DESIGN AND IMPLEMENTATION OF A LOW COMPLEXITY NCO BASED CFO COMPENSATION UNIT

Jochen Rust, Till Wiegand and Steffen Paul

Institute of Electrodynamics and Microelectronics (ITEM)
University of Bremen, Bremen, Germany
+49(0)421/218-62538 {rust, wiegand, steffen.paul}@me.uni-bremen.de

ABSTRACT

The use of Orthogonal Frequency Division Multiplex (OFDM) modulation has become increasingly important for actual and future mobile communication systems [1]. Within this scope, Carrier Frequency Offset (CFO) compensation is indispensable [2]. Its underlying algorithm requires the calculation of trigonometric functions, which is difficult to achieve by hardware implementation in general.

In this paper we propose a CFO compensation approach based on low complexity and hardware optimized linear function approximation. The Multiplier-less nonuniform Piecewise function Approximation (MPA) methodology significantly reduces the calculation effort of crucial algebraic terms. Matlab simulation is performed in order to prove the results in terms of feasibility. The compensation algorithm is also implemented in VHDL and synthesized as hard-wired Integrated Circuit (IC). A chip area of 0.034mm^2, a frequency of 181MHz and a total power consumption of $18.62\mu\text{W/MHz}$ are obtained.

Index Terms— Carrier Frequency Offset Compensation, Nonuniform Linear Function Approximation

1. INTRODUCTION

The demand for efficient hardware design in the scope of mobile communication has increased continuously in recent years. Nowadays, high data throughput as well as bandwidth efficiency are the prime targets and Orthogonal Frequency Division Multiplex (OFDM) has proven to be the most promising modulation scheme for current and future communication standards such as LTE, WiMAX or WLAN (IEEE 802.11n). Due to the orthogonal subcarrier arrangement, OFDM is in particular sensitive to Carrier Frequency Offsets (CFO), which appear due to a mismatch between the Local Oscillators (LO) at the transmitter- and receiver-front-end. Thereby, the CFO can reach several subcarrier spacings, which causes an Inter Carrier Interference (ICI) in frequency domain and needs to be compensated to ensure a reliable data demodulation. Considering an OFDM transmission, the system equation is defined by

$$y_{l,n} = h_{l,n} * s_{l,n} + w_{l,n}, \quad (1)$$

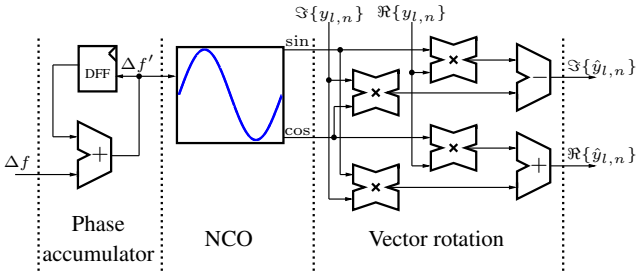

Fig. 1. Overview of a common CFO compensation hardware architecture.

where y denotes the receive-signal, l the OFDM-signal index and n the sample-time index; s is the transmitted symbol, h denotes the discrete channel and w the system noise. Under the effect of an CFO Δf_{CFO} the system equation is described by

$$y_{l,n} = \mathrm{e}^{j2\pi \frac{\Delta f_{\text{CFO}} \cdot (n + l(N_{\text{FFT}} + N_{\text{CP}}))}{f_s}} (h_{l,n} * s_{l,n} + w_{l,n}), \quad (2)$$

where N_{FFT} denotes the FFT size, N_{CP} the cyclic prefix (CP) length and f_s is the sampling frequency. Thus, a CFO causes a continuous symbol rotation in time domain and thereby, after estimating the CFO [3, 2], it can be compensated in time domain, either by adjusting a Voltage Controlled Oscillator (VCO) or derotating the symbol using a Numerically Controlled Oscillator (NCO). So for the latter, the CFO compensation can be expressed as

$$\hat{y}_{l,n} = y_{l,n} \mathrm{e}^{-j2\pi \frac{\Delta f_{\text{CFO}} \cdot (n + l(N_{\text{FFT}} + N_{CP}))}{f_s}}, \quad (3)$$

with $\hat{y}_{ln}$ as the synchronized receive-signal. Its default hardware architecture is given in Fig. 1.

