Performance Bound Analysis of Analog Circuits Considering Process Variations *

Zhigang Hao[†§], Sheldon X.-D. Tan[§], Ruijing Shen[§] and Guoyong Shi[†]
[§]Department of Electrical Engineering, University of California,Riverside, CA 92521, USA
[†]School of Microelectronics, Shanghai Jiao Tong University, Shanghai, 200240, China
zhigang.g.hao@gmail.com, stan@ee.ucr.edu, rshen@ee.ucr.edu, shiguoyong@ic.sjtu.edu.cn

ABSTRACT

In this paper, we propose a new performance bound analysis of analog circuits considering process variations. We model the variations of component values as intervals measured from tested chip and manufacture processes. The new method applies a graph-based symbolic analysis and affine interval arithmetic to derive the variational transfer functions of analog circuits (linearized) with variational coefficients in forms of intervals. Then the frequency response bounds (maximum and minimum) are obtained by performing analysis of a finite number of transfer functions given by the Kharitonov's polynomial functions. We show that symbolic de-cancellation is critical for the affine interval analysis. The response bound given by the Kharitonov's functions are conservative given the correlations among coefficient intervals in transfer functions. Experimental results demonstrate the effectiveness of the proposed compared to the Monte Carlo method.

Categories and Subject Descriptors

B.8.2 [**Performance and Reliability**]: Performance Analysis and Design Aids

General Terms

Algorithm, Performance

Keywords

performance bound, symbolic, interval, process variation

1. INTRODUCTION

Analog and mixed-signal circuits are very sensitive to the process variations as many matchings are required. This situation becomes worse as technology continues to scale to 90nm and below owing to the increasing process-induced variability [9, 13]. Transistor level mismatch is the primary obstacle to reach a high-yield rate for analog designs in sub-90nm technologies. For example, due to an inverse-square-root-law dependence with the transistor area, the mismatch of CMOS devices nearly doubles for each process generation less than 90nm [4, 7]. Since the traditional worst-case or corner-case based analysis is too pessimistic to sacrifice the speed, power, and area, the statistical approach [11] thereby becomes a trend to estimate the analog mismatch and performance variations. The variations in the analog components can come from systematic (or global spatial variation) ones and stochastic (or local random variation) ones. In this paper, we model both variations as the parameter intervals on the components of analog circuits.

Analog circuit designers usually perform a Monte-Carlo (MC) analysis to analyze the stochastic mismatch and predict the variational responses of their designs under faults. As MC analysis requires a large number of repeated circuit simulations, its computational cost is expensive. Moreover, the pseudo-random generator in MC introduces numerical noises that may lead to errors. More efficient variational analysis, which can give the performance bounds, is highly desirable.

Bounding or worst case analysis of analog circuits under parameter variations has been studied in the past for fault-driven testing and tolerance analysis of analog circuits [5, 17, 22]. The proposed approaches includes sensitivity analysis [23], the sampling method [18], and interval arithmetic based approaches [5,12,17,22]. But sensitivity based method can't give the worst case in general and the sampling based method is limited to a few variables. And interval arithmetic methods are in general have the reputation of overly pessimistic in the past. Recently worst-case analysis of linearized analog circuits in frequency domain has been proposed [12], where Kharitonov's functions [3] was applied to obtain the performance bounds in frequency domain, but no systemic method was proposed to obtain variational transfer functions.

In this paper, we propose a new performance bound analysis algorithm of analog circuits considering the process variations. The new method employs several techniques to compute the bounding responses of analog circuits in the frequency domain. First the new method model the variations of component values as intervals measured from tested chip and manufacture processes. Then the new method applies determinant decision diagram (DDDs) graph-based symbolic analysis to derive the exact symbolic transfer functions from linearized analog circuits. After this, affine interval arithmetic is applied to compute the variational transfer

*This research was supported in part by NSF grants under No. CCF-1017090, No. OISE-0929699, and OISE-1051797 in part by National Natural Science Foundation of China (NSFC) grant under No. 60828008 and 60876089 and in part by the China Ministry of Education Special Fund for Doctoral Education Program (2010).

DAC 2011, June 5-10, 2011, San Diego, California, USA.

functions of the analog circuit with variational coefficients in forms of intervals. Finally, the frequency response bounds (maximum and minimum) are obtained by performing evaluations of a finite number of special transfer functions given by the Kharitonov's theorem, which shows the proved response bounds for given interval polynomial functions in frequency domain. We show that symbolic de-cancellation is critical for reducing inherent pessimism in the affine interval analysis. We also show that response bounds given by the Kharitonov's functions are conservative given the correlations among coefficient intervals in transfer functions. Experimental results demonstrate the effectiveness of the proposed compared to the Monte Carlo method.

