MONTE CARLO LIMIT CYCLE CHARACTERIZATION

*D. Luengo, D. Osés**

Univ. Politécnica de Madrid
Dep. of Circuits and Sytems Engineering
Madrid, Spain

*L. Martino**

University of Helsinki
Dep. of Mathematics and Statistics
Helsinki, Finland

ABSTRACT

The fixed point implementation of IIR digital filters usually leads to the appearance of zero-input limit cycles, which degrade the performance of the system. In this paper, we develop an efficient Monte Carlo algorithm to detect and characterize limit cycles in fixed-point IIR digital filters. The proposed approach considers filters formulated in the state space and is valid for any fixed point representation and quantization function. Numerical simulations on several high-order filters, where an exhaustive search is unfeasible, show the effectiveness of the proposed approach.

Index Terms— IIR filters, finite wordlength effects, limit cycles, Monte Carlo methods.

1. INTRODUCTION

The fixed point implementation of IIR digital filters leads to the appearance of many undesirable finite wordlength effects that degrade the performance of the system: quantization noise, deviation from the desired frequency response due to coefficient sensitivity, appearance of zero-input limit cycles, etc. [1]. In this paper we focus on limit cycles (LCs), which can hinder the performance of an IIR filter substantially, especially in devices requiring a low-power consumption and thus an implementation with a reduced number of bits.

The state-space formulation of an LTI single-input single-output (SISO) IIR digital filter is [2, 3]

$$\mathbf{w}[n+1] = \mathbf{A}\mathbf{w}[n] + \mathbf{b}x[n], \tag{1}$$

$$y[n] = \mathbf{c}^\top \mathbf{w}[n] + dx[n], \tag{2}$$

where $x[n]$ and $y[n]$ denote the n-th sample of the input and output respectively, $\mathbf{w}[n] = [w_1[n], \ \ldots, \ w_M[n]]^\top$ is the $M \times 1$ state vector at instant n, $\mathbf{A}$ is the $M \times M$ state transition matrix, $\mathbf{b}$ and $\mathbf{c}$ are the $M \times 1$ input and output transfer vectors respectively, and d is the scalar feedforward gain. Under zero-input conditions (i.e., $x[n] = 0$), Eq. (1) becomes

$$\mathbf{w}[n+1] = \mathbf{A}\mathbf{w}[n] = \mathbf{A}^2\mathbf{w}[n-1] = \cdots = \mathbf{A}^{n+1}\mathbf{w}[0]. \tag{3}$$

Making use of the eigen-value decomposition of $\mathbf{A}$, Eq. (3) can be alternatively expressed as [4]

$$\mathbf{w}[n+1] = \mathbf{U}\mathbf{\Lambda}^{n+1}\mathbf{U}^\top\mathbf{w}[0], \tag{4}$$

where $\mathbf{U}$ is an $M \times M$ unitary matrix whose columns contain the eigen-vectors of $\mathbf{A}$ and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ \ldots, \ \lambda_M)$ is the $M \times M$ diagonal matrix with the corresponding eigen-values. Hence, if the filter is stable (i.e., $|\lambda_m| < 1$ for $1 \leq m \leq M$ [3]) and $x[n] = 0$, $\mathbf{\Lambda}^{n+1} = \text{diag}(\lambda_1^{n+1}, \ \ldots, \ \lambda_M^{n+1}) \to \mathbf{0}$ as $n \to \infty$, implying from Eq. (4) that $\mathbf{w}[n+1] \to \mathbf{0}$ as $n \to \infty$, i.e., any initial state $\mathbf{w}[0] \neq \mathbf{0}$ should eventually reach the zero-state after a transient.

