Verifying Jitter in an Analog and Mixed Signal Design using Dynamic Time Warping

Rajeev Narayanan, Alaeddine Daghar, Mohamed H. Zaki and Sofiène Tahar
Dept. of Electrical and Computer Engineering, Concordia University, Montreal, Quebec, Canada
Email: {r_naraya, daghar, mzaki, tahar}@ece.concordia.ca

***Abstract*—We present a variant of dynamic time warping (DTW) algorithm to verify jitter properties associated with an analog and mixed signal (AMS) design. First, the AMS design with stochastic jitter component is modeled using a system of difference equations for analog and digital parts and then evaluated in a MATLAB simulation environment. Second, MonteCarlo simulation is combined with DTW and hypothesis testing to determine the probability of acceptance/rejection of those simulation results. Our approach is illustrated on analyzing the jitter effect on the "lock-time" property of a phase locked loop (PLL) based frequency synthesizer.**

I. Introduction

One of the major challenges for the verification of an AMS design, such as the phase locked loop (PLL), is evaluating the uncertainties due to short-term frequency perturbation known as the *jitter* [2]. Jitter, a time-domain measure, is an unwanted contraction or expansion in the output oscillating signal from its ideal position and may result in wrong synchronization of the AMS design. The sources of jitter could be inherited from the circuit elements or from unwanted interaction between different analog/digital blocks. The amplitude associated with the jitter can be either bounded (*deterministic jitter*) or unbounded (*random jitter*) with respect to time [4]. Evaluating the jitter metrics strongly depends on the jitter model, the verification environment and the class of AMS circuit. Due to its random nature, the first step in jitter measurement is to characterize its behavior as a probabilistic function and the most prominent approach is the use of a gaussian distributed model [9].

Lots of progress has been made in estimating the effect of jitter for an AMS design in phase/voltage domains. For instance, for a PLL design, researchers achieve a good fit between the measured and the extracted values by advocating the use of gaussian distributed jitter models that are simulated at higher level of abstraction using the phase domain method [1]. Taking a step further, the author in [7] used the ideas of [1] to develop a methodology for commercial purposes by integrating Verilog-A based jitter model with Spectre [8]. As this involves behavioral simulation, it lays a strong foundation for a system level verification of AMS designs. The use of a voltage-domain method in a Verilog-A environment has been campaigned by the authors in [12], wherein, the jitter properties of the synthesizer are extracted from transistor level through simulation.

From a verification perspective, the authors in [13] have made use of the jitter models from [1] and have combined MonteCarlo and hypothesis testing to provide a statistical estimate of the jitter property specification. Unfortunately, simulation based verification approaches remain rigid to that particular AMS design, and taking a unified approach requires colossal changes to the methodology and hence, is impractically expensive. An unconventional verification method in the context of pattern matching was proposed by the authors in [10] using a variant of longest common subsequence (LCS). The idea is to find the longest closest subsequence between an ideal and non-ideal analog signal. But, this method is deemed inaccurate for handling instabilities due to jitter. This paper tries to address some of the above shortcoming by warping the jittery and the ideal signal to measure their similarity independent of their non-linear variations.

Dynamic Time Warping (DTW) [3] is a pattern matching algorithm that finds its applications in audio, video, and graphics designs to cope for different signal speeds. DTW is a method that finds an optimal path between two given time series sequences. As opposed to the traditional approach of comparing the output of the design to its specification value, we can extend DTW to measure the best possible match between the jittery output signal and the specification. The main advantage of this approach is that it works on circuit simulation traces and is independent of the circuit models. Therefore, it facilitates a unified approach for verifying jitter in an AMS design. In addition, due to the random nature of jitter, it is necessary to use statistical techniques to draw a conclusion on the optimal path. We use hypothesis testing [11], a technique that exploits probabilistic random sampling, to determine the acceptance region for the optimal path.

In this paper, we present to the best of our knowledge, the first dynamic time warping (DTW) algorithm to verify random jitter properties associated with an AMS design. The idea is to model the AMS designs with stochastic jitter component using systems of difference equations and then evaluate them in a MATLAB simulation environment. Thereafter, we combine MonteCarlo simulation with DTW and hypothesis testing to determine the acceptance/rejection of those simulation traces. Our approach is illustrated by the analysis of the jitter effect on the "lock-time" property of a PLL based frequency synthesizer.

