rd}$  channel as a function of phase difference between signals  $x_1$  and  $x_2$  are shown in Fig. 27 (a) and (b) respectively.

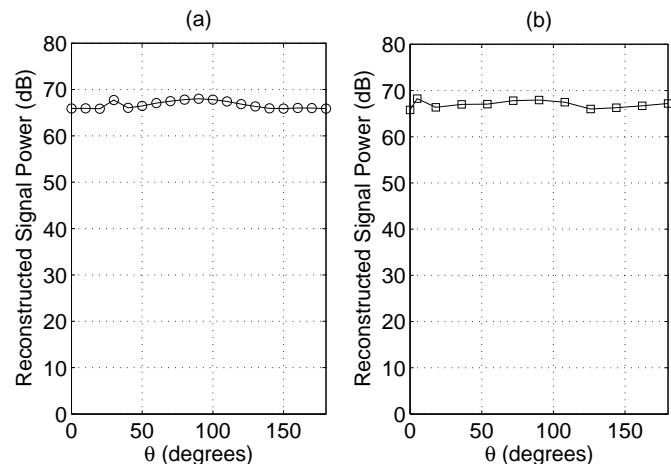


Fig. 27. The Reconstructed signal power for the (a)  $2^{nd}$  channel and (b) the  $3^{rd}$  channel.

The signal  $x_2$  was reconstructed with the help of transform parameter  $a_{21}$  and the modulator output of the first channel  $d_1$  where as the signal  $x_3$  was reconstructed with the help of parameters  $a_{31}$ ,  $a_{32}$  and the modulator outputs  $d_1$  and  $d_2$ . It can be seen that the reconstructed error less than  $0.5LSB$  can be achieved. The result demonstrates that the learned compression manifold  $\mathbf{A}$  faithfully captures the support of the input data which is consistent with the Monte-Carlo simulations.

## VII. DISCUSSIONS AND CONCLUSIONS

In this paper, we have presented an optimization framework that integrates manifold learning with  $\Sigma\Delta$  modulation. The framework produces not only a quantized sequence of transformed analog signals but also a quantized representation of the transform itself. The approach was shown to be applicable to higher-order modulations and also to different forms of analog signal compression. It was shown through extensive Monte-Carlo simulations and results obtained from a fabricated prototype that the proposed algorithm is robust to computational artifacts introduced by analog computation which includes mismatch and non-linearity, demonstrating that the approach could be effectively used for designing high-dimensional analog-to-digital converters. The future work in this area would entail optimization of the analog transforms in terms of area and power. One of the possible methods could be to incorporate sub-microwatt matrix-vector-multipliers as reported in [35] or to use charge-pump circuits to eliminate the area consuming counters. Reducing the area of the analog transforms is important since more input channels ( $D$ ) can be accommodated. We have already shown using equation (3) that increasing  $D$  would improve the performance of the  $\Sigma\Delta$  learner compared to a conventional architecture in terms of its energy efficiency. Also future work in this area would include extending  $\Sigma\Delta$  learning to multi-bit continuous time modulators [36] and to neurally inspired modulators which includes time-encoding machines [37].

### APPENDIX I

Let  $P_{\Sigma\Delta}$  denote the power dissipation of a single  $\Sigma\Delta$  modulator and  $P_{adt}$  denote the power dissipation required for mixed-signal multiplication and adaptation. During the first phase of the  $\Sigma\Delta$  learner operation (fraction  $\alpha$  of the total time), all the  $D$  modulators and  $D^2$  adaptation elements are active. Once the subspace of dimension  $M$  has been identified, the rest of the operation (fraction  $1-\alpha$  of the operational time),  $M$   $\Sigma\Delta$  modulators and  $MD$  matrix elements are active. The total power dissipation  $P_{STL}$  for operating the  $\Sigma\Delta$  learner is given by

$$P_{STL} = \alpha (DP_{\Sigma\Delta} + D^2P_{adt}) + (1 - \alpha) (MP_{\Sigma\Delta} + MDP_{adt}). \quad (29)$$