The rest of this paper is organized as follows: Section 2 gives a review on interval arithmetic and affine arithmetic. Our proposed performance bound analysis method is presented in Section 3. Section 4 shows the experimental results and Section 5 concludes this paper.

2. REVIEW ON INTERVAL ARITHMETIC AND AFFINE ARITHMETIC

Interval arithmetic was introduced by Moore in the 1960s [8] to solve range estimation considering uncertainties. In interval arithmetic, a classical variable x is represented by an interval $\hat{x} = [x^-, x^+]$ which satisfies $x^- \leq x \leq x^+$. However, the interval arithmetic suffers the overestimation problem as it often yields an interval that is much wider than the exact range of the function.

As an example, given $\hat{x} = [-1, 1]$, the interval evaluation of $\hat{x} - \hat{x}$ produces $[-1-1, 1-(-1)] = [-2, 2]$ instead of $[0, 0]$, which is the actual range of that expression.

Affine arithmetic is proposed by Stolfi and de Figueiredo [2] to overcome the error explosion problem of standard interval analysis. In affine interval, the affine form $\hat{x}$ of random variable x is given by

$$\hat{x} = x_0 + \sum_{i=1}^{n} x_i \varepsilon_i, \tag{1}$$

in which each *noise symbol* $\varepsilon_i (i = 1, 2, \ldots, n)$ is an independent component of the total uncertainties of x which satisfies $-1 \leq \varepsilon_i \leq 1$, the coefficient x_i is the magnitude of ε_i and x_0 is the *central value* of $\hat{x}$. The conversion from affine intervals to classical intervals is easy as $\hat{x}$ in (1) can be converted to $[x_0 - rad(\hat{x}), x_0 + rad(\hat{x})]$ in which $rad(\hat{x}) = \sum_{i=1}^{n} |x_i|$ is defined as the radius of the affine expression $\hat{x}$. Basic operation of addition and subtraction of affine arithmetic is defined by

$$\hat{x} \pm \hat{y} = (x_0 \pm y_0) + \sum_{i=1}^{n} (x_i \pm y_i)\varepsilon_i. \tag{2}$$

Return to the previous example, if x has the affine form $\hat{x} = 0 + \varepsilon_1$ then $\hat{x} - \hat{x} = \varepsilon_1 - \varepsilon_1 = 0$ gives the accurate result. Affine arithmetic multiplication is defined as

$$\hat{x} \cdot \hat{y} = x_0 \cdot y_0 + \sum_{i=1}^{n} (x_0 \cdot y_i + x_i \cdot y_0)\varepsilon_i + rad(\hat{x}) \cdot rad(\hat{y}) \cdot \varepsilon_{n+1}, \tag{3}$$

in which ε_{n+1} is a new noise symbol that is distinct from all the other noise symbols $\varepsilon_i (i = 1, 2, \ldots, n)$. We notice that affine operations mitigate the problem associated with symbolic cancellations in addition, but for multiplication, the symbolic cancellation can still exist. For instance, $\hat{x} \cdot \hat{y}$ - $\hat{y} \cdot \hat{x} = 0$, but they will generate two different ε_{n+1}'s when multiplication is done first and the complete cancellation will not happen any more.

3. THE PROPOSED PERFORMANCE BOUND ANALYSIS METHOD

We first present the whole flow of the proposed new performance bound analysis algorithm in Fig. 1. Basically the proposed method consists of two major computing steps. The first step is to compute the variational transfer functions from the variational circuit parameters, which will be done via DDD-based symbolic analysis method and affine interval arithmetic (Steps 1-3). Second we compute the frequency response bounds via Kharitonov's functions, which just requires a few transfer function evaluations (Step 4). Kharitonov's functions can lead to approved upper and lower bounds for the frequency domain responses for a variational transfer function. We will present the two major computing steps in the following subsections.

Algorithm: NEW PERFORMANCE BOUND ANALYSIS OF ANALOG CIRCUITS

Input: netlist, affine interval of selected circuit parameters.
Output: conservative performance bound (magnitude, phase) of the circuit.

1. Parse the netlist using modified nodal analysis and construct the corresponding determinant decision diagram (DDD).
2. Expend the DDD into s-expended DDD and perform the term de-cancellation.
3. Evaluate the s-expended DDD using affine interval arithmetic to obtain the interval coefficients of the nominator and denominator of the transfer function.
4. Using the Kharitonov's functions to obtain the magnitude and phase bounds of the transfer function in frequency domain.

