

# Earth Mover's Distance as a Comparison Metric for Analog Behavior

Alexander Rath, Sebastian Simon, Volkan Esen, Wolfgang Ecker

► **To cite this version:**

Alexander Rath, Sebastian Simon, Volkan Esen, Wolfgang Ecker. Earth Mover's Distance as a Comparison Metric for Analog Behavior. 24th IFIP/IEEE International Conference on Very Large Scale Integration - System on a Chip (VLSISOC), Sep 2016, Tallinn, Estonia. pp.173-191, 10.1007/978-3-319-67104-8\_9. hal-01675194

**HAL Id: hal-01675194**

**<https://hal.inria.fr/hal-01675194>**

Submitted on 4 Jan 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Earth Mover's Distance as a Comparison Metric for Analog Behavior

Alexander W. Rath, Sebastian Simon, Volkan Esen, and Wolfgang Ecker

Infineon Technologies AG  
*firstname.lastname@infineon.com*

**Abstract.** Evaluating the outcome of analog simulations is a common, mostly manually carried out task in the pre-silicon verification process of mixed-signal ICs. Its non-automated nature makes it an error-prone and time-consuming procedure. For this very reason, we introduce a novel approach for performing this evaluation automatically resulting in significantly reduced turnaround times as well as a considerably increased reliability of verification results. The presented concept is motivated by an algorithm that is used in optical pattern recognition and is called Earth Mover's Distance. Furthermore, we compare our approach with already existing algorithms, namely Fréchet Distance and Pearson Coefficient, in order to analyze its capability. Finally, we present a case study in which we prove the algorithm by applying it to the results of a mixed-signal simulation at chip-level demonstrating the efficiency of our approach.

**Keywords:** Mixed-signal verification, Analog behavior, Earth mover's distance, Pearson correlation coefficient, Fréchet distance

## 1 Introduction

Due to better technology scaling of digital blocks compared to analog blocks more and more parts of the analog implementation of modern IC designs are shifted to the digital domain. However, certain analog components are still indispensable, leading to mixed-signal designs. Hence, the demand for comprehensively verifying such designs is continuously growing and compared to the highly automated verification methodologies in the digital domain, pre-silicon verification in the analog domain usually implies a substantial amount of manual work and computational effort.

On analog module level, pre-silicon verification primarily targets the analysis of electrical characteristics as for instance gain, distortion or input resistance. Although determining electrical characteristics is also targeted on chip top and system level, confirming the functional correctness and the application fitness of the design by means of simulation is the main goal in this regard. The authors of [1], e. g., point out that it is still a common practice to manually inspect waveforms when performing the latter. This effectively obstructs the usage of

automated checking and regression-based simulations inevitably leading to an increased error-proneness as well as vast expenditures of time. Therefore, they advise to leverage automated solutions to overcome these problems. We were also able to identify these difficulties and we still see room for improvement in this field. For that reason we set ourselves the goal to offer an approach which enables an automated way of comparing analog behavior and allows mixed-signal system level functionality to be verified in a regression-based way.

The contribution is structured as follows. First, we give an overview of related work that covers approaches for comparing analog behavior. Their results form the basis for the motivation of our work. Following this, we explain the theoretical principles of our developed approach. In Section 4 we select two competitive algorithms and define a number of comparison criteria in order to show the strengths of our approach. Finally, a case study is outlined in which we point out the achieved benefits.

## 2 Related Work

If analog behavior of a design shall be checked, one of the most common approaches is to make use of analog assertions. There are different works [2][3] which successfully apply this technique for mixed-signal verification. However, assertions generally lack in abstraction since they have to be defined at signal level. Apart from that, they can usually be applied to check only one particular property of the behavior and are therefore highly application-specific.

Further works, which deal with this topic, are presented in [4][1][5]. They propose approaches which are based on application-specific checkers. These checkers continuously monitor and evaluate circuit characteristics like, for instance, open-loop gain or output noise. However, their solutions require a recurring effort for the development of checkers since they have to be elaborated and implemented for each design characteristic.

Due to the aforementioned deficiencies it is necessary to look for more generic approaches. For this reason we investigated several metrics that are capable of determining the similarity between discrete signals. One frequently used metric is, for instance, the Pearson correlation coefficient (PC). The authors of [6] demonstrate how to employ this measure for the purpose of mixed-signal verification. An overview of other potential metrics like quadratic mean error or Fréchet distance (FD) can be moreover found in [7]. Their results marked the starting point of our work, in which we strove for finding a better suited similarity measure. This newly developed approach is presented and assessed in the subsequent sections.

## 3 Earth Mover's Distance

The algorithm we propose is based on the so-called earth mover's distance (EMD). This measure is widely used in the field of content-based image retrieval and is capable to determine the degree of similarity between two distributions.

An essential part of this algorithm is the repetitive application of a ground distance. Therefore, we first show how to construct an appropriate ground distance before dwelling on the overall concept of our EMD approach.

### 3.1 Ground Distance

In order to quantify the similarity of analog signals it is necessary to operate on discrete-time signals. A discrete-time signal  $x[i] \in \mathbb{R}$  is a series of data with the discrete time  $i \in \mathbb{N}_0$  and  $i < n$ , where  $n \in \mathbb{N}$  is the number of sample points, i. e. the length of the signal.

In this publication, however, we rely on an alternative definition that regards the signal  $x[i]$  as a vector  $\mathbf{x} \in \mathbb{R}^n$ :

$$\begin{aligned}\mathbf{x} &= (x_0, \dots, x_i, \dots, x_{n-1}) \\ &= (x[0], \dots, x[i], \dots, x[n-1])\end{aligned}$$

For our ground distance  $\delta_G(x_i, y_j)$ , we claim the following fundamental requirement:

The ground distance should be *bounded* to the interval  $[0, 1]$ , as it can be shown that a bounded ground distance is a prerequisite for the EMD to be bounded as well.