II. Proposed Methodology

Figure 1 shows the overall verification methodology based on DTW. Thereafter, given an AMS design, the idea is to

978-3-9810801-8-6/DATE12/©2012 EDAA

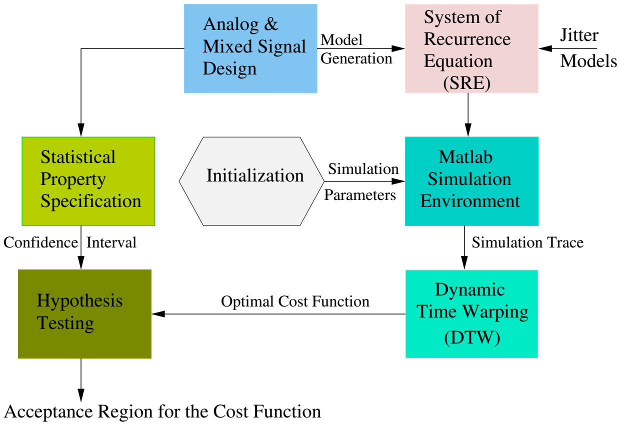

Fig. 1: Overview of the Verification Methodology

generate a system of difference (or recurrence) equations (SREs) that describe the behavior of the design. A recurrence equation is the discrete version of an analog differential equation [13]. For digital blocks in the AMS design, the SRE can be expressed as an *if-else* logical formula. For continuous time components, the formulation is based on the semi-automatic generation of recurrence equation models as described in [5].

We make use of the jitter model from [7], which is added to the difference equation to form the overall behavior of the AMS design with jitter. The next step is to evaluate the AMS model with jitter in a MATLAB simulation environment for the specified environment constraints such as the initial condition of the circuit current and voltages and simulation parameters such as step-size, total simulation cycle and so on. We generate two sets of sequences, one for the ideal signal *X* and the other for the jitter signal *Y*. These two sequences (X and Y) are evaluated using the DTW algorithm, as shown in Figure 1, in order to generate the optimal cost function. We then combine MonteCarlo simulation with the bounded hypothesis testing to evaluate those cost functions for a given confidence level in order to derive the acceptance region for the optimal cost as given in Figure 1. The acceptance region concludes that if the AMS design operates outside this range, it is bound to fail.

Hypothesis testing [11] is the use of statistics to make decisions about acceptance or rejection of some statements based on the data from a random sample, meaning, to determine the probability that a given hypothesis is true for a certain confidence interval. Hypothesis testing in general has two parts: *null hypothesis*, denoted by H_0, which is what we want to test (e.g., $jitter\_period \leq 3.2\ ns$), and *alternative hypothesis*, denoted by H_1, which is what we want to test against the null hypothesis (e.g., $jitter\_period > 3.2\ ns$). If we reject H_0, then the decision to accept H_1 is made. The conclusion is drawn with certain probability of error for a specific confidence level.

One of the verification challenges of AMS design simulation is the memory usage because output signals usually have a high number of samples. As the origin of the DTW is in the arena of speech/video processing, many researchers have applied the DTW algorithm on the signal spectrogram which allows doing a short-time Fourier transform [6]. A spectrogram plots the variation of frequency of any signal as function of time with the frequency in Y-axis and time in X-axis. It is used when there is a need for understanding the relativeness of different frequencies, i.e., to know which frequency came before the other [6]. This spectrogram technique has been adopted in this paper in order to avoid memory overflow for large simulation traces.

Dynamic time warping (DTW) is an algorithm developed by the speech recognition community to handle the matching of non-linearly expanded or contracted signals [6]. The algorithm finds the optimal path through a matrix of points representing possible time alignments between the signals. The optimal alignment can be efficiently calculated via dynamic programming.

Property 1. *Given any two sequences $x = \{x_1, x_2, \dots x_m\}$ and $y = \{y_1, y_2, \dots y_n\}$, then the distance of the best possible partial path is defined as*

$$D(i,j) = d(x_i, y_j) + min \begin{cases} D(i, j-1) \\ D(i-1, j) \\ D(i-1, j-1) \end{cases} \tag{1}$$

where $1 \leq i \leq m$; $1 \leq j \leq n$. $d(x,y)$ is a distance between the signals, e.g., $d(x_i,\ y_j) \ = \ |\ x_i \ - \ y_j\ |$.

To better understand the algorithm, let us apply it to the following traces shown in the Figure 2.