In a fixed-point implementation, the output of the system has to be quantized. Given a quantization function $Q(x)$, if a double-length accumulator is available (type 1 realization in [5]), Eq. (3) becomes

$$\mathbf{w}[n+1] = Q\left(\mathbf{A}\mathbf{w}[n]\right) = \begin{bmatrix} Q\left(\sum_{j=1}^{M} a_{1j} w_j[n]\right) \\ \vdots \\ Q\left(\sum_{j=1}^{M} a_{Mj} w_j[n]\right) \end{bmatrix}, \tag{5}$$

with $a_{ij} = \mathbf{A}(i,j)$ denoting the (i,j)-th element of $\mathbf{A}$.[1] Several fixed point representations and quantizer types can be considered. Here we focus on the magnitude-sign representation and round-off quantizers. Assuming that $w_m[0]$ is quantized using a wordlength of $P+1$ bits (1 sign bit plus P magnitude bits), denoted as $b_{i,m}[0] \in \{0,1\}$ for $0 \leq i \leq P$ and $1 \leq m \leq M$, then we can express $w_m[0]$ as

$$w_m[0] = (-1)^{b_{0,m}[0]} \sum_{i=1}^{P} b_{i,m}[0] \times 2^{i-1} \times \Delta, \tag{6}$$

where $\Delta = 2^{-P}$, $b_{0,m}[0]$ is the sign bit ($b_{0,m}[0] = 0 \Leftrightarrow w_m[0] \geq 0$ and $b_{0,m}[0] = 1 \Leftrightarrow w_m[0] < 0$) and $b_{i,m}[0]$ ($1 \leq i \leq M$ with $i = 1$ and $i = M$ indicating the least and most significant bits (LSB and MSB) respectively) are the magnitude bits.

*Thanks to the Spanish government for funding through projects COMONSENS (CSD2008-00010), COMPREHENSION (TEC2012-38883-C02-01) and DISSECT (TEC2012-38058-C03-01).

[1] In a single precision implementation (type 2 realization in [5]), $w_m[n] = \sum_{j=1}^{M} Q(a_{mj} w_j[n])$. In the sequel we focus on the double precision case, as most modern digital systems contain double precision ALUs.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

It is well-known that the fixed point implementation of a stable IIR digital filter may contain zero-input limit cycles (LCs), s. t. $\mathbf{w}[n+1] \nrightarrow \mathbf{0}$ as $n \to \infty$ when Eq. (5) is iterated [2, 3]. The only way to guarantee that a fixed point IIR filter is free from LCs is through an exhaustive search in the filter's state space [5, 6], which requires exploring up to $S_T = 2^{(P+1)M}$ states, and is thus unfeasible for high-order filters. Many theoretical bounds on the maximum amplitude that can be sustained by an LC (cf. [4, 5, 6, 7]) have been developed, decreasing the number of states to be explored to $S_R = \prod_{m=1}^{M} (2K_m + 1) - 1$, where $K_m \in \mathbb{Z}^+$ is the maximum number of quantization steps that can be reached by $|w_m[n]|$ as $n \to \infty$. However, since the resulting number of states can still be extremely large for high-order filters, some heuristic algorithms that partially explore the state space using a complicated set of rules have been developed [8].

All of these algorithms consider only the detection of LCs and not their characterization, i.e., obtaining important features like the number of different LCs, their maximum amplitudes or their periods. In this paper, we introduce a Monte Carlo algorithm that explores a fixed-point IIR filter's state space in an efficient and systematic way, by taking advantage of the fact that LCs tend to concentrate on low-amplitude states. The proposed approach can be used to characterize any filter formulated in the state space, for any fixed point representation and quantization function.

2. MONTE CARLO LIMIT CYCLE CHARACTERIZATION ALGORITHM

Monte Carlo (MC) methods were introduced in the 1940s to deal with intractable problems in statistical physics [9, 10, 11], and have been extended to a wide range of applications since then [12, 13, 14]. Essentially, an MC approach is based on generating many initial conditions according to a given probability density function (PDF), usually known as *proposal density*, letting them evolve following the rules of the problem under study, and using the final results obtained to estimate the quantities of interest.