Several publications [8][9][10] utilize the Euclidean distance  $|x_i - y_j|$  as ground distance. This measure is obviously not compatible with the requirement of boundedness. However, there are methods to subsequently bound such distances, e. g. one could use a division by a parameter that incorporates information about maximum and minimum signal values. Alternatively, one could apply an appropriate function like hyperbolic tangent to it. All these methods have the crucial drawback that the resulting distance functions still treat pairs  $(x_i, y_j)$  of big values differently than pairs of small values, although the ratios of the pairs are the same. For example consider

$$\begin{aligned}\frac{x_0 = 1000}{y_0 = 900} &= \frac{x_1 = 10}{y_1 = 9}, \text{ but} \\ \tanh |x_0 - y_0| &\approx 1 \neq \tanh |x_1 - y_1| \approx 0.1\end{aligned}$$

This phenomenon effectively cancels out all sections of the involved signals where the signal's values have a relatively small magnitude. Thus, we claim another requirement:

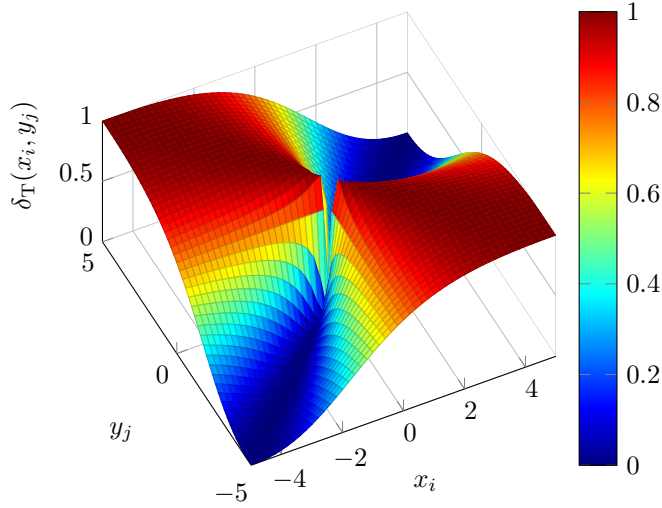
The magnitude of the ground distance's arguments should not affect the result, i. e. the ground distance has to be *scale-invariant*, such that the following property holds:

$$\delta_G(x_i, y_j) = \delta_G(ax_i, ay_j), \quad a \in \mathbb{R}^* \tag{1}$$

In order to fulfill this requirement, we drop the Euclidean distance and use a scaled version of the Tanimoto distance [11][12] instead:

$$\delta_{\Gamma}(x_i, y_j) = \begin{cases} \frac{-0.75 \cdot x_i \cdot y_j}{x_i^2 + y_j^2 - x_i \cdot y_j} + 0.75 & \text{if } (x_i, y_j) \neq (0, 0) \\ 0 & \text{else} \end{cases} \quad (2)$$

A 3D-Plot of the function is shown in Fig. 1.



**Fig. 1.** Tanimoto distance

There are a number of important properties to be mentioned about the scaled Tanimoto distance: It is limited to the interval  $[0, 1]$  and becomes 0, whenever  $x_i = y_j$ . Also, it becomes 1 if the arguments are equivalent in magnitude, but differ in sign (for  $x_i \neq 0$ ):  $\delta_{\Gamma}(x_i, -x_i) = 1$ .

Furthermore, it is continuous for  $(x_i, y_j) \neq (0, 0)$  and it is scale-invariant, i. e. the distance does not depend on the order of magnitude of the arguments. Therefore it satisfies Equation 1.

Since the scaled Tanimoto distance does not take the position of the elements  $x_i$  and  $y_j$  into account, we need to extend it. This extension is done by calculating the difference between the positions normalized to the biggest possible difference  $n - 1$ . The extension and the scaled Tanimoto distance are then combined using the hypot function. Thus, the ground distance is defined as follows:

$$\delta_G(x_i, y_j) = \frac{1}{\sqrt{2}} \cdot \sqrt{\delta_{\Gamma}(x_i, y_j)^2 + \frac{(i - j)^2}{(n - 1)^2}} \quad (3)$$

The division by  $\sqrt{2}$  is required to retain the limitedness to the interval  $[0, 1]$ . Note that this expression can only be applied for  $n > 1$ . For the very unusual case that  $n = 1$  we set  $\delta_G(x_i, y_j) = \delta_T(x_i, y_j)$ .

### 3.2 Algorithmic Concept

The earth mover's distance is an approach for measuring the distance between two multi-dimensional distributions. Informally, if one must successively transport soil from one pile to another in order to equalize them, the earth mover's distance calculates the minimum cost for the total transport. The distributions are also called *signatures* and represent weighted feature vectors of both signals  $\mathbf{x}$  and  $\mathbf{y}$ . In the following the concept of this approach, which is based on [9], shall be explained.

Given are two signatures  $\mathcal{X}$  and  $\mathcal{Y}$  where  $x_i, y_j$  denote elements from the respective feature vectors  $\mathbf{x}$  and  $\mathbf{y}$ . Each signature consists of  $n$  clusters.

$$\begin{aligned}\mathcal{X} &= \{(x_0, w_{x_0}), \dots, (x_{n-1}, w_{x_{n-1}})\} \\ \mathcal{Y} &= \{(y_0, w_{y_0}), \dots, (y_{n-1}, w_{y_{n-1}})\}\end{aligned}$$

For each element  $x_i, y_j$  a particular weight  $w_{x_i}, w_{y_j}$  can be assigned. In our case a constant weight  $W$  is sufficient since no element shall be prioritized:

$$W = w_{x_i} = w_{y_j} = \frac{1}{n} \quad 0 \leq i \leq n-1, 0 \leq j \leq n-1 \quad (4)$$

Moreover the possibility for transports between both signatures is defined as *flow*  $\mathbf{F} = [f_{ij}]$ , with  $f_{ij}$  the flow between  $x_i$  and  $y_j$ .

Now the idea is to find a flow  $\mathbf{F}$  that minimizes the costs  $C$  for the overall work:

$$C(\mathbf{x}, \mathbf{y}, \mathbf{F}) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \delta_G(x_i, y_j) \cdot f_{ij} \quad (5)$$

For our ground distance we chose the combined distance from Equation 3 since this distance can be used for scalars and is bounded above and below.