Figure 1: The flow of proposed algorithm.

3.1 Variational transfer function computation

In this subsection, we first provide a brief overview of determinant decision diagram (DDD) [15]. Next we show how affine arithmetic can be applied to compute the variational transfer function.

3.1.1 Symbolic analysis by determinant decision diagrams

Determinant Decision Diagrams [15] are compact and canonical graph-based representation of determinants. The concept is best illustrated using a simple RC filter circuit shown in Fig. 2.

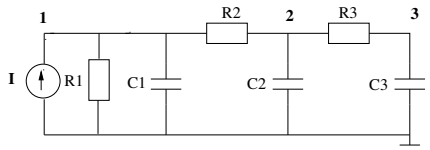

Figure 2: An example circuit.

Its system equations can be written as

$$\begin{bmatrix} \frac{1}{R_1} + sC_1 + \frac{1}{R_2} & -\frac{1}{R_2} & 0 \\ -\frac{1}{R_2} & \frac{1}{R_2} + sC_2 + \frac{1}{R_3} & -\frac{1}{R_3} \\ 0 & -\frac{1}{R_3} & \frac{1}{R_3} + sC_3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix}$$

We view each entry in the circuit matrix as one distinct symbol, and rewrite its system determinant in the left-hand side of Fig. 3. Then its DDD representation is shown in the right-hand side.

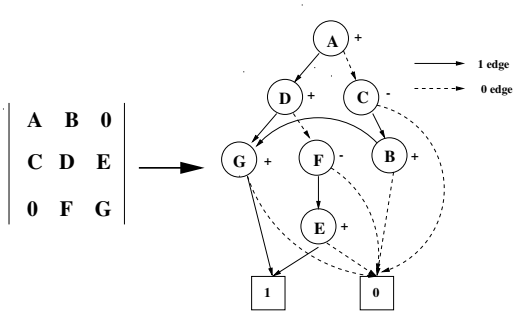

Figure 3: A matrix determinant and its DDD representation.

A DDD is a signed, rooted, directed acyclic graph with two terminal vertices, namely the *0-terminal* vertex and the *1-terminal* vertex. Each non-terminal DDD vertex is labeled by a symbol in the determinant denoted by a_i (A to G in Fig. 3), and a positive or negative sign denoted by $s(a_i)$. It *originates* two outgoing edges, called *1-edge* and *0-edge*. Each vertex a_i represents a symbolic expression $D(a_i)$ defined recursively as follows:

$$D(a_i) = a_i \cdot s(a_i) \cdot D_{a_i} + D_{\overline{a}_i}, \tag{4}$$

where D_{a_i} and $D_{\overline{a}_i}$ represent, respectively, the symbolic expressions of the vertices pointed by the 1-edge and 0-edge of a_i. The 1-terminal vertex represents expression 1, whereas the 0-terminal vertex represents expression 0. For example, vertex E in Fig. 3 represents expression E, and vertex F represents expression $-EF$, and vertex D represents expression $DG - FE$. We also say that a DDD vertex D represents an expression defined the DDD subgraph rooted at D.

A *1-path* in a DDD corresponds with a product term in the original DDD, which is defined as a path from the root vertex (A in our example) to the 1-terminal including all symbolic symbols and signs of the vertices that originate all the 1-edges along the 1-path. In our example, there exist three 1-paths representing three product terms: ADG, $-AFE$ and $-CBG$. The root vertex represents the sum of these product terms. Size of a DDD is the number of DDD vertices, denoted by $|DDD|$.

Once a DDD has been constructed, its numerical values of the determinant it represents can be computed by performing the depth-first type search of the graph and performing (4) at each node, whose time complexity is linear function of the size of the graphs (its number of nodes). The computing step is call *Evaluate(D)* where D is a DDD root. With proper node ordering and hierarchical approaches, DDD can be very efficient to compute transfer functions of large analog circuits [15, 19].

In order to compute the symbolic coefficients of the transfer function in different powers of s, the original DDD can be expended to the *s-expanded* DDD [16]. By doing this, each coefficient of the transfer function is represented by a *coefficient* DDD. The s-expanded DDD can be constructed from the complex DDD in linear time in the size of the original complex DDD [16].