In this paper we propose NCO-based CFO compensation using Multiplier-less nonuniform Piecewise function Approximation (MPA), which bases on linear approximation of a given function [4]. The use of appropriate sub-functions enables fast and efficient processing. A detailed description is given in the next Section. Section 4 and 5 present the architecture and implementation issues of our CFO compensation unit. In Section 6 IC synthesis results are depicted, before the most important aspects are summarized in the last Section.

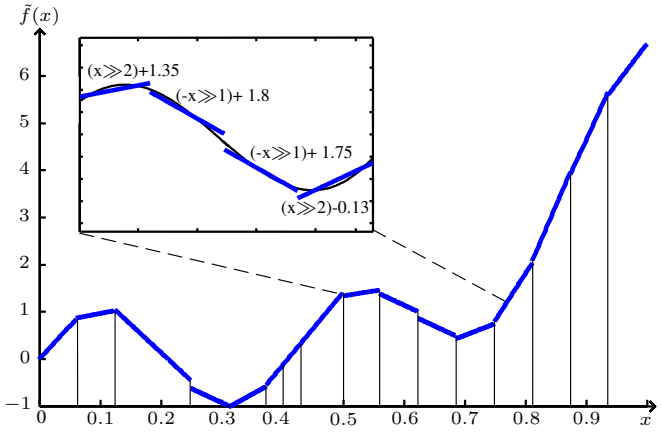

Fig. 2. Basic principle of multiplier-less nonuniform piecewise approximation.

2. RELATED WORK

Several approaches have been considered in order to increase the efficiency of CFO compensation. Its main issue is an insufficient calculation performance for the phase-to-sine mapper implementation. Thus, a huge range of optimizations have been considered, like mapping the corresponding sine function results to a ROM or LookUp-Table (LUT) [5]. Although this technique rises the runtime performance, it also noticeably increases the hardware complexity. Other approaches focus on iterative function approximation, e.g. CORDIC-based calculation [6]. Its underlying algorithm focuses on incremental result refinement by superposition of microrotations, which allows function estimation with low hardware complexity. On the other hand it only achieves high accuracy for a high number of iteration steps, which badly affects latency, critical path delay or chip area. Moreover, several approaches for sine function slope emulation use Piecewise function Approximation (PA). Recently, De Caro et al. presented a nonuniform PA method that comprises a fixed segmentation scheme as well as gradient quantization, which reduce the calculation effort to only three additions [7].

In order to increase this approach, we propose an even higher gradient quantization, reducing the mathematical effort of sine function calculation to only one addition. The underlying principle is described in detail in the next Section.

3. MULTIPLIER-LESS NONUNIFORM PIECEWISE APPROXIMATION

In this paper the MPA approach is used for NCO realization. Its basic principle focuses on linear function approximation, which enables efficient calculation of crucial algebraic terms at the cost of accuracy [4]. Generally, linear equations are defined as

$$f(x) = \alpha_0 x + \beta_0 \ , \tag{4}$$

where x denotes the input data, $f(x)$ the corresponding function, α_0 the gradient and β_0 the function offset.

For multiplier-less linear function approximation, the calculation complexity is reduced to a small set of partial products [4], which can be expressed in mathematical terms as gradient quantization. In order to keep the approximation accuracy at a convenient level, the common linear PA approach, which considers static equidistant segments, cannot be used in this scope. Thus, nonuniform segments are estimated by a variable number of bisection steps of the original function or the resultant sub-functions. This enables an accuracy-driven function approximation, as a relative or maximum approximation error can be specified in advance. Thus, MPA-based approximation can be expressed

$$\tilde{f}(x) = \left[\sum_{j=0}^{n-1} \pm \begin{pmatrix} 2^{\lambda_{0,j}} \\ 2^{\lambda_{1,j}} \\ \dots \\ 2^{\lambda_{m-1,j}} \end{pmatrix} x + \boldsymbol{\beta} \right] \cdot \boldsymbol{\kappa}(x) \ ; \ \lambda_{i,j} \in \mathbb{Z} \ , \tag{5}$$

with $\lambda_{i,j}$ as the actual partial product and m, n determining the total number of segments and partial products, respectively. The $\boldsymbol{\kappa}$ vector is responsible for the selection of the actual valid linear equation