As an alternative to exhaustive search or heuristic approaches, here we propose the Monte Carlo limit cycle characterization (MC-LCC) algorithm, which is summarized in Algorithm 1. The algorithm takes as inputs the state transition matrix, $\mathbf{A}$, the quantization function, $Q(x)$, the precision, P, and the proposal PDF used to draw initial states $\mathbf{w}[0]$, $p(\mathbf{w}[0]; \boldsymbol{\theta})$ with $\boldsymbol{\theta}$ denoting the proposal's parameter vector, and returns the set of the limit cycles found, $\mathcal{C}$.

In Algorithm 1, we obtain first the maximum amplitude that can be attained by an LC for each state, $A_m = K_m \Delta$, where K_m can be obtained using one of the many theoretical bounds available [4, 5, 6], and provides us with the minimum number of bits required to represent $w_m[n]$,

$$B_m = \lceil \log_2(K_m + 1) \rceil, \tag{7}$$

with $\lceil x \rceil$ indicating the smallest integer larger or equal than $x \in \mathbb{R}^+$. We use this information to compute the theoretical bound on the period of a limit cycle [4, 5], $N_{\max} = \prod_{m=1}^{M} 2^{(B_m+1)}$, and construct the proposal PDF, $p(\mathbf{w}[0]; \boldsymbol{\theta})$, as shown in Section 3. From the proposal PDF, we generate L initial test filter states and let them evolve using Eq. (5). For each initial filter state, we stop the iteration either when the zero-state or when a limit cycle has been attained (in which case we store it). The function `CheckLC` determines whether a limit cycle has been reached or not. Many possibilities exist for implementing this function [8]. As a simple alternative, we check whether the filter has reached a previously visited state or not after $2^r N_0 < N_{\max}$ iterations for $r = 0, 1, \ldots, R$.

Algorithm 1 MC limit cycle characterization (MC-LCC)

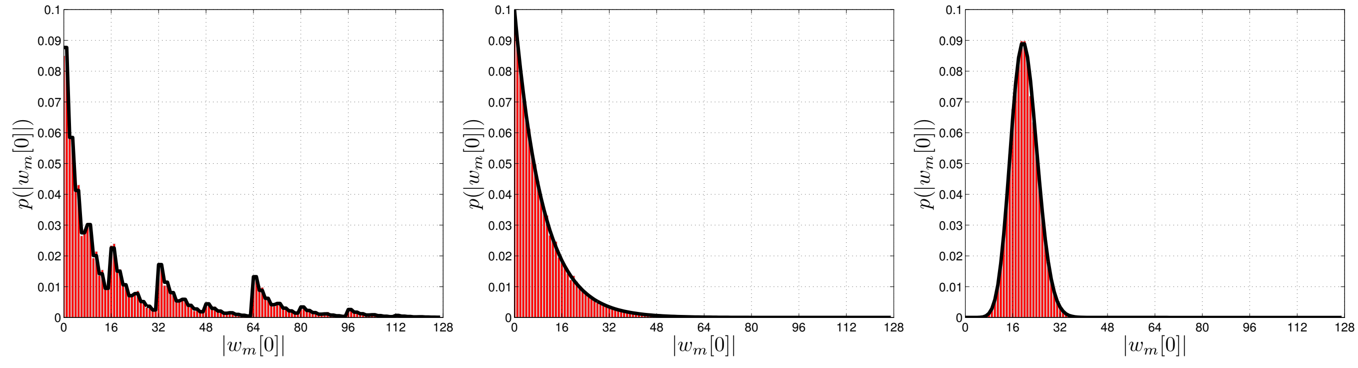
```
Input:
  • $\mathbf{A}$: state transition matrix.
  • $Q(x)$, $P$: quantization function and precision.
  • $p(\mathbf{w}[0]; \boldsymbol{\theta})$: proposal PDF for $\mathbf{w}[0]$.
Algorithm:
  1. Compute $B_m$ ($m = 1, \ldots, M$) and $N_{\max}$, and construct the proposal PDF, $p(\mathbf{w}[0]; \boldsymbol{\theta})$ using Eq. (8).
  2. For $\ell = 1, \ldots, L$:
     (a) Draw $\mathbf{w}^{(\ell)}[0] \sim p(\mathbf{w}[0]; \boldsymbol{\theta})$.
     (b) For $n = 0, \ldots, N_{\max} - 1$:
          i. Obtain $\mathbf{w}^{(\ell)}[n+1]$ using Eq. (5).
         ii. If $\mathbf{w}^{(\ell)}[n+1] = \mathbf{0}$, then Break.
        iii. Else, then CheckLC($\mathbf{w}^{(\ell)}[n+1]$).
Output:
  • $\mathcal{C}$: set of limit cycles found.
```