3.1.2 Variational transfer function

Assume that each circuit parameter $\hat{x}$ becomes an affine interval $\hat{x} = x_0 + \sum_{i=1}^{n} x_i \varepsilon_i$ due to process variations, now we want to compute the variational transfer function. The resulting transfer function will take the following s-expanded rational form:

$$H(s) = \frac{N(s)}{D(s)} = \frac{\sum_{i=0}^{m} \hat{a}_i s^i}{\sum_{j=0}^{n} \hat{b}_j s^j}, \tag{5}$$

where coefficients $\hat{a}_i$ and $\hat{b}_j$ are all affine intervals. This can be computed by means of affine arithmetic [2]. Basically the DDD *Evaluation* operation traverses the DDD in a depth-first style and performs one multiplication and one addition at each node as shown in (4). Now the two operations will be replaced by the addition and multiplication from affine arithmetic.

3.1.3 Symbolic de-cancellation in DDD evaluation using affine arithmetic

As mentioned before that the interval and affine arithmetic operations are very sensitive to the symbolic term cancellations, which, however, have significant presences in the DDD and s-expanded DDD. It was shown that about 70%-90% terms in the determinant of a MNA-formulated circuit matrix are canceling terms [20]. Notice that symbolic cancellation always happens even in the presence of parameter variations.

In DDD evaluation, we have both addition and multiplication as shown in (4). Cancellation can lead to large errors if not removed. For example, considering two terms $\hat{x} \cdot \hat{y} \cdot \hat{z}$ and $\hat{z} \cdot \hat{y} \cdot (-\hat{x})$, and suppose $\hat{x} = 1 + \varepsilon_1, \hat{y} = 1 + \varepsilon_2, \hat{z} = 1 + \varepsilon_3$, then

$$\begin{aligned} \hat{x} \cdot \hat{y} \cdot \hat{z} &= (1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_4) \cdot \hat{z} \\ &= 1 + \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 + 3\varepsilon_5, \\ \hat{z} \cdot \hat{y} \cdot (-\hat{x}) &= (1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_6) \cdot (-\hat{x}) \\ &= -1 - \varepsilon_1 - \varepsilon_2 - \varepsilon_3 - \varepsilon_6 - 3\varepsilon_7. \end{aligned}$$

However, the addition of these two terms is

$$\hat{x} \cdot \hat{y} \cdot \hat{z} + \hat{z} \cdot \hat{y} \cdot (-\hat{x}) = \varepsilon_4 + 3\varepsilon_5 - \varepsilon_6 - 3\varepsilon_7, \tag{6}$$

which should be 0. The reason is that in affine multiplication defined in (3), the new noise symbol is actual a function of the original noise symbols $\varepsilon_i (i = 1, 2, \ldots, n)$, but affine arithmetic assumes the new symbol is independent from the original ones. As a result, the symbolic canceling terms will result to inaccurate results, which can be as large as $[-8, 8]$ for (6).

Fortunately, we can perform the de-cancellation operation on coefficient DDDs in the s-expanded DDDs in a very efficient way during or after the coefficient DDD construction, so that the resulting coefficient DDD is cancellation free [20], which can significantly improve the interval computation accuracy as shown in the experimental results.

3.1.4 Increase the accuracy of affine arithmetic by considering second order noise symbols

The affine arithmetic used in DDD evaluation is addition and multiplication. The affine addition is accurate as it does not add any new noise symbol. However, for affine multiplication shown in (3), every time a new noise symbol ε_{n+1} is added and this process will reduce the accuracy of the bound of affine arithmetic compared with real bound. In our implementation, we store the coefficients of first order as well as second order noise symbols and we only add new

noise symbol for higher orders. The affine multiplication in (3) is changed to:

$$\hat{x}\hat{y} = x_0y_0 + \sum_{i=1}^{n}(x_0y_i + x_iy_0)\varepsilon_i + \sum_{i=1}^{n} x_iy_i\varepsilon_i^2 + \sum_{i=1}^{n}\sum_{j=i+1}^{n}(x_iy_j + x_jy_i)\varepsilon_i\varepsilon_j. \tag{7}$$

For simplicity, assume $x^-, x^+, x_i, y^-, y^+, y_i > 0 (i = 0, 1, \cdots, n)$, the bound of $\hat{x}\hat{y}$ in (7) is $[x_0y_0 - rad_1, x_0y_0 + rad_2]$, in which

$$rad_1 = \sum_{i=1}^{n}(x_0y_i + x_iy_0) - \sum_{i=1}^{n}\sum_{j=1}^{n} x_iy_j, \tag{8}$$

$$rad_2 = \sum_{i=1}^{n}(x_0y_i + x_iy_0) + \sum_{i=1}^{n}\sum_{j=1}^{n} x_iy_j, \tag{9}$$

which is more accurate than the bound $[x_0y_0 - rad_2, x_0y_0 + rad_2]$ obtained by original affine multiplication in (3). For other combinations of the value of $x^-, x^+, x_i, y^-, y^+, y_i$, the accuracy of affine multiplication can also be increased accordingly via considering second order noise symbols.

