Interval-Valued Reduced-Order Statistical Interconnect Modeling

James D. Ma and Rob A. Rutenbar, *Fellow, IEEE*

***Abstract*—We show how advances in the handling of correlated interval representations of range uncertainty can be used to approximate the mass of a probability density function as it moves through numerical operations and, in particular, to predict the impact of statistical manufacturing variations on linear interconnect. We represent correlated statistical variations in resistance–inductance–capacitance (RLC) parameters as sets of correlated intervals and show how classical model-order reduction methods—asymptotic waveform evaluation and passive reduced-order interconnect macromodeling algorithm—can be retargeted to compute interval-valued, rather than scalar-valued, reductions. By applying a simple statistical interpretation and sampling to the resulting compact interval-valued model, we can efficiently estimate the impact of variations on the original circuit. Results show that the technique can predict mean delay and standard deviation with errors between 5% and 10% for correlated RLC parameter variations up to 35%.**

***Index Terms*—Affine arithmetic, design-for-manufacturing, interconnect simulation, numerical analysis, statistical modeling.**

I. Introduction

With continued technology scaling, semiconductor designs are increasingly sensitive to random manufacturing variations, and the scale of these unavoidable variations continues to increase relative to the so-called "nominal" outcomes we would prefer. This growing problem is compounded by the fact that different manufacturing steps can affect each die *globally* (perturbing all circuit components in a similar way) or *locally* (perturbing spatially nearby clusters of components in a similar way) and that these individual variations may be arbitrarily correlated [1].

Against this background, it is perhaps no surprise that static timing analysis was the first design step intensely studied for how it can be enhanced to handle today's mix of correlated variations [2]–[6]. First, static timing analysis is still the essential sign-off step in many designs; a statistical engine fits neatly into today's design flows. Second—and more relevant to our work in this paper—static timing depends on a very small number of essential statistical "operators." If we can efficiently represent the distribution of arrival times at each input of a gate, we need only be able to compute *sums* and *maximums* of these distributions to propagate a statistical model of worst-case timing through the network of gates. Propagating a set of correlated normal delay distributions seems the most successful approach today [3]–[6], in large part because the sum and maximum of a set of *normal* distributions may be calculated (or approximated) by another normal distribution.

Unfortunately, this powerful technique is not a general technique, although it is amenable to a wide range of numerical computer-aided design (CAD) problems. Consider, for example, the related problem of statistical interconnect analysis. Model-order reduction techniques [7]–[9] pioneered in the last decade work remarkably well to approximate complex interconnect with low-order linear models, which can then be analyzed for useful results, e.g., interconnect delay. It is well understood that wire delay is a vital component of overall circuit delay, which is affected in complex ways by manufacturing variations [10]. However, as compared to the case of static timing, it is not so easy to see how to represent the essential statistics in interconnect analysis. For example, it is easy to add two normal distributions; it is *not* easy to extract the dominant eigenvalues from a matrix whose entries are themselves correlated normal distributions and represent this as yet *another* normal distribution. We can of course fall back on some form of Monte Carlo sampling and simulation over the space of manufacturing variations using a sufficiently fast interconnect modeling tool (e.g., rapid interconnect circuit evaluation [11]). This strategy is general, flexible, straightforward, and widely trusted. It provides the maximum accuracy but at a cost that may be unattractive for large designs and large numbers of random parameters.

Prior specialized approaches to statistical interconnect analysis can be categorized as based on either: 1) model-order reduction [12]–[15] or 2) performance parameter (e.g., delay) extraction [16]–[18]. Liu *et al.* [12] showed how the pole analysis via congruence transformation [8] and passive reduced-order interconnect macromodeling algorithm (PRIMA) [9] interconnect model-reduction methods can be rendered in variational forms and how modest sampling in the space of manufacturing variations yields the data needed to form a set of linear equations whose solution parameterizes these variational models. This technique works with high accuracy on some important problems, e.g., clock tree analysis [10]. As presented, however, the approach handled only global

Manuscript received July 27, 2005; revised April 26, 2006 and October 20, 2006. This work was supported in part by the MARCO Focus Center for Circuit and System Solutions (C2S2) under MARCO Contract 2003-CT-888. This paper was recommended by Associate Editor L. M. Silveira.