3. PROPOSAL DENSITIES

The proposal density for $\mathbf{w}[0]$ is constructed as

$$p(\mathbf{w}[0]; \boldsymbol{\theta}) = \prod_{m=1}^{M} p(w_m[0]; \boldsymbol{\theta}), \tag{8}$$

where $\boldsymbol{\theta}$ is the vector containing the parameters of the proposal and $p(w_m[0]; \boldsymbol{\theta}) = p(s_m[0]) p(|w_m[0]|; \boldsymbol{\theta})$ with $s_m[0] = \text{sign}(w_m[0]) = 1 - 2b_{i,m}[0]$. For the sign bit we use an equi-probable distribution, $\Pr\{b_{i,m}[0] = 0\} = \Pr\{b_{i,m}[0] = 1\} = \frac{1}{2}$. For the modulus, we exploit the fact that LCs tend to concentrate on low-amplitudes (as the filter is stable) [5, 8], and consider several possibilities, as described in the following sections.



(a)

(b)

(c)

Fig. 1. Proposal PDFs: **(a)** Eq. (10) with $\alpha = 0.8$; **(b)** Eq. (11) with $\gamma = 0.9$; **(c)** Eq. (14) with $\lambda = 20$.

3.1. Exponential Distribution on the Bit Representation

As a first possibility, we define $p(|w_m[0]|; \alpha, \{B_m\}_{m=1}^M)$, where B_m is given by Eq. (7) and $0 < \alpha \leq 1$ is a parameter controlling the decay of the proposal, through the probability associated to each of the bits used to represent $|w_m[0]|$:

$$\Pr\{b_{i,m}[0] = 1\} = \begin{cases} \alpha^{i-1}/2, & 1 \leq i \leq B_m; \\ 0, & B_m < i \leq P. \end{cases} \tag{9}$$

Note that we select each $b_{i,m}[0]$ and $w_m[0]$ independently from the rest and from any previously selected state. Note also that for the LSB we assign the same probability to a zero and a one, whereas the probability of a zero increases with m (i.e., we penalize high-amplitude initial states). Combining Eqs. (6) and (9), it can be shown that

$$\Pr\{|w_m[0]| = W\} = \prod_{i=1}^{B_m} \left[\frac{\alpha^{i-1}}{2} b_i^* + \left(1 - \frac{\alpha^{i-1}}{2}\right) \overline{b_i^*} \right], \tag{10}$$

where b_i^* is the i-th bit ($1 \leq i \leq B_m$) in the binary representation of W and $\overline{b_i^*}$ denotes the logical not operation on b_i^*. In this case, we have $\mathbb{E}\{|w_m[0]|/\Delta\} = \sum_{i=1}^{B_m} \frac{(2\alpha)^{i-1}}{2}$ and $\text{Var}\{|w_m[0]|/\Delta\} = \frac{1}{16} \sum_{i=1}^{B_m} 2^{2i} \alpha^{i-1} (2 - \alpha^{i-1})$. Fig. 1(a) shows the proposal for $\alpha = 0.8$.