The optimization problem in Equation 5 must furthermore satisfy the following constraints where constraint 7–9 can be simplified according to our assumption in Equation 4:

$$f_{ij} \geq 0 \quad 0 \leq i \leq n-1, 0 \leq j \leq n-1 \quad (6)$$

$$\sum_{j=0}^{n-1} f_{ij} \leq w_{x_i} = \frac{1}{n} \quad 0 \leq i \leq n-1 \quad (7)$$

$$\sum_{i=0}^{n-1} f_{ij} \leq w_{y_j} = \frac{1}{n} \quad 0 \leq j \leq n-1 \quad (8)$$

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} f_{ij} = \min \left[ \sum_{i=0}^{n-1} w_{x_i}, \sum_{j=0}^{n-1} w_{y_j} \right] = 1 \quad (9)$$

Constraint 6 allows only unidirectional flows from  $\mathcal{X}$  to  $\mathcal{Y}$ . Constraint 7 ensures that the weight in  $\mathcal{Y}$  matched to  $x_i$  does not exceed  $n^{-1}$ , while constraint 8 ensures that also the weight in  $\mathcal{X}$  matched to  $y_j$  does not exceed  $n^{-1}$ . Finally, constraint 9 forces the total amount of weight matched to be equal to the overall weight of each signature.

In summary this particular metric consists in solving the following transportation problem for a pair of distributions on condition that the above-stated constraints are satisfied:

$$d_{\text{EM}}(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{F}=[f_{ij}]} C(\mathbf{x}, \mathbf{y}, \mathbf{F}) \quad (10)$$

This expression represents a linear optimization problem. Hence, we can solve it by an appropriate algorithm like the simplex method. However, this method shows an exponential worst-case but polynomial average complexity [13], which in turn has a determining influence on the overall performance of this measure. The results for  $d_{\text{EM}}$  eventually yield values within the interval  $[0, 1]$  where 1 symbolizes complete dissimilarity and 0 a full match.

## 4 Assessment

In order to assess the efficiency of EMD, we compare its behavior with two other algorithms mentioned in Section 2; the Pearson correlation coefficient and the Fréchet distance.

In this section we describe how we carried out the assessment and discuss its results. The assessment setup consists of a set of deviation algorithms that are used to generate distorted variants of a given source signal as well as implementations of the EMD, the PC and the FD that compare the source signal with its distorted variants. In this way, we can analyze the behavior of the different algorithms with respect to various types of signal deviations.

### 4.1 Deviations

A deviation  $D(x_i, p)$  is defined as a function that maps the elements of a signal  $\mathbf{x}$  to the elements of a signal  $\mathbf{y}$  of the same length:

$$y_i = D(x_i, p), \quad (11)$$

where  $p \in \mathbb{R}$  is the severity of the deviation.

In addition, we introduce the following symbols for the sake of readability: The amplitude  $\hat{x}$  of a signal  $\mathbf{x}$

$$\hat{x} = \max_i |x_i|, \quad (12)$$

and the swing  $\Delta x$  of a signal  $\mathbf{x}$ <sup>1</sup>

$$\Delta x = \max_i x_i - \min_i x_i \quad (13)$$

---

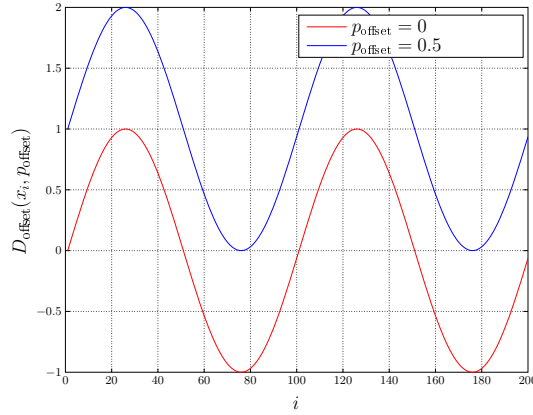
<sup>1</sup> Please note that we assume that  $\mathbf{x}$  is not a constant signal, i. e.  $\Delta x \neq 0$  and hence  $\hat{x} \neq 0$ . We make this limitation as constant signals would unnecessarily complicate

**Offset** The first deviation to be defined here is the offset. An offset is a constant, global shift of the source signal along the signal axis. Parameter  $p_{\text{offset}}$  specifies the severity of the offset:

$$D_{\text{offset}}(x_i, p_{\text{offset}}) = x_i + p_{\text{offset}} \cdot \Delta x \quad (14)$$

In this definition, the actual offset is  $p_{\text{offset}} \cdot \Delta x$ . This causes  $p_{\text{offset}}$  to be relative to the swing of the signal. The advantage of this definition is that an offset with, say,  $p_{\text{offset}} = 1$  is always significant. This allows for comparing different values of  $p_{\text{offset}}$  independent of the actual signal.

Fig. 2 depicts the signal change in case of an offset deviation.



**Fig. 2.** Sine signal with deviations in offset.

**Edge Steepness** This deviation describes a change in the edge steepness of the signal while retaining the signal's original characteristics:

$$D_{\text{edge}}(x_i, p_{\text{edge}}) = \begin{cases} \frac{\hat{x} \cdot \tanh \frac{p_{\text{edge}} \cdot x_i}{\hat{x}}}{\tanh p_{\text{edge}}} & \text{if } p_{\text{edge}} > 0 \\ \frac{\hat{x} \cdot \text{artanh} \frac{p_{\text{edge}} \cdot x_i}{\hat{x}}}{\text{artanh } p_{\text{edge}}} & \text{if } -1 < p_{\text{edge}} < 0 \\ x_i & \text{if } p_{\text{edge}} = 0 \end{cases} \quad (15)$$

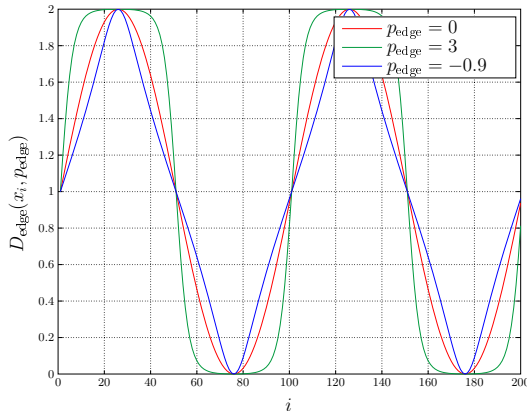
The applied changes to the signal can be regarded as a three-step process: The values of  $x$  are scaled to the interval  $[-1, 1]$ . Then, the signal is compressed using  $\tanh$  or  $\text{artanh}$ , respectively, and finally, it is rescaled to the original scale.