3.2 Performance bound computation via Kharitonov's functions

Given a transfer function with variational coefficients, one can perform Monte Carlo based approach to compute the variational responses in frequency domain. However, more efficient works can be done via Kharitonov's functions which is only a few, but can give the approved bounds of the responses in frequency domain.

Kharitonov's seminar work proposed in 1978 [3] was originally concerned with the stability issues of a polynomial (with real coefficients) with coefficient uncertainties (due to perturbations). He showed that one needs to verify only four special polynomials to ensure that all the variational polynomials are stable.

Specifically, given a family of polynomials with real and variational coefficients

$$P(s) = p_0 + p_1s + \ldots + p_ns^n, p_i^- \leqslant p_i \leqslant p_i^+, i = 0, \cdots, n. \tag{10}$$

Then the four special Kharitonov's functions are:

$$Q_1(j\omega) = P_{emin}(\omega) + jP_{omin}(\omega) \tag{11}$$

$$Q_2(j\omega) = P_{emin}(\omega) + jP_{omax}(\omega) \tag{12}$$

$$Q_3(j\omega) = P_{emax}(\omega) + jP_{omin}(\omega) \tag{13}$$

$$Q_4(j\omega) = P_{emax}(\omega) + jP_{omax}(\omega), \tag{14}$$

where

$$P_{emin}(\omega) = p_0^- - p_2^+\omega^2 + p_4^-\omega^4 - p_6^+\omega^6 + \cdots \tag{15}$$

$$P_{emax}(\omega) = p_0^+ - p_2^-\omega^2 + p_4^+\omega^4 - p_6^-\omega^6 + \cdots \tag{16}$$

$$P_{omin}(\omega) = p_1^-\omega - p_3^+\omega^3 + p_5^-\omega^5 - p_7^+\omega^7 + \cdots \tag{17}$$

$$P_{omax}(\omega) = p_1^+ - p_3^-\omega^3 + p_5^+\omega^5 - p_7^-\omega^7 + \cdots. \tag{18}$$

One important observation is that the four special functions given by Kharitonov's theorem create a rectangle (called Dasgupta's rectangle) [1] in the complex response domain as shown in Fig. 4(a), where the rectangle has edges in parallel with real and image axis. The four Kharitonov functions (polynomials) correspond to the four corners of the rectangle.

Later Levkovich etc. [6] show that Kharitonov's theorem can be used to calculate the amplitude and phase envelops of a family of interval rational transfer functions of continuous-time systems in frequency domain. The results can be easily interpreted based on the Dasgupta's rectangle (which is also called Kharitonov's rectangle), which can clearly show what is the largest magnitude (the longest distance from origin of the complex plane to one corner of the rectangle). Same thing can be derived for the smallest magnitudes and the bounds of the phase responses.

Specifically, in the complex frequency domain, the magnitude and phase response of Kharitonov's rectangle in the complex plane can be divided into nine states, which is shown in Fig. 4(b) [6]. And the corresponding maximum and minimum magnitude and phase of the nine states are shown in Table. 1.

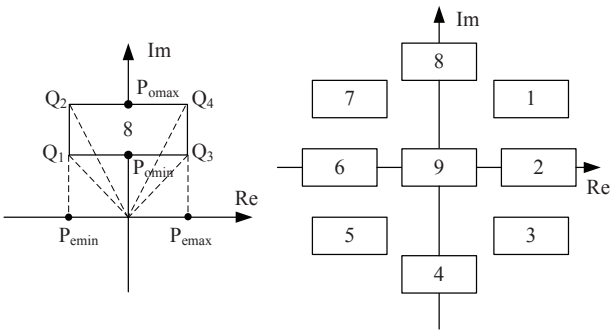

Figure 4: (a) Kharitonov's rectangle in state 8. (b) Kharitonov's rectangle for all 9 states.