3.2. Exponential Distribution on the Modulus

As a simpler alternative, we consider a discretized exponential distribution directly on $|w_m[0]|$:

$$p(|w_m[0]|; \gamma, \{B_m\}_{m=1}^M) = c\gamma^k, \quad 0 \leq k \leq K_m, \tag{11}$$

where $0 < \gamma \leq 1$ is another decay parameter, and the normalizing constant is $c = \frac{1-\gamma}{1-\gamma^{K_m+1}}$. Now we have

$$\mathbb{E}\left\{\frac{|w_m[0]|}{\Delta}\right\} = \frac{\gamma[1 - (K_m+1)\gamma^{K_m} + K_m\gamma^{K_m+1}]}{(1-\gamma)(1-\gamma^{K_m+1})}, \tag{12}$$

and

$$\begin{aligned} \text{Var}\left\{\frac{|w_m[0]|}{\Delta}\right\} &= \gamma[1 + \gamma - (K_m+1)^2\gamma^{K_m} - K_m^2\gamma^{K_m+2} \\ &+ (2K_m(K_m+1) - 1)\gamma^{K_m+1}][(1-\gamma)^2(1-\gamma^{K_m+1})]^{-1} \\ &- \mathbb{E}\{|w_m[0]|/\Delta\}^2 \,. \end{aligned} \tag{13}$$

This proposal is shown in Fig. 1(b) for $\gamma = 0.9$.

3.3. Poisson Distribution on the Modulus

As a third and final alternative, we consider a Poisson distribution directly on $|w_m[0]|$:

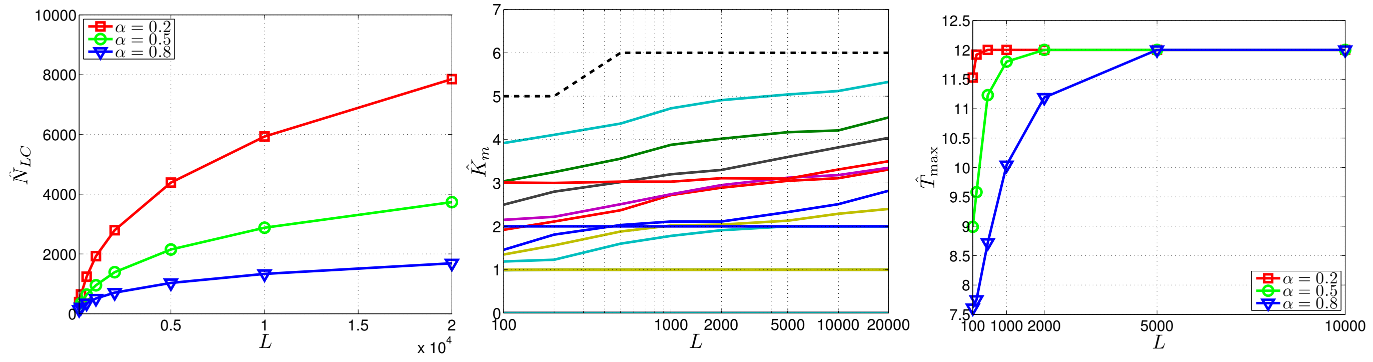
$$p(|w_m[0]|; \lambda, \{B_m\}_{m=1}^M) = \frac{\lambda^k}{k!} \exp(-\lambda), \quad 0 \leq k \leq K_m, \tag{14}$$

where $\lambda > 0$ is a third decay parameter. In this case, for large enough values of K_m, we have $\mathbb{E}\{|w_m[0]|/\Delta\} = \text{Var}\{|w_m[0]|/\Delta\} \approx \lambda$. The proposal is shown in Fig. 1(c) for $\lambda = 20$.

4. NUMERICAL RESULTS

In order to validate the MC-LCC algorithm, we use it to characterize six filters described in the state space:

- **Butt:** Low-Pass Butterworth filter of order $M = 18$ with passband edge frequency $\omega_p = 0.2\pi$ rad, stopband edge frequency $\omega_s = 0.33\pi$ rad, passband ripple $R_p = 0.01$ dB and stopband ripple $R_s = 60$ dB.
- **Cheb1:** Low-Pass Chebyshev filter of order $M = 5$ in [4]: $\omega_p = 0.2022\pi$ rad, $\omega_s = 0.4044\pi$ rad, $R_p = 0.0187$ dB and $R_s = 54$ dB.
- **Cheb2:** Band-Pass Chebyshev filter of order $M = 14$ with passband $[0.3\pi, 0.6\pi]$ rad, stopbands $[0, 0.2\pi]$ and $[0.7\pi, \pi]$ rad, passband ripple $R_p = 0.1$ dB and stopband ripple $R_s = 45$ dB.