The estimated power dissipation for a conventional  $D$  channel data acquisition system (shown in Fig. 3 b) is given by

$$P_{conv} = DP_{\Sigma\Delta} + MDP_{DSP} \quad (30)$$

where  $P_{DSP}$  refers to the power dissipation for a single DSP operation used for estimating the transform  $\mathbf{A}_\infty$ . For the sake of simplicity we will ignore the power dissipation due to the DSP. To achieve superior power efficiency compared to the conventional system the relationship  $P_{STL} < P_{conv}$  needs to be satisfied which leads to

$$\frac{P_{adt}}{P_{\Sigma\Delta}} < \frac{(1 - \alpha)(1 - M/D)}{D[\alpha + (1 - \alpha)M/D]}. \quad (31)$$

For  $M \ll D$  and  $\alpha \ll 1$ , the relative power dissipation of the mixed-signal adaptation needs to satisfy

$$\frac{P_{adt}}{P_{\Sigma\Delta}} < \frac{1}{D(\alpha + M/D)}$$

which proves the inequality (3).

### APPENDIX II

To prove the claim in Lemma 2, that  $S = \|\mathbf{w}[n-1] - \lambda \mathbf{d}[n]\|_\infty < \lambda$ , given that  $\|\mathbf{w}[n-1]\|_\infty < 2\lambda$ , we will use the relationship  $\mathbf{d}[n] = \text{sgn}(\mathbf{w}[n-1])$ . The following relationships will prove the claim

$$S = \|\mathbf{w}[n-1] - \lambda \text{sgn}(\mathbf{w}[n-1])\|_\infty \quad (32)$$

$$\leq \|\text{sgn}(\mathbf{w}[n-1])\|_\infty (\|\mathbf{w}[n-1]\|_\infty - \lambda) \quad (33)$$

$$\leq \|\text{sgn}(\mathbf{w}[n-1])\|_\infty \lambda \quad (34)$$

$$= \lambda \quad (35)$$

where we have used the equality  $\|\text{sgn}(\mathbf{w}[n-1])\|_\infty = 1$ .

### APPENDIX III

To prove the expression (22), we start with the recursions for a second order  $\Sigma\Delta$  learner given by

$$\begin{aligned} \mathbf{w}[n] &= \mathbf{w}[n-1] \\ &+ (\mathbf{A}[n-1]\mathbf{x}[n-1] - \lambda \mathbf{d}[n]) \\ &+ \mathbf{w}[n-1] - \mathbf{w}[n-2] \end{aligned} \quad (36)$$

Let  $\mathbf{v}[n] = \mathbf{w}[n] - \mathbf{w}[n-1]$ , therefore, the recursion is written as

$$\mathbf{v}[n] = \mathbf{v}[n-1] + (\mathbf{A}[n-1]\mathbf{x}[n-1] - \lambda \mathbf{d}[n]) \quad (37)$$

Assuming the initial condition of  $\mathbf{v}[0] = 0$  and solving the discrete time recursion leads to

$$\frac{\lambda}{N} \sum_{n=1}^N \mathbf{d}[n] = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{A}[n]\mathbf{x}[n] + \frac{1}{N} \mathbf{v}[n] \quad (38)$$

Since  $\mathbf{v}[n]$  is finite and bounded, therefore asymptotically we get

$$\frac{\lambda}{N} \sum_{n=1}^N \mathbf{d}[n] = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{A}[n]\mathbf{x}[n] \quad (39)$$

Therefore, the empirical expectation  $\mathcal{E}_n(\cdot)$  of both sides give

$$\mathcal{E}_n\{\mathbf{d}[n]\} = \frac{1}{\lambda} \mathcal{E}_n\{\mathbf{A}[n]\mathbf{x}[n]\} \quad (40)$$

Substituting  $\mathbf{v}[n] = \mathbf{w}[n] - \mathbf{w}[n-1]$  we get

$$\mathcal{E}_n\{\mathcal{E}_n\{\mathbf{d}[n]\}\} = \frac{1}{\lambda} \mathcal{E}_n\{\mathbf{A}[n]\mathbf{x}[n]\} + \frac{1}{N} \mathbf{w}[n] \quad (41)$$

The finite and bounded nature of  $\mathbf{w}[n]$  gives

$$\mathcal{E}_n\{\mathcal{E}_n\{\mathbf{d}[n]\}\} = \frac{1}{\lambda} \mathcal{E}_n\{\mathbf{A}[n]\mathbf{x}[n]\} \quad (42)$$