Table 1: Extreme values of $|P(j\omega)|$ and $Arg\ P(j\omega)$ for nine states

state	max $\lvert P(j\omega)\rvert$	min $\lvert P(j\omega)\rvert$	max $arg[P(j\omega)]$	min $arg[P(j\omega)]$
1	Q_4	Q_1	Q_2	Q_3
2	Q_3 *or* Q_4	P_{emin}	Q_2	Q_1
3	Q_3	Q_2	Q_4	Q_1
4	Q_1 *or* Q_3	P_{omax}	Q_4	Q_2
5	Q_1	Q_4	Q_3	Q_2
6	Q_1 *or* Q_2	P_{emax}	Q_3	Q_4
7	Q_2	Q_3	Q_1	Q_4
8	Q_2 *or* Q_4	P_{omin}	Q_3	Q_1
9	Q_1 *or* Q_2 *or* Q_3 *or* Q_4	0	2π	0

In Table. 1, $|P(j\omega)|$ and $arg[P(j\omega)]$ are defined as the magnitude and phase of the polynomial $P(j\omega)$. Once the variational transfer function is obtained from (5), the coefficients can be converted from affine interval to classical interval as $\hat{a}_i = [a_i^-, a_i^+]$ and $\hat{b}_j = [b_j^-, b_j^+]$. Afterwards, one can compute the upper and lower bounds of the transfer function easily. Since the maximum and minimum magnitude and phase of nominator $N(s)$ and denominator $D(s)$ has only a few possible cases which is shown in Table 1, it is very straightforward to obtain the magnitude and phase bound of $H(s)$ compared to large sampling based Monte Carlo simulations [6].

It was shown that if all the variational coefficients are not correlated and the value of each coefficient in numerator and denominator belongs to finite real interval, the magnitude and phase bound is precise (real bound) [6], i.e. each bound will be attained by one function in the variational function family. But in our problem, we know that each circuit parameter may contribute to several coefficients during

the evaluations of coefficient DDDs and thus the variational coefficients are not independent.

However, DDD can generate the dominant terms of each coefficient in different powers of s by performing the shortest path algorithm [21]. The shared parameters in the dominant terms can be removed from different coefficients to tighten the affine interval bounds and reduce the correlation between coefficients.

In the experiment part, we show that the bounds given by Kharitonov's theorem are conservative and it indeed cover all the responses from the Monte Carlo simulation results.

4. EXPERIMENTAL RESULTS

We have implemented the proposed method in C++ and the affine arithmetic part is based on [10]. All the experimental results were carried out in a Linux system with quad Intel Xeon CPUs with 3GHz and 16GB memory. The proposed performance bound method was tested on two sample circuits, one is a CMOS low-pass filter (shown in Fig. 5), another is a CMOS cascode op-amp circuit [14] where the small signal model is used to model the MOSFET transistors.

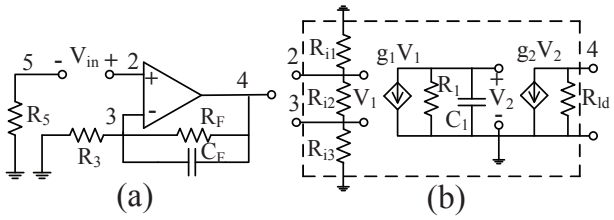

Figure 5: (a)A low-pass filter. (b)A linear model of the op-amp in the low-pass filter.

The information about the complexity of complex DDD, s-expended DDD after symbol de-cancellation are shown in column 1 to 7 in Table. 3, in which $NumP$, $DenP$ are the total numbers of product terms in the numerator and denominator of the transfer function and $|DDD|$ is the size(number of vertices) of the DDD representing both the numerator and the denominator of the transfer function. From the table, we can see that s-expanded DDDs are able to represent a huge number of product terms with a relatively small number of vertices by means of sharing among different coefficient DDDs.

First, we show that term de-cancellation is critical in improving the accuracy for interval bounds in DDD evaluation using affine interval. Table 2 shows the effect of coefficient affine radius reduction considering term de-cancellation for the given two example circuits during the DDD evaluation under different set of variations. Var, Nom, Den represents process variation, nominator and denominator, respectively. As can been seen from the table, the average radius reduction amount is 144% and 64% for nominators and denominators, respectively, and the reduction effect grows with the increasing of process variation. As a result, symbolic de-cancellation can indeed significantly reduce the pessimism of affine arithmetic.

Second, we present the performance of our proposed method. For the low-pass filter example, we introduce 3 noise symbols ε as the local variation source for the $VCCS$, *resistor* and *capacitor* inside linear op-amp model shown in Fig. 5(b). And we introduce another 4 noise symbols ε for other devices of the filter as global variation. For the cascode op-amp example, we introduce 3 noise symbols ε for the $VCCS$, *resistor* and *capacitor* inside the small signal model for each MOSFET transistor as local variation source and introduce another 6 noise symbols ε for other devices in the op-amp as global variations. The total number of noise symbols for each testing circuit is shown in the 8^{th} column in Table 3. As an DDD expression is exactly symbolic and does not have any approximations, it is proved to be accurate compared with SPICE (which uses the simple linearized device models). In our experiments, we compare our result with the Monte Carlo simulations using DDD. We tested our algorithm on different global/local variation pairs as is shown in column 9. We introduce the *bound range*, which is the average value of the result of the bound of the Monte Carlo simulation divided by the bound of the proposed method.