J. D. Ma was with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now with IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: djma@us.ibm.com).

R. A. Rutenbar is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: rutenbar@ece.cmu.edu).

Digital Object Identifier 10.1109/TCAD.2007.895577

independent variations. Daniel *et al.* [13] and Wang *et al.* [14] developed control-theoretic approaches that approximate system transfer function by low-order weakly nonlinear polynomials of process parameters. However, the multiparameter moment-matching algorithm [13] may result in extremely large model size, particularly when there exist many correlated random process parameters. Phillips [15] extends the truncated balanced realization scheme [19] for variational model-order reduction by constructing the variational Grammian via very economic sampling over the manufacturing space. In [16], Agarwal *et al.* represented random parameters in a form similar to the first-order Taylor series and used these to derive low-order analytical formulas for interconnect delay metrics. Li *et al.* [17] proposed an asymptotic probability extraction methodology for nonnormal distributions of circuit performance (e.g., delay) via binomial evaluation of high-order moments that are matched to approximate the characteristic function of the circuit performance by a rational function. Finally, rather than performing the calculations of model-order reduction on real-valued scalars, Harkness and Lopresti [18] instead used classical *intervals* on the real line to approximate each statistical variation as a *range*. The idea is appealing but has yet to be shown to be practical. Classical interval calculations are notoriously pessimistic, and attempts to date have been restricted to simplistic Elmore-style models that have a small closed form.

In this paper, we revisit the idea of using intervals as the principal mechanisms for representing uncertainties in numerical algorithms. The reasons for this are threefold. First, we show how advances in the handling of *correlated* interval computations [20] provide a key piece of the puzzle missing in prior attempts and greatly reduce the errors seen in arbitrary numerical calculations. Second, and more generally, we believe that intervals offer a *flexible* mechanism for creating workable robust versions of algorithms from successful nominal versions. Despite the impressive successes of the ideas in, for example, [5] and [21]–[23], what shall we do when none of these elegant techniques fits the problem we seek to solve in a new statistical form? Interval ideas originally evolved to solve a similar problem: arbitrary computations on *uncertain* quantities. The central idea is to replace, operator by operator, each simple numerical variable (hereafter referred to as a *scalar* for clarity) in the numerical recipe at hand with an interval-valued variable. Said differently, interval computations are *retrofit* on top of the existing numerical recipe; in a modern object-oriented programming language, we can simply *overload* the operators [24] with suitable interval-valued versions. Of course, intervals represent ranges, not statistics. Thus, our third motivation is that it is relatively straightforward to add a simple approximate statistical interpretation to the correlated interval form we employ, allowing us to build interval-valued statistical models.

To demonstrate this idea, we show in this paper how to replace the entire numerical recipes for asymptotic waveform evaluation (AWE)- and PRIMA-style model-order reductions with interval calculations and show what changes are needed in the algorithms to make this feasible. (Note that this is in contrast with the *interval-valued* models of [18], which work only for the low-order Elmore formulas.) Borrowing an idea from [12], we also use sampling as an intrinsic part of our reduction method. However, in contrast to [12], our methods produce interval-valued reduced-order transfer functions or eigensystems—interval poles and residues with an explicit correlation structure. Sampling these compactly represented intervals produces a set of scalar-valued poles and residues, each of which is approximately one sample of the variational behavior of the overall interconnect. The virtue of the approach is that the final low-order model has significantly fewer variational terms than the original large interconnect and can be quickly sampled.