---

some of the deviation definitions that we give in the following subsections without providing any further insight. For example, consider Equation 14, where  $\Delta x = 0$  would cancel out the effect of  $p_{\text{offset}}$ .



The first branch using  $\tanh$  causes the signal's edges to become more convex, while the second branch using  $\operatorname{artanh}$  causes the edges to become more concave. Neither of both branches is defined for  $p_{\text{edge}} = 0$ , but it can be shown that  $\lim_{p_{\text{edge}} \rightarrow 0} D_{\text{edge}} = x_i$  for both branches. Hence, it makes sense to define a third branch  $D_{\text{edge}} = x_i$  for  $p_{\text{edge}} = 0$ , leaving the signal as it is. Examples are illustrated in Fig. 3.



**Fig. 3.** Sine signal with deviations in edge steepness.

**Delay** The delay is a constant, global shift of the signal along the time axis. When shifting a signal along the time axis, it is necessary to fill the emerging gap. Here we have chosen to fill the gap with the value  $x_0$  leading to an acyclic delay. Thus, we define the delay for  $0 \leq p_{\text{delay}} \leq 1$  as follows:

$$D_{\text{delay}}(x_i, p_{\text{delay}}) = \begin{cases} x_{i-n \cdot p_{\text{delay}}} & \text{if } i \geq n \cdot p_{\text{delay}} \\ x_0 & \text{else} \end{cases} \quad (16)$$

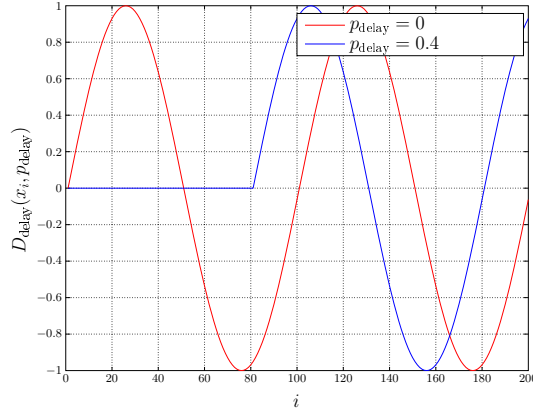
The severity  $p_{\text{delay}}$  is relative to the signal's length  $n$ .

An example for delay deviation can be found in Fig. 4.

**Scaling** Scaling means that the signal is vertically stretched or compressed through multiplying its function by a constant factor  $p_{\text{scale}}$ :

$$D_{\text{scale}}(x_i, p_{\text{scale}}) = x_i \cdot p_{\text{scale}} \quad (17)$$

Fig. 5 shows the effect of adding scale to a signal.



**Fig. 4.** Sine signal with deviations in delay.

**Peak** In contrast to any other deviations presented here, a peak is only a local phenomenon. For  $n$  being odd, we define the peak as single spike in the middle of the signal:

$$D_{\text{peak}}(x_i, p_{\text{peak}}) = \begin{cases} x_i + p_{\text{peak}} \cdot \Delta x & \text{if } i = \lfloor \frac{n}{2} \rfloor \\ x_i & \text{else} \end{cases} \quad (18)$$

For  $n$  being even, we vary the two adjacent points in the middle of the signal:

$$D_{\text{peak}}(x_i, p_{\text{peak}}) = \begin{cases} x_i + p_{\text{peak}} \cdot \Delta x & \text{if } i \in \{ \frac{n}{2} - 1, \frac{n}{2} \} \\ x_i & \text{else} \end{cases} \quad (19)$$

Examples for peak deviations are illustrated in Fig. 6.

**Noise** In principal, adding noise to a signal is like adding offset, except that the offset is now random over time rather than constant:

$$D_{\text{noise}}(x_i, p_{\text{noise}}) = x_i + u_i, \quad (20)$$

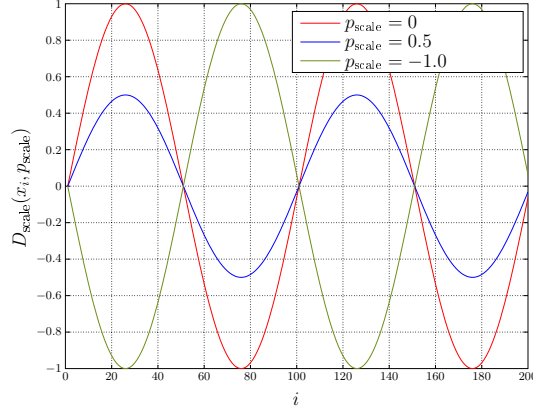
where  $u_i$  is a uniformly distributed random number with

$$|u_i| \leq \frac{p_{\text{noise}} \cdot \Delta x}{2}. \quad (21)$$

An example for noise deviation is provided in Fig. 7.

## 4.2 Comparable Algorithms

In this section, we present two additional algorithms that we use to evaluate our EMD approach.



**Fig. 5.** Sine signal with deviations in scale.

**Pearson Correlation Coefficient** The Pearson correlation coefficient is defined as

$$s_{\text{pea}}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=0}^{n-1} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n-1} (x_i - \bar{x})^2 \cdot \sum_{i=0}^{n-1} (y_i - \bar{y})^2}}, \quad (22)$$

where  $\bar{x}$  and  $\bar{y}$  represent the arithmetic mean:

$$\bar{x} = \frac{1}{n} \sum_{i=0}^{n-1} x_i \quad \text{and} \quad \bar{y} = \frac{1}{n} \sum_{j=0}^{n-1} y_j$$

The result of  $s_{\text{pea}}$  is a value in the interval  $[-1, 1]$ , where 1 indicates a full match, 0 a total mismatch and  $-1$  a full match except the sign.