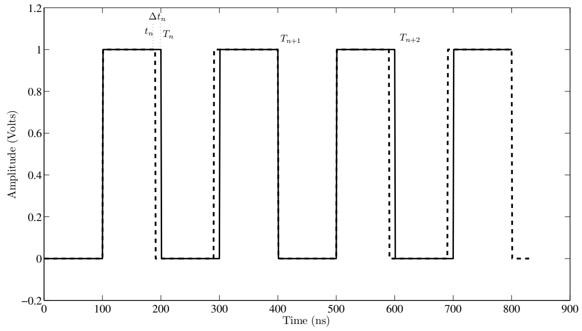

Fig. 2: Dynamic Time Warping Example

The dotted waveform represents the jitter signal (Y) and the bold trace is the ideal signal (X). We have the following values for X and Y,

$$\begin{aligned} X &= [\ 00000111110000011111\ ] \\ Y &= [\ 00000111100000111111\ ] \end{aligned}$$

The first step is to determine a *"m by n"* matrix that represents the best possible distance between X and Y. The importance of this matrix is to understand the contraction and expansion of the two simulation traces. By contraction, we mean the jitter signal period is shifted to the left of the ideal signal and expansion represents the right shift of the jittery signal with respect to the ideal signal.

The implementation of the matrix is described in Algorithm 1. The algorithm starts by reading the two sequences X and Y. The entries in the matrix table are generated as follows: First, we define a matrix *D* that fills the first row and column with ∞ and $D(0,0)$ with 0 (lines 3-9). Then, it takes the minimum between $D(i-1,j-1), D(i,j-1), D(i-1,j)$, and adds to it the cost which is the distance between x_i and y_j (lines 10-15). Here, *i* and *j* represent the index of the sequences X and Y, respectively. The algorithm continues until we reach the end of the sequence, and it returns the matrix table *D* for determining the optimal path and allows us to have the minimum cost alignment *D(n,m)*. Once the matrix is generated, it is necessary to trace back the path to detect those values that represent the contraction and expansion.

Algorithm 1 : DTW Algorithm

```
Require: $x, y$
 1: $n \leftarrow length(x)$
 2: $m \leftarrow length(y)$
 3: for $i \leftarrow 1 \ to \ m$ do
 4:     $D(0,i) \leftarrow inf$
 5: end for
 6: for $j \leftarrow 1 \ to \ n$ do
 7:     $D(j,0) \leftarrow inf$
 8: end for
 9: $D(0,0) \leftarrow 0$
10: for $i \leftarrow 1 \ to \ m$ do
11:     for $j \leftarrow 1 \ to \ n$ do
12:         $cost \leftarrow d(x(i) - y(j))$
13:         $D(i,j) \leftarrow cost + min(D(i-1,j-1), D(i,j-1), D(i-1,j))$
14:     end for
15: end for
16: return $D$
```

III. PLL Based Frequency Synthesizer

Figure 3 shows a PLL based frequency synthesizer that is commonly used in communication systems for clock generation and recovery. It is composed of two comparators, a phase/frequency detector, a charge pump, an analog filter, a voltage controlled oscillator (VCO) and a divider.

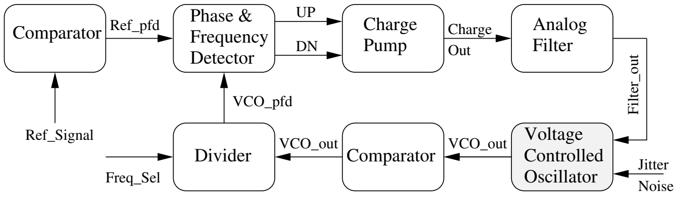

Fig. 3: PLL Based Frequency Synthesizer

The reference signal (*Ref_Signal*) at the input is a simple sinusoidal wave with frequency ω_0. The VCO output (*VCO_out*) is a cosine signal with frequency *N+1* times of the reference frequency, where *N* is determined by the frequency select signal (*Freq_Sel*). If the *Freq_Sel* is '0', the frequency of the reference input and VCO output will be the same or else the frequency will be divided accordingly based on the divider. This paper addresses the issue of random jitter associated with the VCO. We incorporate in a MATLAB simulation environment the SRE based models [7] for the VCO with jitter and the other blocks of the PLL design. We have applied the DTW in two ways: First, to study the effect of jitter on the "lock-time" property, and second, to estimate the optimal cost alignment by combining MonteCarlo simulation for "1000" trials with the bounded hypothesis testing.

A. Lock-Time Property Observation

The critical property of a frequency synthesizer is the "lock-time", meaning, if the *Freq_Sel* is activated, the PLL will lock at the desired frequency within a certain time as identified in the specification. However, the jitter in the VCO circuit may cause a drift in its output that may lead to changes to the lock-time. The lock-time is an isolated property for all PLL based frequency synthesizers, i.e., once the PLL gets locked, the VCO will start oscillating until there is a change to the *Freq_Sel* signal. The conventional method [7] of verifying the "lock time" property is to check if the output of the low-pass filter has reached a new DC value within the lock time. Unlike such an approach that is dependent on the design under test, the proposed DTW method allows designers to work on the VCO simulation trace directly by finding the lock time and the minimum cost function associated with it.