Of course, the technique is not without its own new set of problems. Misestimation of interval endpoints and correlations creates a new source of errors, rather like floating-point roundoff errors but more macroscopic. Stability and passivity are similarly problematic since essentially every calculation is now interval valued. Pathological combinations of scalar parameters may be easily masked inside interval representations, which "usually"—but not always—give useful solutions. Approximating a statistical distribution with a range necessarily results in some loss of accuracy and requires some assumptions and approximations. In addition, comparing interval-valued quantities for essential decision steps in algorithms creates a wholly new difficulty. We describe these problems, and some empirical solutions, in the sequel.

This paper extends on the interval-valued computational ideas first presented in [25] and is organized as follows. Section II reviews the background on the new interval model and arithmetic. Section III details our interval-valued AWE and PRIMA algorithms. Section IV shows experimental results for interconnect delay, comparing Monte Carlo estimates with those from interval-valued models. Finally, Section V offers concluding remarks.

II. Background

A. Basics of Affine Intervals and Arithmetic

Classical interval arithmetic was invented in the 1960s by Moore [26] to solve range estimation problems in the presence of uncertainties. It has been the subject of extensive investigation, ranging even to complex problems such as linear and nonlinear equations expressed and solved over interval-valued variables [27]. In classical interval analysis, the uncertainty of a variable x is represented by an interval $\overline{x} = [\overline{x}.lo, \overline{x}.hi]$. The true value of x is only known to satisfy $\overline{x}.lo \leqslant x \leqslant \overline{x}.hi$. Basic arithmetic is redefined to yield interval solutions from interval operands, e.g., for addition, i.e.,

$$\overline{z} = \overline{x} + \overline{y} = [\overline{x}.lo + \overline{y}.lo,\ \overline{x}.hi + \overline{y}.hi]. \tag{1}$$

However, due to the lack of information about operand dependences, a serious problem is *overestimation*. To illustrate this, suppose $\overline{x} = [-1, 1]$, $\overline{y} = [-1, 1]$, and that x and y have the relationship as $y = -x$. If we compute $\overline{z} = \overline{x} + \overline{y}$, we can only obtain $\overline{z} = [-2, 2]$, whereas in reality, $z = x + y = 0$. This is a classical example of *range explosion*, when interval computations are pushed through long chains of calculations.

This situation was not improved until a novel range arithmetic model—*affine arithmetic*—was proposed in [20]. In

contrast to simple interval models, the affine arithmetic model preserves correlations among variables, in a form analogous to a first-order Taylor series. In this model, the uncertainty of a variable x is represented as a range in an affine form $\widehat{x}$, which is given by

$$\widehat{x} = x_0 + x_1\varepsilon_1 + x_2\varepsilon_2 + \cdots + x_n\varepsilon_n \quad (-1 \leqslant \varepsilon_i \leqslant 1; \quad i = 1, 2, \ldots, n). \quad (2)$$

Each uncertainty symbol ε_i stands for an independent component of the total uncertainties of the variable x; the corresponding coefficient x_i gives the magnitude of that component. We note immediately that, in contrast to traditional interval methods defined by their endpoints, affine intervals are defined by their central point x_0 and a set of symmetric excursions about this point. Still taking addition as an example, if $\widehat{x} = x_0 + x_1\varepsilon_1 + x_2\varepsilon_2$ and $\widehat{y} = y_0 + y_1\varepsilon_1 + y_3\varepsilon_3$

$$\widehat{z} = \widehat{x} + \widehat{y} = (x_0 + y_0) + (x_1 + y_1)\varepsilon_1 + x_2\varepsilon_2 + y_3\varepsilon_3 \quad (3)$$

which is again in an affine form. More importantly, we can see that one symbol, i.e., ε_i, may contribute to the uncertainties of two or more variables, indicating dependence among them. When these variables are combined, uncertainty terms may actually be cancelled.