Shown in Fig. 6 and Fig. 7 are the two results for comparison for the proposed and the Monte Carlo method under 10% global, 10% local variation and 5% global, 10% local variation, in which $Affine\ DDD$ is our proposed method and the $Nominal$ is the response of the circuit without parameter variation. During all our simulations, we found that the bound calculated by Kharitonov's functions in our method is always the conservative bound compared with Monte Carlo. However, further investigation are needed to obtain tighter bound using affine arithmetic. We chose the Monte Carlo samples to be 10000. The speed up of the new method compared with Monte Carlo is shown in column 12 in the Table 3. The average speed up is $90X$ for given circuits.

Table 2: Summary of coefficient radius reduction with cancellation.

Var.	Ave.		Max.		Min.	
	Nom.	Den.	Nom.	Den.	Nom.	Den.
5%	59%	27%	252%	53%	16%	5.5%
10%	139%	62%	301%	123%	35%	12%
15%	234%	103%	516%	305%	56%	21%

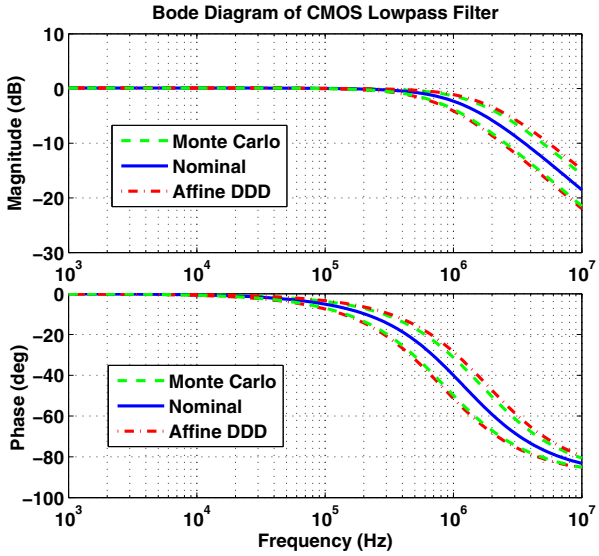

Figure 6: Bode diagram of the CMOS lowpass filter.

5. CONCLUSIONS

Table 3: Summary of DDD information and performance of the proposed method.

1	2	3	4	5	6	7	8	9	10	11	12
circuit	complex DDD			s-expanded DDD			number	global/local	bound range		speed up
	NumP	DenP	$\|DDD\|$	NumP	DenP	$\|DDD\|$	of ε	variation	mag	pha	to MC
Low-pass	5	8	31	7	70	32	7	5% / 10%	95.1%	93.8%	115
								10% / 10%	92.5%	91.9%	101
Cascode	76	216	153	4143	13239	561	30	5% / 10%	83.9%	84.3%	77
								10% / 10%	81.1%	80.2%	68

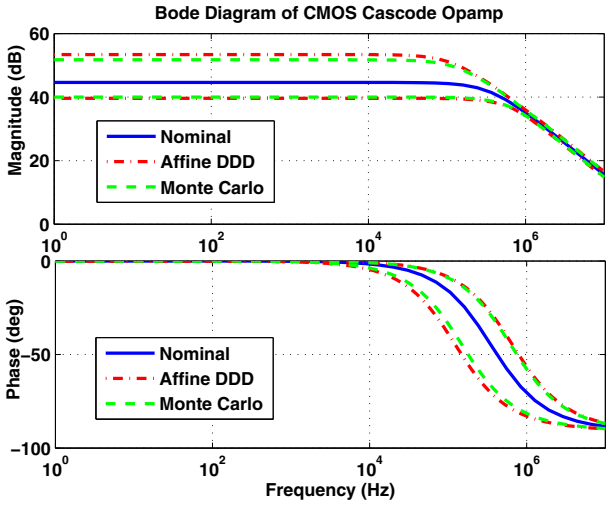

Figure 7: Bode diagram of the CMOS cascode opamp.

We have proposed a new performance bound analysis algorithm of analog circuits considering process variations. The new method applies a graph-based symbolic analysis and affine interval arithmetic to derive the variational transfer functions of linearized analog circuits with variational coefficients. Then the frequency response bounds were obtained by using the Kharitonov's polynomial theorem. We have shown that symbolic de-cancellation is important and necessary to reduce pessimism for affine interval analysis. We also showed that the response bound given by the Kharitonov's functions are conservative given the correlations among coefficient intervals in transfer functions. Experimental results demonstrated the effectiveness of the proposed algorithm compared to the Monte Carlo method.