(a)

(b)

(c)

Fig. 2. Average results for $N_s = 100$ simulations of the Butterworth filter using the proposal PDF from Eqs. (9) and (10). **(a)** $\hat{N}_{LC}$ for $\alpha \in \{0.2, 0.5, 0.8\}$. **(b)** $\hat{K}_m$ ($m = 1, \ldots, 18$) and $\hat{K}_{\max}$ (dashed line) for $\alpha = 0.2$. **(c)** $\hat{T}_{\max}$ for $\alpha \in \{0.2, 0.5, 0.8\}$.

Table 1. Results for the Butterworth filter: mean ± standard deviation using $N_s = 100$ and $L = 2 \cdot 10^4$. $\mathcal{B}(K_{\max}) = 162$.

Proposal	**Exponential Bits**		**Exponential Modulus**		**Poisson**	
Parameter	**0.2**	**0.8**	**0.3**	**0.9**	**5**	**20**
$\hat{N}_{LC}$	7851.8 ± 147.0	1686.8 ± 57.2	6551.1 ± 137.8	1646.7 ± 52.3	1609.2 ± 61.8	999.0 ± 37.5
$\hat{K}_{\max}$	5.1 ± 0.3	5.3 ± 0.5	5.3 ± 0.5	5.3 ± 0.4	5.0 ± 0.2	5.0 ± 0.4
$\hat{T}_{\max}$	12.0 ± 0.0	12.0 ± 0.0	12.0 ± 0.0	12.0 ± 0.0	12.0 ± 0.0	12.0 ± 0.0

- **Elli1:** Low-Pass elliptic filter of order $M = 5$ in [4]: $\omega_p = 0.2022\pi$ rad, $\omega_s = 0.4044\pi$ rad, $R_p = 0.0187$ dB and $R_s = 54$ dB.
- **Elli2:** Band-Pass elliptic filter of order $M = 6$ in [4]: $\omega_{p1} = 0.168\pi$ rad, $\omega_{p2} = 0.42\pi$ rad, $\omega_{s1} = 0.096\pi$ rad, $\omega_{s2} = \frac{3\pi}{5}$ rad, $R_p = 0.1$ dB and $R_s = 30$ dB.
- **Elli3:** Low-Pass elliptic filter of order $M = 7$ with passband edge frequency $\omega_p = 0.2\pi$ rad, stopband edge frequency $\omega_s = 0.33\pi$ rad, passband ripple $R_p = 0.01$ dB and stopband ripple $R_s = 60$ dB.

For all these filters, we obtain $\mathbf{A}$ and compute the theoretical bound for K_m using [4], $\mathcal{B}(K_{\max}) = \max\{\mathcal{B}(K_m)\}$, which allows us to calculate $B_{\max}$ from (7) and $N_{\max}$ Then we apply the MC-LCC algorithm (using $P = B_{\max}$, $N_0 = 40$, $R = 4$) to estimate: **(1)** the number of states belonging to different LCs, $\hat{N}_{LC}$; **(2)** the maximum number of quantization steps reached by an LC, $\hat{K}_m$ and $\hat{K}_{\max} = \max\{\hat{K}_m\}$; **(3)** the maximum period of any limit cycle, $\hat{T}_{\max}$.

Table 1 shows the results for the Butterworth filter using the three proposal PDFs introduced and different parameters. Note that, although all of them provide similar results in terms of $\hat{K}_{\max}$ and $\hat{T}_{\max}$, the exponential PDFs outperform the Poisson PDF in terms of $\hat{N}_{LC}$ and simulation speed (not shown), as they are more focused on the area where LCs tend to concentrate. Hence, we choose the PDF in Eqs. (9) and (10) to obtain the results for the remaining filters shown in Table 2. Finally, Fig. 1 illustrates the evolution of $\hat{N}_{LC}$, $\hat{K}_m$ and $\hat{T}_{\max}$ for the Butterworth filter as a function of L. Note that, although $\hat{N}_{LC}$ is still increasing for $L = 20000$, with $\alpha = 0.2$ and $L = 500$ we already obtain the same values of $\hat{K}_{\max}$ and $\hat{T}_{\max}$ as using $L = 20000$.