$$\boldsymbol{\kappa}(x) = \begin{cases} (1,0,...,0)^T; & \mathrm{A} \leq x < \mathrm{seg}(1) \\ (0,1,...,0)^T; & \mathrm{seg}(1) \leq x < \mathrm{seg}(2) \\ \dots & \\ (0,0,...,1)^T; & \mathrm{seg}(m-1) \leq x < \mathrm{B} \end{cases} , \tag{6}$$

with A and B denoting the beginning and ending of the original function and $\mathrm{seg}(i)$ as the current approximated segment defined as

$$\mathrm{seg}(i) = \mathrm{seg}(i-1) + \frac{\mathrm{B}-\mathrm{A}}{2^{h_i}} \ ; \ h_i \in \mathbb{N} \ , \tag{7}$$

where h_i is the size coefficient of the ith segment and $\mathrm{seg}(0) = \mathrm{A}$. In order to enable high performance, an input range of $\mathrm{B} - \mathrm{A} = 2^{h_{max}}, h_{max} \in \mathbb{N}^+$ must be used. This constraint allows the MPA approach to perform segmentation by regarding the Most Significant Bits (MSB) of the input operand. Note, that h_i may differ for each segment. This effect may cause a varying number of MSB, which must therefore be considered. A graphical example of an MPA-based approximation is depicted in Fig. 2.

As briefly said before, besides the advantage of very low hardware complexity, the MPA approach also enables accuracy-driven approximation. By comparing the approximation accuracy of the actual segment to a specified value, further bisection can be performed if necessary. Thus, an automatic sequential approximation processing can be established, enabling accuracy-driven approximation as well as a simple comparison of different MPA configurations.

Table 1. Overview of four NCO configurations with different average accuracies and SFDR. The number of nonuniform segments for PA rises related to higher accuracy or SFDR.

Configuration	NCO_{50}	NCO_{55}	NCO_{60}	NCO_{65}
Accuracy $[10^{-3}]$	7.32	5.73	3.42	1.47
SFDR [dBc]	52.8	58.6	61.6	66.1
No. Segments	13	19	26	63

4. ARCHITECTURE

In general, the MPA-based compensation unit can be divided into two subunits. On the one hand, an NCO unit is required realizing the continuous phase-to-sine mapping. On the other hand, derotation must be performed, which can be easily achieved by vector rotation. An overview of the architecture is given in Fig. 1.

4.1. Numerically controlled oscillator

The NCO consists of a register accumulating the 16 bit input values and a phase-to-sine mapper. In order to enable hardware convenient signal processing, the full input phase operand range is scaled to 2^n values. In this paper the phase-to-sine mapper operates with the Q.11 fixed point number system ($n = 12$). Due to sine quarter wave symmetry, further reduction is achieved. The remaining range can be covered by function swapping, realized by adapted operand negations. For CFO compensation, the cosine value is required as well. As its relation to the sine function is given by $\cos(x) = \sin(x + \frac{1}{2}\pi)$, it can be derived in our case by adapting the two MSB.

In order to find an appropriate MPA-based design of the (co-)sine function, several parameters must be specified. On the one hand, the average accuracy is estimated by regarding the Spurious Free Dynamic Range (SFDR) [8]. Thus, several simulations are performed in this paper in order to achieve an appropriate candidate (see Sec. 4.3). An overview of the results is given in Fig. 3.

4.2. Vector Rotation

In general, the vector rotation step is used for phase correction and can be achieved easily by multiplying the rotation matrix $\mathbf{R}(\Delta f)$ with the offset $\Delta f = \Delta f_{\mathrm{CFO}}$:

$$\begin{pmatrix} \Re\{\hat{y}_{l,n}\} \\ \Im\{\hat{y}_{l,n}\} \end{pmatrix} = \begin{pmatrix} \Re\{y_{l,n}\}\cos\Delta f_{\mathrm{CFO}} + \Im\{y_{l,n}\}\sin\Delta f_{\mathrm{CFO}} \\ \Im\{y_{l,n}\}\cos\Delta f_{\mathrm{CFO}} - \Re\{y_{l,n}\}\sin\Delta f_{\mathrm{CFO}} \end{pmatrix} \tag{8}$$