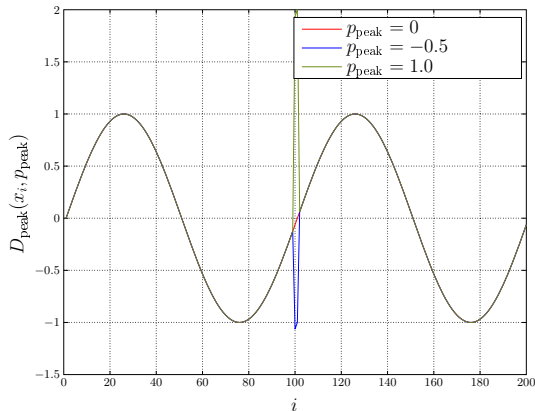
It can be moreover shown that there is a direct relation between Pearson correlation coefficient and normalized cross-correlation. The latter is given in [14] and defined as

$$(\mathbf{x} \star \mathbf{y})_{\text{norm}}(j) = \frac{\sum_{i=0}^{n-j-1} (x_i - \bar{x}) \cdot (y_{i+j} - \bar{y})}{\sqrt{\sum_{i=0}^{n-1} (x_i - \bar{x})^2 \cdot \sum_{i=0}^{n-j-1} (y_{i+j} - \bar{y})^2}} \quad (23)$$

With the help of this equation it can be seen that Pearson correlation coefficient is equivalent to normalized cross-correlation at position  $j = 0$ :

$$s_{\text{pea}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \star \mathbf{y})_{\text{norm}}(0)$$

In other words, an evaluation of the Pearson correlation coefficient additionally embraces an analysis of normalized cross-correlation, which in turn is a widely used measure of similarity in the field of signal processing.



**Fig. 6.** Sine signal with deviations in peak.

**Fréchet Distance** For discrete time signals, the Fréchet distance can be defined as

$$d_F(\mathbf{x}, \mathbf{y}) = \min_{\alpha, \beta} \max_k \delta_G(x_{\alpha(k)}, y_{\beta(k)}), \quad (24)$$

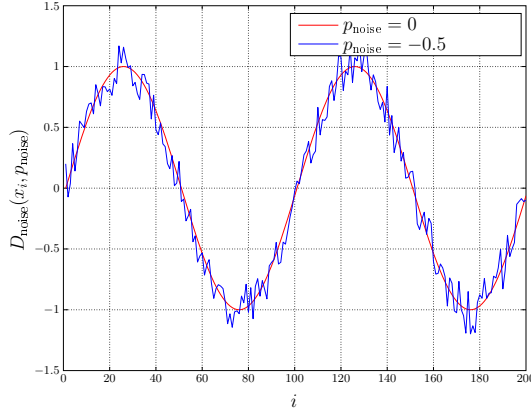
where  $\alpha(k)$  and  $\beta(k)$  are non-decreasing surjections  $k \in \mathbb{N}_0 \mapsto [0, n - 1]$ . Values for  $d_F$  are in the same interval as for  $\delta_G$ . Since we use the ground distance  $\delta_G$  as given in Equation 3, the interval is  $[0, 1]$  where 0 indicates a full match and 1 a total mismatch.

The definition in Equation 24 is not very intuitive, which is why a more intelligible explanation featuring a dog and its owner has become popular [10]: The Fréchet distance is the shortest length of a leash required for a dog and its owner to walk on two routes—represented by our signals—from one endpoint to the other. Both are allowed to change their speed, but they are not allowed to walk back.

Furthermore, the definition does not indicate how to implement the Fréchet distance. Luckily, [15] provides a pseudo implementation for discrete-time signals.

### 4.3 Results

The results of the assessment are shown in Fig. 8. In order to make the metrics comparable, we mapped the results of the FD and the EMD to the interval of the PC by subtracting their results from 1. Thus, 1 indicates a full match, 0 a total mismatch and  $-1$  full match except the sign. Please note that the interval  $[-1, 0)$  is only applicable to the PC. The source signal  $\mathbf{x}$  that was used to generate the results is a sine signal with an amplitude of 1 going over two full periods. Furthermore,  $n$  is set to 200.