6. ACKNOWLEDGMENTS

The authors would like to thank Prof. Zhou Dian from University of Texas, Dallas for the valuable introduction and discussion on the Kharitonov's polynomial theorem.

7. REFERENCES

[1] S. Dasgupta, "Kharitonov's theorem revisited," *Systems & Control Letters*, vol. 11, no. 5, pp. 381 – 384, 1988.

[2] L. H. de Figueiredo and J. Stolfi, "Self-validated numerical methods and applications," in *Brazilian Mathematics Colloquium monographs*, IMPA/CNPq, Rio de Janeiro, Brazil, 1997.

[3] V. L. Kharitonov, "Asymptotic stability of an equilibrium position of a family of systems of linear differential equations," *Differential. Uravnen.*, vol. 14, pp. 2086 –2088, 1978.

[4] J. Kim, K. Jones, and M. Horowitz, "Fast, non-monte-carlo estimation of transient performance variation due to device mismatch," in *Proc. IEEE/ACM Design Automation Conference (DAC)*, 2007.

[5] L. Kolev, V. Mladenov, and S. Vladov, "Interval mathematics algorithms for tolerance analysis," *IEEE Trans. on Circuits and Systems*, vol. 35, no. 8, pp. 967 –975, Aug. 1988.

[6] A. Levkovich, E. Zeheb, and N. Cohen, "Frequency response envelopes of a family of uncertain continuous-time systems," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 42, no. 3, pp. 156 –165, Mar. 1995.

[7] H. Masuda, S. Ohkawa, A. Kurokawa, and M. Aoki, "Challenge: Variability characterization and modeling for 65- to 90-nm processes," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2005.

[8] R. E. Moore, *Interval Analysis*. Prentice-Hall, 1966.

[9] S. Nassif, "Model to hardware correlation for nm-scale technologies," in *Proc. IEEE International Workshop on Behavioral Modeling and Simulation (BMAS)*, Sept 2007, keynote speech.

[10] D. C. Olivier Gay and N. Hurst, "Libaffa: C++ affine arithmetic library for gnu/linux. available at http://savannah.nongnu.org/projects/libaa/," May 2005.

[11] M. Pelgrom, A. Duinmaijer, and A. Welbers, "Matching properties of mos transistors," *IEEE J. Solid-State Circuits*, pp. 1433–1439, 1989.

[12] L. Qian, D. Zhou, S. Wang, and X. Zeng, "Worst case analysis of linear analog circuit performance based on Kharitonov's rectangle," in *Proc. IEEE Int. Conf. on Solid-State and Integrated Circuit Technology (ICSICT)*, Nov. 2010.

[13] R. Rutenbar, "Next-generation design and EDA challenges," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, January 2007, keynote speech.

[14] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*. Oxford University Press, USA, 2009.

[15] C.-J. Shi and X.-D. Tan, "Canonical symbolic analysis of large analog circuits with determinant decision diagrams," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 1, pp. 1–18, Jan. 2000.

[16] ——, "Compact representation and efficient generation of s-expanded symbolic network functions for computer-aided analog circuit design," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 7, pp. 813–827, April 2001.

[17] C.-J. R. Shi and M. W. Tian, "Simulation and sensitivity of linear analog circuits under parameter variations by robust interval analysis," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 4, pp. 280–312, July 1999.

[18] R. Spence and R. Soin, *Tolerance Design of Electronic Circuits*. Addison-Wesley, Reading, MA., 1988.

[19] S. X.-D. Tan, W. Guo, and Z. Qi, "Hierarchical approach to exact symbolic analysis of large analog circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 8, pp. 1241–1250, August 2005.

[20] S. X.-D. Tan and C.-J. Shi, "Efficient DDD-based interpretable symbolic characterization of large analog circuits," *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science(IEICE)*, vol. E86-A, no. 12, pp. 3112–3118, Dec. 2003.

[21] ——, "Efficient approximation of symbolic expressions for analog behavioral modeling and analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 6, pp. 907–918, June 2004.

[22] W. Tian, X.-T. Ling, and R.-W. Liu, "Novel methods for circuit worst-case tolerance analysis," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol. 43, no. 4, pp. 272 –278, Apr. 1996.

[23] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. New York, NY: Van Nostrand Reinhold, 1995.