As real and imaginary values can be treated separately, this is operated by four single multiplications and two additions. In contrast to the sine function calculation, multipliers are easy to implement in hardware, mainly due to their general advantage of parallel computation of partial results [9]. There are several approaches for multiplication that focus on different goals, such as timing or complexity. For our work, we select a modified radix-4 tree-multiplier structure [9], due to its approximation and segmentation ability as described below. In general, partial products of the first operand are calculated by adder and shift blocks (RRM I). Next, the second operand is split up, e.g. in bit pairs controlling the selection of partial products. Finally, the result is accumulated by a tree adder structure (RRM II, RRM III). In order to reduce the common tree-multiplier, LSB sequences of the select operand are truncated. This decreases latency and hardware complexity at the cost of little inaccuracy effects. Thus, in this paper we use a Reduced Radix-4 Multiplier (RRM) with a 16 bit operand input and a 12 bit selector saving three additions at runtime. This causes an accuracy error of $\Delta E = 2^{-12} \approx 2.44 \cdot 10^{-4}$ which is in all cases smaller than the accuracy error of the MPA-based NCO (see Tab. 1).

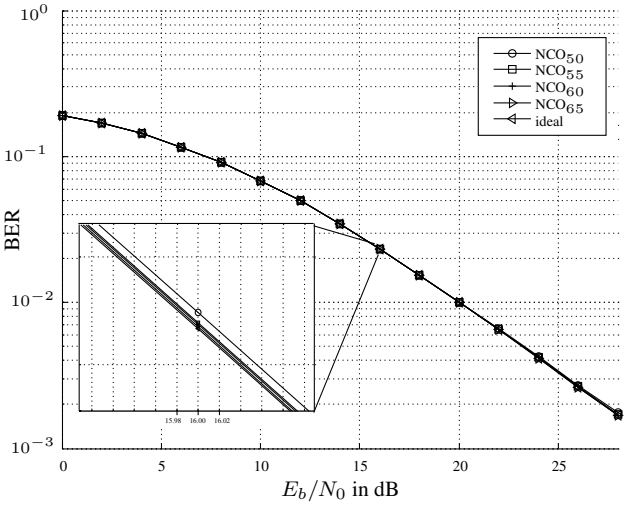

Fig. 3. BER over SNR of MPA-based CFO compensation for 64 QAM SISO.

4.3. Simulation

In order to find an appropriate hardware implementation candidate for the introduced MPA configurations, the CFO compensation algorithm is simulated in Matlab considering the modifications both for phase-to-sine mapper and the RRM. Thus, several MPA-based sine approximations are investigated with 12 bit resolution, one adder for gradient quantization and an SFDR not less than 50 dBc, 55 dBc, 60 dBc and 65 dBc denoted as NCO_{50}, NCO_{55}, NCO_{60} and NCO_{65}, respectively. Next, these configurations are applied to a Single Input Single Output (SISO) QAM64 modulated OFDM transmission model, simulated by use of a Rayleigh multipath channel. The CFO is generated randomly and known at the receiver. The received signal is demodulated by Maximum Likelihood (ML) detection using perfect Channel State Information (CSI). For evaluation, the resulting Bit Error Rates (BER) over E_b/N_0 are compared to ideal CFO compensation (3). As depicted in Fig. 3, the only small accuracy impair-