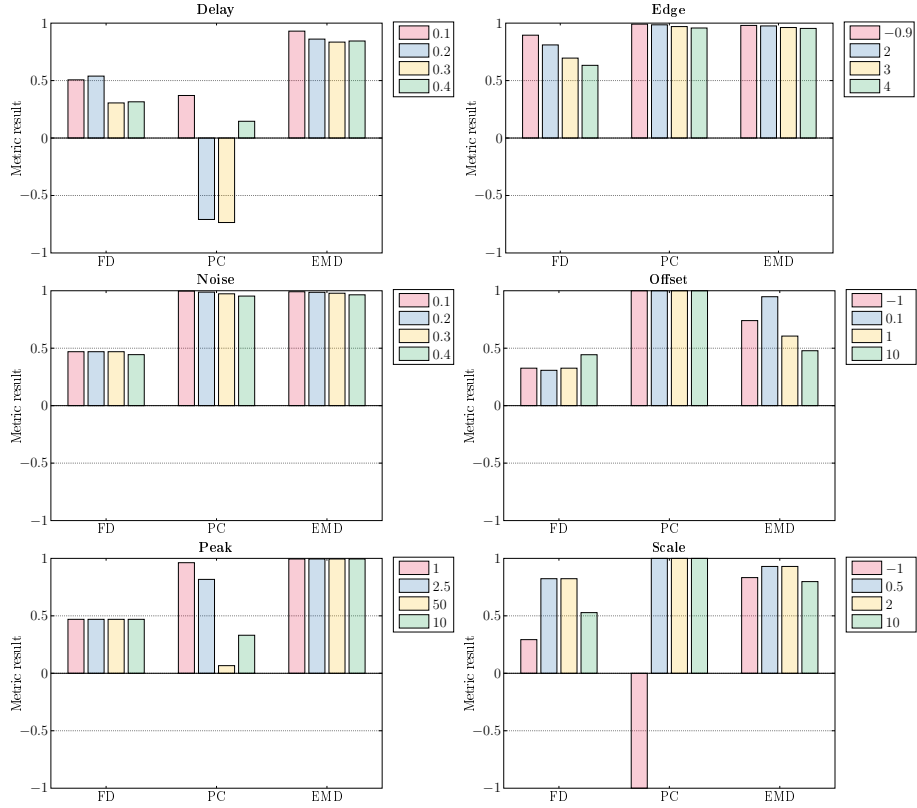


**Fig. 7.** Sine signal with noise deviation.

From the results it can be seen that the comparison algorithms judge the delay deviation very differently. FD and PC react very pessimistic and, thus, suggest that source and deviated signal are hardly related to each other. In contrast, EMD detects the deviation but does not drop very low, indicating that there is still a relation between the two signals. The edge deviation is treated similarly by all algorithms, with FD being somewhat more pessimistic than the others. Noise is detected by all algorithms; again, FD being very pessimistic. In addition, FD is not able to differentiate between the noise severities. The offset deviation is detected well by EMD and FD. However, FD shows a peculiar behavior in the sense that it returns bigger values for severity 1 and 10. PC is not sensitive regarding offset. The peak is very well analyzed by PC, while not being detected at all by EMD. The FD detects the peak, however, it can't differentiate between different severities. Scaling is being detected well by EMD and FD, whilst PC is not sensitive, except on the sign.

**Table 1.** Qualitative analysis of the assessment. The table shows whether an algorithm is able to detect (D) the deviation and whether it still correctly reports a similarity (S) between source and deviated signal.

	FD		PC		EMD	
	D	S	D	S	D	S
<b>Delay</b>	+	-	+	-	+	+
<b>Edge</b>	+	+	+	+	+	+
<b>Noise</b>	+	-	+	+	+	+
<b>Offset</b>	+	-	-	+	+	+
<b>Peak</b>	+	-	+	-	-	+
<b>Scale</b>	+	-	-	+	+	+



**Fig. 8.** Results of the assessment. Each diagram stands for a type of deviation. The horizontal axis shows the applied algorithms. The vertical axis shows the result of the comparison between the source signal and the deviated signal with the severity  $p$  of the deviation indicated by color. A value of 1 on the vertical axis stands for a full match, i. e. the metric states that the signals being compared are identical; 0 stand for a full mismatch, i. e. the metric states that the signals are completely dissimilar;  $-1$  stands for a full match except the sign (note that only PC can produce such a result). For an interpretation of the results, see Table 1.

In order to summarize the assessment, we compressed the results from Fig. 8 into Table 1. That table indicates, whether a metric is able to detect (D) a distortion and whether it can still see a similarity (S) between the original signal and its distorted version. We count a distortion as detected if the average of the absolute value of resulting metric  $M$  is significantly smaller than 1:

$$D = \begin{cases} + & \text{if } |\bar{M}| < 0.99 \\ - & \text{else} \end{cases} \quad (25)$$

For example, PC can't detect a scale distortion, as all four result in Fig. 8 are  $\pm 1$  leading to an average of the absolute of  $1 \geq 0.99$ .

As a criterion for similarity being counted as detected, we define the average of the absolutes to be greater than 0.66:

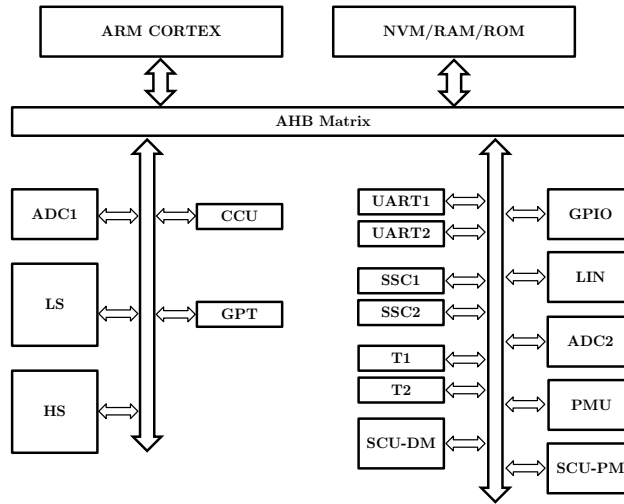
$$S = \begin{cases} + & \text{if } |\bar{M}| > 0.66 \\ - & \text{else} \end{cases} \quad (26)$$

It can be seen that the EMD can detect all kind of signal distortions, except peak. Referencing the explanation of EMD, where soil is transported from one pile to another, the peak is not detected because the amount of work that is needed to transport the peak is low. This is due to the peak being used in our assessment is very narrow, i. e. it does not contain much soil. However, the biggest advantage of EMD is that it is always able to correctly detect similarity between source and deviated signal.

## 5 Case Study

In order to test the reliability of the presented approach, we applied the EMD to a chip level verification of a real-world design. This case study is outlined in the following paragraphs.

### 5.1 Application



**Fig. 9.** Block diagram of the SoC used for our analyses.

The design we applied the EMD to is micro-controller-based and features a LIN interface and power switches (see Fig. 9). Its application is focused on

driving small electric motors—e. g. for window lifts or cooling fans—in cars. Being an automotive application, power consumption is of great concern for the design. In order to fulfill the requirements regarding power consumption, the design features a sleep mode as well as wake-up mechanisms. The verification of sleep and wake-up on chip level is crucial, since it affects three design domains—digital and analog hardware as well as software. The chip level verification was performed as a mixed-signal simulation in which the hand-shaking between the domains and the voltage curves of the power rails were checked for correctness. The checking was done in such a way that the results of the initial simulation were inspected manually and stored in the file system. The results of all subsequent simulations were then automatically compared against the results of the initial run using EMD.