The first step in finding the optimal cost is to simulate the design and generate two sets of sequences. This is followed by generating the spectrogram of those two VCO signals because the total simulation trace of the VCO output has more than one million samples. This spectrogram of the VCO output with jitter will be compared with the spectrogram of the ideal output signal which has a constant frequency (horizontal line). We then apply the DTW algorithm to determine the minimum cost alignment between the two outputs. If the observed cost is very big with respect to the cost of the ideal output, it can be concluded that there is a large deviation in the signal frequency compared with the ideal one.

The lock time can now be determined by looking at the time when the minimum path calculated by the DTW crosses the diagonal as shown in Figure 4. The novelty of such an approach lies in the fact that the DTW will not only classify outputs based on the frequency quality but can also determine the value of the lock time. In this case, the lock time was determined to be *1.0944 ms*.

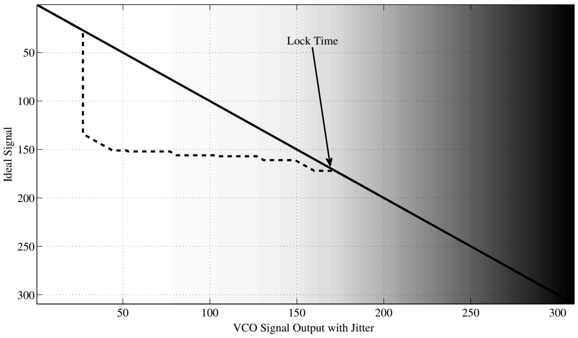

Fig. 4: VCO Output Warped using DTW

B. Decision Based on Hypothesis Testing

Since the jitter is considered to be a random noise that has gaussian distribution, we have performed MonteCarlo simulation for "1000" trials to evaluate the cost and then

Due to lack of space, the DTW path finding algorithm is not provided.

TABLE I: DTW and Hypothesis Tesing Results for the PLL

J	Jitter Effect			Minimum Alignment Cost			Lock Time
	Mean	**Variance**	**Acceptance Region**	**Mean**	**Variance**	**Acceptance Region**	**(ms)**
1e-13	-2.73e-7	8.12e-5	[-1.338e-4 - 1.333e-4]	153.190	0.00544	[153.181 - 153.199]	1.0944
1e-12	-1.0032e-5	8.0281e-4	[0.00131 - -0.00133]	153.197	0.00748	[153.185 - 153.209]	1.0944
1e-10	-6.6129e-4	0.0810	[-0.13396 - 0.13264]	153.122	0.14995	[152.875 - 153.369]	1.0944
5e-10	0.0015	0.4081	[-0.6697 - 0.6728]	152.930	0.65085	[151.860 - 154.001]	1.0944
9e-10	0.0068	0.7304	[-1.1946 - 1.2082]	153.443	1.16306	[151.529 - 155.355]	1.0944
1e-9	-0.0015	0.8215	[-1.3528 - 1.3497]	153.691	1.33621	[151.492 - 155.888]	1.2096
2e-9	-0.0035	1.6325	[-2.6887 - 2.6817]	158.179	2.36523	[154.288 - 162.069]	1.7472
3e-9	-0.01598	2.4520	[-4.0493 - 4.0174]	166.015	3.26501	[160.644 - 171.385]	—
5e-9	-0.01079	4.1609	[-6.8549 - 6.8333]	185.937	3.69045	[179.867 - 192.007]	—

used hypothesis testing to reason about the results. For this kind of verification, one would be interested to know *"for the given confidence level α, and M trials, what is the region of acceptance and rejection of the circuit?"* As a part of the specification, designers have to specify the confidence interval and the cost. This cost will be checked if it is in the acceptance region of tolerance of the jittery signal with respect to the ideal signal. Table I summarizes the results for different jitter "J" deviations. The table is derived by taking into account the jitter factor that represents the effect of jitter on the phase of the VCO output signal. The influence of jitter on the cost function is shown in Figure 5. The plot shows the minimum cost following a normal distribution like the jitter noise in the VCO but with different means and deviations. From Table I, it can be noted that the minimum cost alignment increases non-linearly with the mean and variance.