### REFERENCES

- [1] K.D. Wise, D. J. Anderson, J. F. Hetke, D. R. Kipke, and K. Najafi, "Wireless implantable microsystems: High-density electronic interfaces to the nervous system," *Proceedings of the IEEE*, vol.92(1), pp.138-145, Jan 2004.
- [2] C. T. Nordhausen, E. M. Maynard, and R. A. Normann, "Single unit recording capabilities of a 100-microelectrode array," *Brain Research*, vol. 726, pp. 129140, 1996.
- [3] Harrison, R. R.; Watkins, P. T.; Kier, R. J.; Lovejoy, R. O.; Black, D. J.; Greger, B.; Solzbacher, F.; "A Low-Power Integrated Circuit for a Wireless 100-Electrode Neural Recording," *IEEE Journal of Solid-State Circuits*, vol. 42(1), pp.123 - 133, Jan 2007.
- [4] J. E. Greenberg and P. M. Zurek, "Microphone array hearing aids in microphone arrays: Signal Processing Techniques and Application," Springer, Berlin, pp. 229-253, 2001.
- [5] R.N. Miles and R.R. Hoy; "The development of a biologically-inspired directional microphone for hearing aids," *Audiology and Neuro-Otology* 11 (2) : 86-94, 2006.
- [6] D. W. Henderson, "Experiencing Geometry: In Euclidean, Spherical, and Hyperbolic Spaces," Prentice-Hall, 2001.
- [7] S. Haykin, *Neural Networks: A comprehensive foundation*. Prentice Hall, 2nd edition, 1998.
- [8] T. Kohonen, *Self-organizing Maps*. Springer publications 2001.
- [9] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science* 22, Vol. 290. no. 5500, pp. 2323-2326, Dec. 2000.
- [10] D. Donoho and C. Grimes, "Hessian eigenmaps: new tools for nonlinear dimensionality reduction," *In Proceedings of National Academy of Science*, pp. 5591-5596, 2003.
- [11] J. C. Candy and G. C. Temes, "Oversampled methods for A/D and D/A conversion in Oversampled Delta-Sigma Data Converters," Piscataway, NJ: IEEE Press, 1992, pp. 1- 29.
- [12] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [13] G. Cauwenberghs and M. Bayoumi, *Learning on Silicon*. Kluwer Academic Publishers, Boston 1999.
- [14] D.L. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, Volume 52, Issue 4, pp.1289 - 1306, April 2006.
- [15] J.N Laska, S. Kirolos, M.F. Duarte, T.S. Ragheb, R.G. Baraniuk, Y. Massoud, "Theory and implementation of an analog-to-information converter using random demodulation," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1959 - 1962, May 2007.
- [16] A. Gore and S. Chakraborty, "Large margin analog-to-digital converters with applications in neural prosthetics," *Adv. Neural Information Processing Systems (NIPS'2006)*.
- [17] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