Table 2. Average Results for $N_s = 100$ using the proposal PDF from Eqs. (9) and (10) with $\alpha = 0.2$ and $L = 10^4$.

Filter	**Cheb1**	**Cheb2**	**Elli1**	**Elli2**	**Elli3**
$\hat{N}_{LC}$	42.0	25950.1	66.0	54.0	362.0
$\hat{K}_{\max}$	2.0	17.9	2.0	2.0	63.0
$\mathcal{B}(K_{\max})$	8	207	50	73	754
$\hat{T}_{\max}$	1.0	612.0	6.0	12.0	10.0

5. CONCLUSIONS AND FUTURE LINES

We have introduced a Monte Carlo limit cycle characterization algorithm (MC-LCC) to analyze the limit cycle (LC) behavior of fixed-point IIR digital filters efficiently and in a systematic way. The MC-LCC algorithm provides much more information than traditional LC detection approaches, is applicable to high-order filters and can be adapted to any realization (single or double precision), quantization function (round-off or truncation) and implementation (sign and magnitude or two's complement). Future work includes extending the algorithm to filter structures not formulated in the state space and developing more sophisticated proposal densities.

6. REFERENCES

[1] S. K. Mitra, *Digital Signal Processing. A Computer-Based Approach*, McGraw-Hill, 4th edition, 2011.

[2] D. Schlichthärle, *Digital Filters. Basics and Design*, Springer-Verlag, 2nd edition, 2011.

[3] P. Diniz, E. da Silva, and S. L. Netto, *Digital Signal Processing. System Analysis and Design*, Cambridge University Press, 2002.

[4] D. Osés, F. Cruz-Roldán, and M. Utrilla-Manso, "Tighter limit cycle bounds for digital filters," *IEEE Signal Processing Letters*, vol. 13, no. 3, pp. 149–152, Mar. 2006.

[5] P.H. Bauer and L.-J. Leclerc, "A computer-aided test for the absence of limit cycles in fixed-point digital filters," *IEEE Transactions on Signal Processing*, vol. 39, no. 11, pp. 2400–2410, Nov. 1991.

[6] K. Premaratne, E. C. Kulasekere, P.H. Bauer, and L.-J. Leclerc, "An exhaustive search algorithm for checking limit cycle behavior of digital filters," *IEEE Transactions on Signal Processing*, vol. 44, no. 10, pp. 2405–2412, Oct. 1996.

[7] B.D. Green and L. E. Turner, "New limit cycle bounds for digital filters," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 4, pp. 365–374, Apr. 1988.

[8] M. Utrilla-Manso, F. López-Ferreras, D. Osés, and P. Martín-Martín, "A computer-aided test for the characterization of parasitic oscillations in IIR digital filters," in *Proc. 2nd Int. Symposium on Image and Signal Processing and Analysis (ISPA)*, Pula (Croatia), June 19–21, 2001, pp. 475–478.

[9] N. Metropolis and S. Ulam, "The Monte Carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, Sep. 1949.

[10] C. C. Hurd, "A note on early Monte Carlo computations and scientific meetings," *Annals of the History of Computing*, vol. 7, no. 2, pp. 141–155, Apr. 1985.

[11] N. Metropolis, "The beginning of the Monte Carlo method," *Los Alamos Science*, vol. 15, pp. 125–130, 1987.

[12] X. Wang, R. Chen, and J. S. Liu, "Monte Carlo Bayesian signal processing for wireless communications," *Journal of VLSI Signal Processing*, vol. 30, pp. 89–105, 2002.

[13] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2nd edition, 2004.

[14] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 152–170, Nov. 2005.