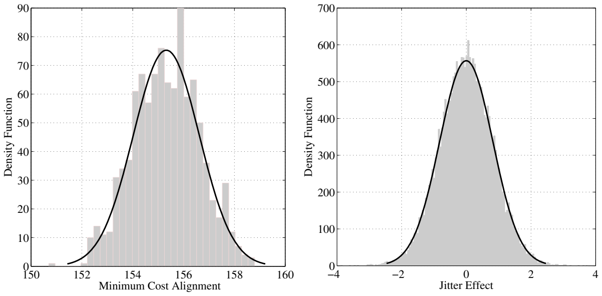

Fig. 5: Influence of the Jitter on the Cost

We also see that when "J" is large, the PLL fails to lock as represented by the "dashed entry" (shaded region) in Table I. The spectrogram in such a case will show the minimum path failing to cross the diagonal line. We use hypothesis testing to find the acceptance region for each "J" as shown in Table I.

IV. Conclusion

This paper presents a methodology based on pattern matching for handling jitter in AMS designs. Our approach is based on modeling the design using system of difference equations, and then using dynamic time warping (DTW) to find the minimum cost alignment. We have also combined MonteCarlo simulation and hypothesis testing to decide on the acceptance/rejection of the circuit. Our approach is illustrated on analyzing the jitter effect on the "lock-time" of a PLL based frequency synthesizer.

Unfortunately, the current verification methodology remains tied to the design under test (DUT), and taking a unified approach requires major changes to the testbench structure and hence, is impractically expensive. However, DTW based techniques work on the simulation trace in polynomial time and hence will be well suited for verifying "black-box" AMS designs. Also, the use of spectrogram has provided an alternative solution to the memory usage problem faced by AMS design verification. The statistical environment provides designers with additional information about the acceptance region, thereby allowing them to make better decisions.

We would like to derive complex and accurate jitter models through circuit level simulation and integrate them with the PLL behavioral model at a higher level of abstraction. Especially, our next step is to incorporate jitter models for the input reference signal, the charge-pump and the phase detector. Also, an extension to accommodate process variation on various passive and active elements is also planned.

References

[1] A. Demir, E. Liu, A. Vincentelli, and I. Vassiliou. Behavioral Simulation Techniques for Phase/Delay Locked Systems. Proc. IEEE Custom Integrated Circuits Conference, pp. 453-456, 1994.

[2] A. Hajimiri, S. Limotyrakis and T. H. Lee. Jitter and Phase Noise in Ring Oscillators. IEEE Journal of Solid-State Circuits, 34(6): 790-804, 1999.

[3] L. R. Rabiner and B. Juang. Fundamentals of Speech Recognition. Prentice-Hall, Inc., 1993.

[4] F. Herzel and B. Razavi. A Study of Oscillator Jitter Due to Supply and Substrate Noise. IEEE Transactions on Circuits and Systems-II, 46(1): 56-62, 1999.

[5] G. Al Sammane, M. Zaki, and S. Tahar. A Symbolic Methodology for the Verification of Analog and Mixed Signal Designs. Proc. IEEE/ACM Design Automation and Test in Europe, pp. 249-254, 2007.

[6] H. Sakoe, and S. Chiba. Dynamic Programming Algorithm Optimization for Spoken Word Recognition, IEEE Transactions on Acoustics, Speech and Signal Processing, 26(1): 43- 49, 1978.

[7] K. Kundert. Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers. https://www.desingers-guide.com, 2003.

[8] K. S. Kundert. The Designers Guide to SPICE and Spectre. Kluwer Academic Publishers, 1995.

[9] M.P. Li. Jitter, Noise, and Signal Integrity at High-Speed. Prentice Hall, 2007.

[10] R. Narayanan, M. Zaki, and S. Tahar: Ensuring Correctness of Analog Circuits in the Presence of Noise and Process Variation Using Pattern Matching; Proc. IEEE/ACM Design Automation and Test in Europe, pp. 1188-1191, 2011.

[11] W. L. Martinez and A. R. Martinez. Computational Statistics Handbook with MATLAB. Chapman & Hall/CRC, 2002.

[12] X. Mao, H. Yang, and H. Wang. Behavioral Modeling and Simulation of Jitter and Phase Noise in Fractional-N PLL Frequency Synthesizer. Proc. IEEE Behavioral Modeling and Simulation Conference, pp.25-30, 2004.

[13] Z. Wang, M. H. Zaki, and S. Tahar. Statistical Runtime Verification of Analog and Mixed Signal Designs. Proc. IEEE International Conference on Signals, Circuits and Systems, pp. 1-6, 2009.