Returning to the previous example, suppose that x and y have affine forms $\widehat{x} = 0 + 1\varepsilon$ and $\widehat{y} = -\widehat{x} = 0 - 1\varepsilon$. In this case, the affine form of the sum $\widehat{z} = \widehat{x} + \widehat{y} = 0$ perfectly coincides with the actual range of the variable z. This is the unique feature of the model and its central advantage over earlier interval methods.

Of course, the real problem is when an operation does *not* yield directly an affine result. For example, multiplication of affine intervals $\widehat{x}$ and $\widehat{y}$ gives the product

$$\begin{aligned}\widehat{z} &= \widehat{x}\widehat{y} \\ &= \left(x_0 + \sum_{i=1}^{n} x_i\varepsilon_i\right)\left(y_0 + \sum_{i=1}^{n} y_i\varepsilon_i\right) \\ &= x_0y_0 + \sum_{i=1}^{n}(y_0x_i + x_0y_i)\varepsilon_i + \left(\sum_{i=1}^{n} x_i\varepsilon_i\right)\left(\sum_{i=1}^{n} y_i\varepsilon_i\right)\end{aligned} \quad (4)$$

which is *not* in affine form any more, due to the quadratic uncertainty terms $(\sum_{i=1}^{n} x_i\varepsilon_i)(\sum_{i=1}^{n} y_i\varepsilon_i)$. According to [20], we can approximate the quadratic uncertainty terms by

$$\left(\sum_{i=1}^{n} x_i\varepsilon_i\right)\left(\sum_{i=1}^{n} y_i\varepsilon_i\right) \approx \left[R\left(\sum_{i=1}^{n} x_i\varepsilon_i\right)\right]\left[R\left(\sum_{i=1}^{n} y_i\varepsilon_i\right)\right]\zeta \quad (5)$$

where ζ is a new uncertainty symbol that is also in $[-1, 1]$ but distinct from all the other uncertainty symbols $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ that have already appeared in the same computation. R is the "radius operator," which is defined as

$$R\left(\sum_{i=1}^{n} x_i\varepsilon_i\right) = \sum_{i=1}^{n} |x_i| \quad (6)$$

which computes the upper error bound of an affine variable. Note that the substitution of $[R(\sum_{i=1}^{n} x_i\varepsilon_i)][R(\sum_{i=1}^{n} y_i\varepsilon_i)]\zeta$ for the quadratic terms implies a loss of information because ζ is assumed to be independent from $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$, while it is in fact a function of them. The result is in affine form again, i.e.,

$$\begin{aligned}\widehat{z} &\approx x_0y_0 + \sum_{i=1}^{n}(y_0x_i + x_0y_i)\varepsilon_i \\ &\quad + \left[R\left(\sum_{i=1}^{n} x_i\varepsilon_i\right)\right]\left[R\left(\sum_{i=1}^{n} y_i\varepsilon_i\right)\right]\zeta \\ &= z_0 + \sum_{i=1}^{n} z_i\varepsilon_i + R(\widehat{x}\widehat{y})\zeta\end{aligned} \quad (7)$$

which is a conservative approximation in the sense that it always bounds the original true affine product given by (4).

Stolfi and de Figueiredo [20] develop workable affine approximations for not only multiplication but also reciprocal, division, square root, $\exp(\cdot)$, $\log(\cdot)$, etc.[1] The basic idea for handling nonaffine intermediate forms (such as for our multiplication example) is to apply Chebyshev approximation or midrange approximation theory [20]; roughly speaking, we select an appropriate set of secants of the nominal function curve and manipulate these to parameterize an affine form to minimize the maximum absolute error of the approximation. The twin goals of this fundamental work were: 1) to create a practical set of interval arithmetic operators with which to replace their scalar counterparts in standard numerical codes and 2) to guarantee conservative bounding for the range uncertainty represented by each affine interval computation. In our work, we use goal 1, but we need abandon goal 2 to move from intervals to statistics.