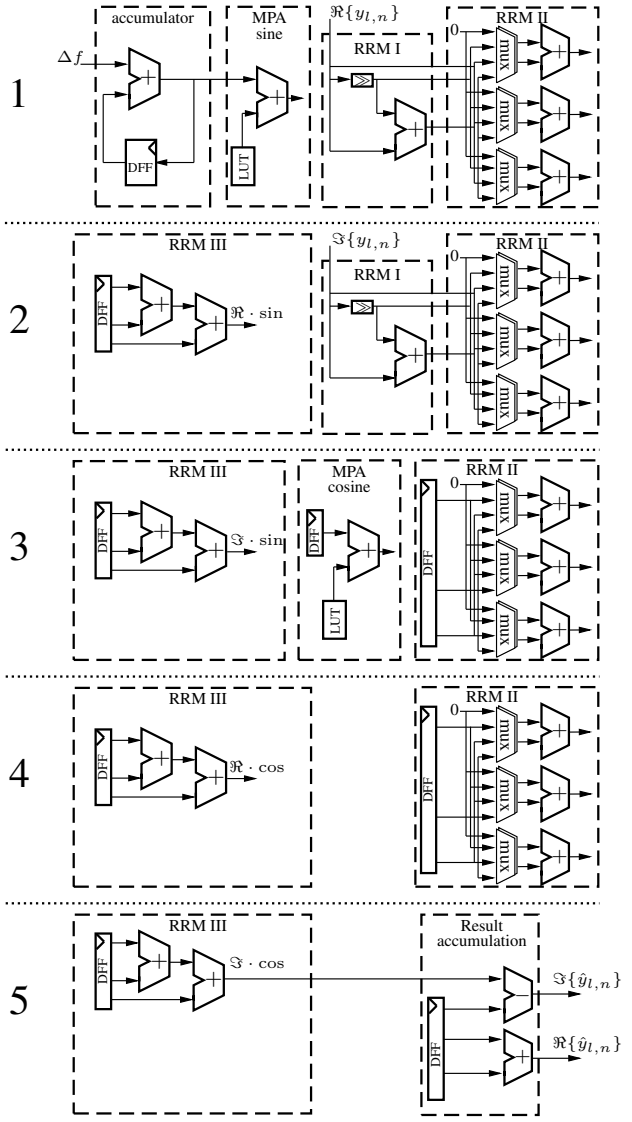

Fig. 4. Overview of multiplier-less NCO-based CFO architecture with six time-shared adders and a latency of five cycles. First, one adder is used for phase accumulation. Again one adder is required by RRM I in the first two cycles. Half of the adders are used in cycle 1-4 for RRM II calculation. Two adders are used for RRM III calculation in cycle 2-5. One adder is required for sine and cosine mapping in cycle 1 and 3 respectively. Finally, the result accumulation is performed in the last two cycles.

ment is recognizable for all NCO configurations. Thus, the 50 dBc configuration is selected for further investigations.

5. IMPLEMENTATION

For further complexity reduction, the CFO algorithm is implemented with time-sharing means. Thus, it is split up into several stages in which the whole calculation takes place. As the amount of additions should be nearly equal in each stage, a total number of six adders is chosen. In addition, a Finite State Machine unit (FSM), which handles the adder input output wiring is necessary. As described before, the 50 dBc SFDR configuration is selected, enabling CFO calculation in five cycles. Further hardware complexity reduction is achieved by sequential sine and cosine function calculation. A detailed functional overview of each stage is given in the following and in Fig. 4

1. In the first cycle, the accumulation and the MPA-based sine calculation is performed, which requires one adder each. The other four adders are used for RRM I and RRM II stage calculation (see Sec. 4.2). For operand input, the unrotated real value is taken. Note, that the NCO_{50} output is connected to the RRM selector input, due to its equal size. Additionally, this causes a smaller critical path than a connection to the operand input.

2. For the second cycle, the first two adders are used for RRM III stage processing. Again the other adders are used for RRM I and RRM II stage calculation, using the imaginary value at operand input.

3. The third cycle starts with the RRM III stage processing. It performs also the cosine calculation that requires one adder. As the RRM II input operands for both real and imaginary values have been already determined in advance, the RRM II processing for the multiplication of the real value and the cosine can also take place in this stage.

4. The fourth cycle has nearly the same wire configuration as the second, except for the cosine value that is used as RRM selector. Also the RRM I stage can be neglected as explained before. One adder remains unused.

5. In the fifth and last cycle, the CFO calculation is finished. For this, the last RRM III calculation is performed. Also the two additions of the vector rotation must be done here. Two adder remain unused.

In order to increase the power efficiency, the low power techniques clock gating and operand isolation are additionally applied [10]. Clock gating is about disabling the clock tree for hardware elements, which are actually not required and thus reduces the dynamic power consumption significantly. This task is automatically inserted by the synthesis tool. The operand isolation technique focuses on data path operators and their dynamic power consumption in case their results are not required in the actual cycle. As said before, some adders in the fourth and fifth cycle have no function in particular and therefore are equipped manually with operand isolation.

6. RESULTS

In order to achieve a detailed evaluation, an IC implementation is performed in this work. Thus, logical and physi-

Table 2. Comparison of the IC synthesis results to actual references.