In Fig. 10, we show the voltage curve of a rail in the power management unit gained during different simulations. The first curve is the one recorded during the initial simulation and serves as a reference for all subsequent simulations. The colored phases have been defined after the initial simulation, such that the automatic comparison of subsequent results with the ones gained during later simulations could be done phase-wise. Curves A to D represent simulation results after several modifications to the design.

## 5.2 Results

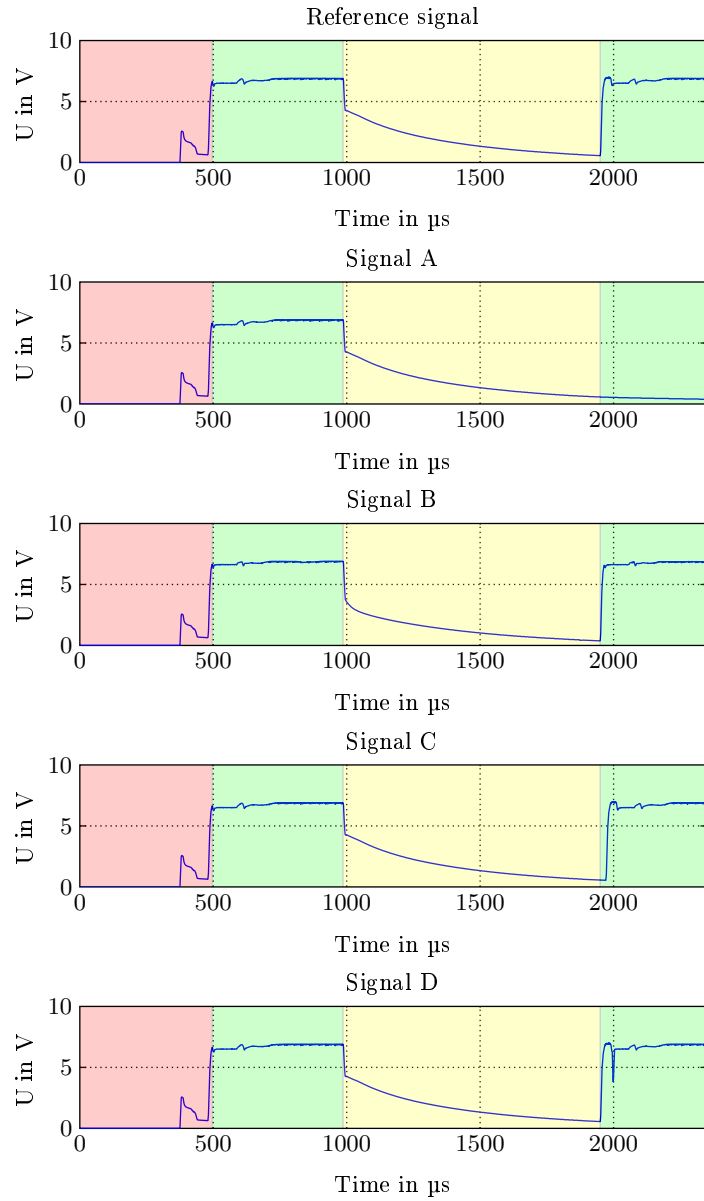
Table 2, 3 and 4 show the results of the comparison between signal curves A to D and the reference using EMD, PC and FD, respectively. In order to detect a low similarity, we empirically assume a threshold value of 0.97 for each analysis. "Empirically" means that the threshold value, which serves as a pass/fail criterion, is application dependent and is therefore to be defined by verification engineers on a per-circuit basis. Results which fall below this threshold indicate an insufficient degree of similarity (i. e. fail) and are marked red in the tables.

First, it can be clearly seen that due to a bug the design does not wake up in curve A, resulting in an EMD value of approximately 0.5. This unintended functional behavior is also detected by the other two metrics.

A less obvious signal deviation can be found in curve B. The obtained results reveal a change of certain capacitors in the design leading to a slightly different discharging behavior in phase 3 (EMD  $\approx$  0.96 and FD  $\approx$  0.93). For this case PC already shows the first deficiencies since the metric is not able to detect the deviation (PC  $\approx$  0.99). The aforementioned change of particular external capacitances was actually not a bug, but a specification-compliant adjustment. Therefore, the reference curve had to be replaced by curve B.

In the third case (signal C), wake-up is triggered at a later point in time from the test bench. In this application, this has no impact on the functional behavior of the design. Therefore, the second active phase, which is consequently being altered by a delay, shall not be flagged as too dissimilar to the reference. However, Table 2 to 4 show that the results for PC and FD fall below the predefined threshold while EMD is able to compensate this deviation. This observation goes





**Fig. 10.** Voltage curves of a power rail in an automotive design during different design states gained by chip level simulation. The red phase shows the power-up of the design, green stands for the active state and yellow for sleep mode.

in line with the previously stated claim that the threshold has to be adjusted per application, i. e. per circuit.

**Table 2.** Comparison of the curves from Fig. 10 using EMD.

Signals	Phase 1	Phase 2	Phase 3	Phase 4
Ref vs Ref	1.0000	1.0000	1.0000	1.0000
Ref vs A	1.0000	1.0000	0.9999	<b>0.5118</b>
Ref vs B	1.0000	0.9999	<b>0.9577</b>	0.9998
Ref vs C	1.0000	1.0000	1.0000	0.9741
Ref vs D	1.0000	1.0000	1.0000	0.9980

**Table 3.** Comparison of the curves from Fig. 10 using Pearson Correlation Coefficient.

Signals	Phase 1	Phase 2	Phase 3	Phase 4
Ref vs Ref	1.0000	1.0000	1.0000	1.0000
Ref vs A	1.0000	0.9920	1.0000	<b>-0.4070</b>
Ref vs B	1.0000	<b>0.9369</b>	0.9883	0.9998
Ref vs C	1.0000	1.0000	1.0000	<b>0.5380</b>
Ref vs D	1.0000	1.0000	1.0000	<b>0.8599</b>

Switching on an external high load during wake-up leads to the signal shape illustrated by curve D. It is characterized by a peak in the fourth phase at 2000  $\mu\text{s}$ , which can be traced back to a short-time voltage sag. Except for the peak, there is no significant difference between curve D and the reference signal. However, PC and FD are again not able to compensate this deviation and would flag a mismatch ( $PC \approx 0.86$  and  $FD \approx 0.87$ ) whereas EMD correctly detects the functionally unchanged wake-up behavior ( $EMD \approx 1.0$ ).