B. From Intervals to Statistics

The essential assumption we make to move from intervals to a practical statistical interpretation is that the mechanics of range calculation for each finite interval are a reasonable approximation of how statistics actually move through the same computations. We cannot afford the expense necessary to calculate the exact statistics resulting from various arithmetic operations. We can, however, perform such operations fairly efficiently with affine interval representations. The affine interval model nicely handles first-order linear correlations among variables, which are captured via shared uncertainty symbols. In their traditional development, these uncertainty symbols ε_i's are independent and arbitrarily distributed within $[-1, 1]$, thus creating a set of (random) excursions about each interval's midpoint. However, what is not present in any of the traditional development of affine arithmetic is any *specific* distribution assumption for these uncertainty terms.

[1] It is assumed that for reciprocal and division, the denominator interval does not include zero, and for square root and $\log(\cdot)$, the operand interval is strictly positive.

This may seem odd, particularly in comparison to recent work on statistical static timing, in which the normal delay distribution assumption is *central* to the computational plan. However, remember that affine intervals are, first and foremost, *intervals*. Emphasis is on a conservative approximation to the *range* of the variational results we seek. Any specific probabilistic assumptions we add must be layered on top of this basic model (e.g., as in [28]); they are not part of the core computations as intervals move through each step of a long chain of numerical calculations.

Obviously, the easiest statistical interpretation for each ε_i is as a *uniformly* distributed value within $[-1, 1]$. This is simple, but it is unfortunately not very realistic for modeling manufacturing variations, which are continuous in nature. On the other hand, as we observed in [28], in the presence of more than just a few uncertainty terms in (2), the central limit theorem [29] suggests that the resulting distribution is asymptotically normal, independent of the distributions of individual uncertainty symbols. Hence, we choose the next most obvious interpretation, namely that each uncertainty symbol ε_i is a *normal* random variable, with $\mu = 0$ and $\sigma = 1$.

Doing so has several implications. First, if a numerical algorithm has input variables of the affine form $\widehat{x} = x_0 + \sum_{i=1}^{m} x_i\varepsilon_i$, we regard this as a *symbolic* form composed of independent zero-mean unit-variance normal random variables. Second, if the numerical algorithm produces outputs of the affine form $\widehat{z} = z_0 + \sum_{i=1}^{n} z_i\varepsilon_i$ after a sequence of affine arithmetic operations, we still assume—an approximation of course—that this is a symbolic form combining a (possibly larger) set of independent normal random variables that still have zero mean and unit variance. Finally, unlike the uniform case, this now means that while our calculations are done on "conservative" ranges, we evaluate any final interval-valued formula by sampling it with values that may actually extend *outside* each of these ranges. In other words, we have abandoned the idea that any range uncertainty has finite support but adopt a heuristic interpretation that the uncertainty terms each model an empirically useful notion of the "essential bulk" of an infinite continuous distribution between $\pm\sigma$. Again, this assumes that the careful mechanics of bound estimation, which is computed for each affine interval operation, can serve as a workable surrogate for how the statistics would actually move through the same computations, i.e., how the essential bulk of the distribution moves through these computations. This is a simple but surprisingly workable approximation as long as a few conditions are satisfied, as we shall see in Section IV.

It is also worthwhile to point out that this is *not* the only interpretation one can choose. In this paper, we use a statistical interpretation that emphasizes speed over accuracy. It is possible to alter the fundamental calculation and interpretation of the uncertainty symbols to produce much more accurate distribution estimates—at the cost of additional central processing unit time. This is not the focus of this paper (see [30] for details).