- [18] Girosi, F., Jones, M. and Poggio, T. *Regularization Theory and Neural Networks Architectures*. Neural Computation, vol. 7, pp 219-269, 1996.
- [19] D. Fudenberg and D. Levine, *The Theory of Learning in Games*. MIT Press, 1998.
- [20] T. Basar and P. Bernhard, *H<sup>∞</sup> Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Springer 1995.
- [21] W. Rudin, *Functional Analysis*. New York: McGraw-Hill, 1991.
- [22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [23] D.E. Rumelhart, G.E. Hinton, R. J. Williams, "Learning internal representations by error propagation," Institute for Cognitive Science, University of California, San Diego, 1986. In D.E. Rumelhart and J.L. McClelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 318-362). Cambridge, MA: MIT Press.
- [24] B. Pearlmutter, "Gradient descent: second order momentum and saturating error," In *Advances in Neural Information Processing Systems (NIPS)*, pp 887-894, 1992.
- [25] P.M. Aziz, H.V. Sorensen, J. van der Spiegel, "An overview of sigma-delta converters," *IEEE Signal Processing Magazine*, Volume 13, Issue 1, pp. 61-84, Jan. 1996.
- [26] I. Galton and H. T. Jensen, "Delta-Sigma Modulator Based A/D Conversion without Oversampling," *IEEE Transactions on Circuits and Systems II*, Volume 42, Issue 12, pp. 773 - 784, Dec. 1995.
- [27] I. Galton and H.T. Jensen, "Oversampling parallel delta-sigma modulator A/D conversion," *IEEE Transactions on Circuits and Systems II*, Volume 43, Issue 12, pp. 801-810, Dec. 1996.
- [28] R. D. Batten, A. Eshraghi, T. S. Fiez, "Calibration of parallel  $\Delta\Sigma$  ADCs," *IEEE Transactions on Circuits and Systems II*, Vol.49(6), pp. 390-399, June 2002.
- [29] V. Ferragina, A. Fornasari, U. Gatti, P. Malcovati, F. Maloberti, "Gain and offset mismatch calibration in time-interleaved multipath A/D sigma-delta modulators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 51(12), pp. 2365 - 2373, Dec. 2004.
- [30] B. Linares-Barranco and T. Serrano-Gotarredona, "On the design and characterization of femtoampere current-mode circuits", *Solid-State Circuits*, *IEEE Journal of* Volume 38, Issue 8, Aug. 2003 Page(s):1353 - 1363
- [31] P.M. Furth and H.A. Ommani. "Low-voltage highly-linear transconductor design in subthreshold CMOS". *Circuits and Systems*, 1997. *Proceedings of the 40th Midwest Symposium on* Volume 1, 3-6 Aug. 1997 Page(s):156 - 159 vol.1
- [32] Kulchyski, S.D.; Trofin, R.; Vleugels, K.; Wooley, B.A.; "A 77-dB Dynamic Range, 7.5-MHz Hybrid Continuous-Time/Discrete-Time Cascaded *SigmaDelta* Modulator," *IEEE Journal of Solid-State Circuits*, Volume 43, Issue 4, April 2008 Page(s):796 - 804
- [33] G. Cauwenberghs and V. Pedroni, "A Charge-Based CMOS Parallel Analog Vector Quantizer," *Adv. Neural Information Processing Systems (NIPS\*94)*, Cambridge, MA: MIT Press, vol. 7, pp. 779-786, 1995.
- [34] I. H. Witten, E. Frank, *Data Mining*. Academic Press, 2005.
- [35] S. Chakrabarty, G.Cauwenberghs, "A Sub-microwatt Analog VLSI Trainable Pattern Classifier", *IEEE Journal of Solid-State Circuits*, vol. 42, no: 5, May 2007.
- [36] J. De Maeyer, P. Rombouts, and L. Weyten, "Efficient multibit quantization in continuous-time  $\Sigma\Delta$  modulators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 4, pp. 757767, Apr. 2007.
- [37] A.A. Lazar, E. K. Simonyi, L.T. Toth, "An Overcomplete Stitching Algorithm for Time Decoding Machines," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 9, pp. 2619-2630, Oct. 2008.



**Shantanu Chakrabarty** (M'96) received his B.Tech degree from Indian Institute of Technology, Delhi in 1996, M.S and Ph.D in Electrical Engineering from Johns Hopkins University, Baltimore, MD in 2001 and 2004 respectively. He is currently an assistant professor in the department of electrical and computer engineering at Michigan State University. From 1996-1999 he was with Qualcomm Incorporated, San Diego and during 2002 he was a visiting researcher at University of Tokyo. His current research interests include low-power analog and digital VLSI systems, hardware implementation of machine learning algorithms with application to biosensors and biomedical instrumentation. Dr. Chakrabarty was a recipient of The Catalyst foundation fellowship from 1999-2004 and received the best undergraduate thesis award in 1996. He is currently a member for IEEE BioCAS technical committee, IEEE Circuits and Systems Sensors technical committee and serves as an associate editor for *Advances in Artificial Neural Systems* from Hindawi publications.



**Amit Gore** received the Bachelor's (B.E.) degree in instrumentation engineering from the University of Pune, Pune, India, in 1998, and the M.S. degree and Ph.D degree in electrical and computer engineering from Michigan State University, East Lansing, in 2002 and 2008 respectively. Currently, he is with General Electric Global Research, Niskayuna, New York. His research interests are low-power sigma-delta converters, analog signal processing and low-power mixed-signal VLSI design.