While the critical “sleep mode”-bug in the above-stated mixed-signal design was detected by all three metrics, EMD reveals its strengths in particular when deviations like delay or peak are distorting the signals. In these cases PC and FD are not able to identify correct functional behavior, which in turn confirms our analyses from Section 4. A concluding summary of our results can be found in Table 5. We furthermore observed the aforementioned deficiencies when analyzing several other metrics like Euclidean distance, cosine similarity and Hausdorff distance, inspiring us with confidence that our approach represents the most reliable one.

## 6 Conclusion and Outlook

In this contribution we introduced and described a novel technique for the automated comparison of analog behavior. The technique tackles the necessity of being able to automatically check the functional correctness of analog and mixed-signal ICs during design and verification. It moreover opens the possibility of reducing verification efforts by avoiding manual and error-prone investigations of waveforms since it can be easily leveraged for regression tests. Beyond that,

**Table 4.** Comparison of the curves from Fig. 10 using Fréchet Distance.

Signals	Phase 1	Phase 2	Phase 3	Phase 4
Ref vs Ref	1.0000	1.0000	1.0000	1.0000
Ref vs A	0.9998	0.9998	0.9993	<b>0.5009</b>
Ref vs B	0.9998	0.9989	<b>0.9250</b>	0.9965
Ref vs C	1.0000	1.0000	1.0000	<b>0.9413</b>
Ref vs D	1.0000	1.0000	1.0000	<b>0.8718</b>

**Table 5.** Overview of results: EMD is the only metric that correctly detects, i.e. compensates all four deviation scenarios.

Signal	Deviation in phase	Reason for deviation	Supposed to be detected?	Detected by
A	4	Bug in the design	yes	EMD, FD, PC
B	3	Modified external capacitance	yes	EMD, FD
C	4	Delayed wake-up	no	FD, PC
D	4	Voltage sag due to temporary high load	no	FD, PC

applying the presented concept consequently leads to significantly reduced costs and speeds time to market.

Furthermore, we assessed the approach by comparing it with related techniques and presented a case study, in which we applied the approach to a real-world project confirming its usefulness and high reliability compared to other metrics.

Aside from the outlined benefits in the field of mixed-signal verification, the developed approach has also high impact on several other topics. It can, for instance, be leveraged to optimize objective functions, which in turn is helpful when performing model calibration. In this case an automated regression could be set up to successively fit model parameters of interest. Another field of application is the regression testing of SPICE-style simulators, which can be used to identify diverging simulation behavior caused by differing simulator versions.

Our future work will focus on the extension of the presented methodology regarding usability and flexibility. The long-term goal is to provide a building box that covers the need of verification engineers to simulate and verify mixed-signal designs. This building box shall include methods and techniques for driving, monitoring and checking analog signals, as well as for coverage collection and reference modeling.

## References

1. Khan, N., Kashai, Y.: From spec to verification closure: a case study of applying uvm-ms for first pass success to a complex mixed-signal soc design. In: Design and

- Verification Conference. (March 2012)
2. Yang, X., Niu, X., Fan, J., Choi, C.: Mixed-signal system-on-a-chip (soc) verification based on systemverilog model. In: System Theory (SSST), 45th Southeastern Symposium on. (2013) 17–21
  3. Neumann, F., Sathyamurthy, M., Kotynia, L., Hennig, E., Sommer, R.: Uvm-based verification of smart-sensor systems. In: Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), International Conference on. (2012) 21–24
  4. Sharma, V., Lakshmanan, G., Tare, S., Dhamankar, S.: Predicting the correlation between analog behavioral models and spice circuits for robust soc verification. In: Behavioral Modeling and Simulation Workshop, Proceedings of the 2008 IEEE International. (2008) 130–135
  5. Khan, N., Kashai, Y., Fang, H.: Metric driven verification of mixed-signal designs. In: Design and Verification Conference. (March 2011)
  6. Rath, A.W., Esen, V., Ecker, W.: A transaction-oriented UVM-based library for verification of analog behavior. In: Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific. (Jan 2014) 806–811
  7. Ohlendorf, O., Steinhorst, S., Hartong, W., Hedrich, L.: Comparing Two Analog Waveforms - A Trivial Task? In: Zuverlässigkeit und Entwurf (ZuD 2008), Ingolstadt. (September 2008) 153–154
  8. Rubner, Y., Tomasi, C., Guibas, L.J.: The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision* **40**(2) (November 2000) 99–121
  9. Cohen, S.: Finding Color and Shape Patterns in Images. PhD thesis, Stanford University (May 1999)
  10. Yoon, S., Yoo, H.M., Yang, S.H., Park, D.S.: Computation of Discrete Fréchet Distance using CNN. In: Cellular Nanoscale Networks and Their Applications (CNNA), 2010 12th International Workshop on. (February 2010) 1–6
  11. Theodoridis, S., Koutrumbas, K.: *Pattern Recognition*. 4th edn. Academic Press (November 2008)
  12. Xiang, H.: Similarity based virtual screening using frequency-based weighting-schemes: Effect of the choice of similarity coefficient. Poster presentation at the Ninth International Conference on Chemical Structures (June 2011)
  13. Klee, V., Minty, G.J.: How good is the simplex algorithm? In: *Inequalities-III: Proceedings of the Third Symposium on Inequalities Held at the University of California, Los Angeles, September 1–9, 1969*, Academic Press (1972) 159–175
  14. Yang, Z.: Fast Template Matching Based on Normalized Cross Correlation with Centroid Bounding. In: *Measuring Technology and Mechatronics Automation (ICMTMA), 2010 International Conference on. Volume 2.* (March 2010) 224–227
  15. Eiter, T., Mannila, H.: Computing Discrete Fréchet Distance. Technical report, Technische Universität Wien (April 1994)