C. From Intervals to Algorithms

Having defined basic operations for correlated intervals, together with a very simple probabilistic interpretation, we can do some remarkably complex interval-valued numerical operations. A good example is interval-valued polynomial root finding. Suppose we have a polynomial over variable λ, with affine interval-valued coefficients $\widehat{c_k}$ $(k = 0, 1, 2, \ldots, q)$, i.e.,

$$\widehat{p}(\lambda) = \widehat{c_0} + \widehat{c_1}\lambda + \cdots + \widehat{c_q}\lambda^q. \tag{8}$$

We want to extract affine interval-valued roots. Several methods are available for the traditional scalar case, but following advice in [31] and [32], we might choose Laguerre's method, which requires no explicit derivatives and can be accomplished with a series of only basic interval arithmetic operations and evaluations of interval absolute values. This method has favorable convergence properties for the scalar case and is notably robust. It is guaranteed to converge monotonically to a scalar root from an arbitrary starting point. We can use the scalar version as a simple template for an interval-valued version.

for $j = 1, 2, \cdots, q$
  $\widehat{\alpha} = \widehat{c_q}$, $\widehat{\beta} = 0$, $\widehat{\gamma} = 0$;
  for $k = q-1, q-2, \cdots, 0$
    $\widehat{\gamma} = \widehat{z_0}\widehat{\gamma} + \widehat{\beta}$;
    $\widehat{\beta} = \widehat{z_0}\widehat{\beta} + \widehat{\alpha}$;
    $\widehat{\alpha} = \widehat{z_0}\widehat{\alpha} + \widehat{c_k}$;
  end
  $\widehat{A} = -\widehat{\beta}/\widehat{\alpha}$;
  $\widehat{B} = \widehat{A}^2 - 2\widehat{\gamma}/\widehat{\alpha}$;
  $\widehat{C} = [\widehat{A} \pm \sqrt{(q-1)(q\widehat{B} - \widehat{A}^2)}]/q$;
  $\widehat{z}_{new} = \widehat{z_0} + 1/\widehat{C}$;
  output j, $\widehat{z}_{new}$;
  if $|\widehat{z}_{new} - \widehat{z_0}| < e$ **then stop**;
  $\widehat{z_0} = \widehat{z}_{new}$;
end

Fig. 1. Interval-valued Laguerre's root-finding method.

In Fig. 1, variables like $\widehat{c_q}$ (with a $\widehat{\ }$) are intervals; variables like q are scalars. $\widehat{z_0}$ is an initial approximate root; $\widehat{z}_{\rm new}$ is an interval-valued root found in each iteration; e is a prespecified error tolerance. Fig. 1 shows the basic iteration and allows us to highlight another crucial point. Replacing scalar arithmetic operations and function approximations like $\sqrt{(\cdot)}$ with affine intervals is necessary but not sufficient for most algorithms. We also must deal with *decision* steps, which compare values. This presents no problems for the scalar case: Either x is greater than y, or not, and so forth. However, such questions are not so well posed for interval-valued operands $\widehat{x}$ and $\widehat{y}$, where specific choices for a scalar element of the range may throw the decision one way or the other. To confront this, we make a simple standard choice as in [33]: *We always use the central value of the interval for all decision steps*. Thus, if we need to compare $\widehat{x} = x_0 + \sum_{i=1}^{n} x_i\varepsilon_i$ against $\widehat{y} = y_0 + \sum_{i=1}^{n} y_i\varepsilon_i$, we do this by simply comparing scalar x_0 against scalar y_0. This is always well defined and efficient and has another desirable feature. As noted in [12], it is highly desirable that, when statistical variations are set to zero, a variational analysis algorithm should be able to solve for the nominal case, i.e., to produce an answer that is *identical* to the deterministic solution for the nominal case. By using central values for all interval-valued decisions, we guarantee this useful property.

III. Interval-Valued Interconnect Modeling and Order Reduction

In this section, we develop interval-valued versions of the standard AWE [7] and PRIMA [9] model-order reduction algorithms. The overall strategy is easily summarized: We represent variational interconnect circuit element values as affine intervals and then push these interval values through the numerical linear and nonlinear solution steps that comprise each algorithm. The outcome—which is arrived at through different paths in these two algorithms—is a set of interval-valued poles and residues. Our expectation is that the large number of parameters in the original unreduced interconnect circuit will be replaced by a small number of interval-valued coefficients in the final pole/residue form. We statistically sample these final "reduced" intervals to produce an estimate of the delay distribution for the original variational interconnect.