Reference	Configuration	SFDR [dBc]	Technology [nm]	Frequency [MHz]	Latency [cycles]	Norm. Area [10^5]	Power [μW/MHz]
This work	**nonuniform**	**52.8**	**130**	**181**	**5**	**20.16**	**18.62**
[5]	ROM	90	130	132	4	94.6	63.64
[6]	CORDIC	93.3	250	385	5	35.20	400

cal synthesis is done, considering timing back-annotation as well as parasitic effects. In addition, the results are compared to actual references. The CFO compensation unit reaches a maximum frequency of 181MHz. Considering the latency of five cycles, CFO compensation results can be computed with 36.2MHz, which is sufficient e.g. for the LTE communication standard. A power consumption of $18.62\mu\mathrm{W/MHz}$ is obtained, as well as a total chip area of $0.034\mathrm{mm}^2$. In order to make a comparison to actual references as fair as possible, also the normalized area is regarded here, which is introduced in [11]. This technique considers the technology size, by calculating the quotient of chip area over the squared technology size. Thus, for our approach, a normalized area of $20.16 \cdot 10^5$ is reached. Compared to other approaches, our proposal achieves the best values considering power consumption and chip area. An overview is given in Tab. 2.

7. CONCLUSION

In this paper a Carrier Frequency Offset (CFO) compensation unit is presented, which uses Multiplier-less nonuniform Piecewise function Approximation (MPA). In detail, the phase-to-sine mapper, required by the Numerically Controlled Oscillator (NCO), is realized by linear function approximation working on appropriate sub-functions with restricted variable sizes each. Also gradient quantization is performed by superposition of hardware efficient shifts. Further complexity reduction is achieved by the simplification of the vector rotation multiplier. Thus, a Reduced Radix-4 Multiplier (RRM) is installed, that processes 12 and 16 bit operands with six adders and an average accuracy of $\Delta\mathrm{E} = 2^{-12}$. These simplifications allow an efficient installation of time sharing means for CFO processing. Thus, for a total amount of six adders, a latency of only five cycles is possible. This HDL implementation has been simulated by generating Bit Error Rate (BER) curves over E_b/N_0, which delivered hardly any loss of accuracy. In a second step, the design is logically and physically synthesized. The comparison to actual references, proved our CFO compensation design to be a highly efficient and powerful design, regarding hardware complexity as well as power consumption.

8. REFERENCES

[1] H. Bolcskei, "MIMO-OFDM wireless systems: basics, perspectives, and challenges," *IEEE Wireless Communications*, vol. 13, no. 4, pp. 31–37, Aug. 2006.

[2] P.H. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Transactions on Communications*, vol. 42, no. 10, pp. 2908–2914, Oct 1994.

[3] T. M Schmidl and D. C Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, Dec. 1997.

[4] O. Gustafsson and K. Johanson, "Multiplierless piecewise linear approximation of elementary functions," in *Fortieth Asilomar Conference on Signals, Systems and Computers (ACSSC '06)*, 29 2006-Nov. 1 2006, pp. 1678–1681.

[5] Wen Fan and Chiu-Sing Choy, "Power efficient and high speed frequency synchronizer design for MB-OFDM UWB," in *IEEE International Conference on Ultra-Wideband (ICUWB 2009)*, Sept. 2009, pp. 669–673.

[6] D. De Caro, N. Petra, and A. G. M. Strollo, "A 380 MHz direct digital synthesizer/mixer with hybrid CORDIC architecture in 0.25 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 1, pp. 151–160, Jan. 2007.

[7] D. De Caro, N. Petra, and A.G.M. Strollo, "Direct digital frequency synthesizer using nonuniform piecewise-linear approximation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 10, pp. 2409–2419, Oct. 2011.

[8] Bar-Giora Goldberg, *Direct digital frequency synthesis demystified*, LLH Technology Publishing, 1999.

[9] Behrooz Parhami, *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, 2010.

[10] Michael Keating, *Low power methodology manual: for system-on-chip design*, Series on integrated circuits and systems. Springer, New York, NY, 2008.

[11] A. Ashrafi, R. Adhami, and A. Milenkovic, "A direct digital frequency synthesizer based on the quasi-linear interpolation method," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 4, pp. 863–872, April 2010.