A. Modeling Resistance–Inductance–Capacitance (RLC) Parameter Variations

Manufacturing process variations are random in nature, and the true causes are complicated. In general, the variations can be classified into two categories: *global* variation and *local* variation, as in [1] and [12]. Global variations, such as critical dimension variations, are *interdie* and can be assumed to affect all the devices and interconnects in a similar way within the same chip. Local variations, such as metal width, metal thickness, and interlayer dielectric variations, are *intradie* and often exhibit spatial correlations, i.e., the device and interconnect parameters are affected similarly by a common source of variation when these physical elements are close enough to each other. Global variations used to dominate local variations. As semiconductor technology scales and die size grows rapidly, however, local variations are becoming as important as global variations [1].

In this paper, consider a linear interconnect circuit of general topology where the input driver voltage source is assumed to be scalar and deterministic, and the resistance R_i, capacitance C_i, and inductance L_i are subject to linear combinations of global and local variations with correlations in the basic affine forms introduced earlier. Thus

$$R_i = R_{i,0} + \sum_{j=1}^{l} \Delta R_{i,j}\varepsilon_j + \sum_{j=l+1}^{m} \Delta R_{i,j}\varepsilon_j + \sum_{j=m+1}^{n} \Delta R_{i,j}\varepsilon_j \tag{9}$$

$$C_i = C_{i,0} + \sum_{j=1}^{l} \Delta C_{i,j}\varepsilon_j + \sum_{j=l+1}^{m} \Delta C_{i,j}\varepsilon_j - \sum_{j=m+1}^{n} \Delta C_{i,j}\varepsilon_j \tag{10}$$

$$L_i = L_{i,0} + \sum_{j=1}^{l} \Delta L_{i,j}\varepsilon_j - \sum_{j=l+1}^{m} \Delta L_{i,j}\varepsilon_j - \sum_{j=m+1}^{n} \Delta L_{i,j}\varepsilon_j \tag{11}$$

where $R_{i,0}$, $C_{i,0}$, and $L_{i,0}$ are the nominal parameter values. Each unique source of global or local variation is modeled by an uncertainty symbol ε_j. $\Delta R_{i,j}$, $\Delta C_{i,j}$, and $\Delta L_{i,j}$ are the magnitude of the linearized parameter variations due to ε_j, which is a particular source of variation [34]. Any individual source of uncertainty may contribute to more than one parameter and thus lead to direct correlations among these parameters. Equations (9)–(11) simply emphasize the fact that any uncertainty symbol ε_j can appear with positive or negative proportional impact in any of the linearized formulas for any R, C, and L. For example, when metal width increases from its nominal value, the metal ground capacitance may be *increased* while the metal resistance is *decreased*. This is a simple model, but it can capture global and local variations and correlations, and it maps perfectly onto our preferred affine interval model of computation.

B. Interval-Valued AWE

Before beginning, it is worth commenting on why we include any discussion of AWE when more stable approaches such as PRIMA already exist. The answer in our case is that the AWE "recipe" was numerically simpler and, thus, was an easier first target in our development of this approach. Most of the computations in AWE are standard matrix–vector operations, which are easily handled by basic affine interval arithmetic. The new challenges in AWE are the need for interval-valued LU decomposition and polynomial root finding. LU decomposition itself is nothing more than a set of matrix–vector loops with a few critical decision steps, e.g., for pivoting. As we described earlier, all such decision steps on interval quantities simply compare the central values, thus guaranteeing that the central point of the outcome at least matches the nominal version of the algorithm. The same is true for the Laguerre-based root finder we described in Section II-C.

Let us also introduce some consistent terminologies. We shall again reserve the "$\widehat{\ }$" symbol for interval-valued quantities. The interval-valued modified nodal analysis (MNA) formulation in the frequency domain is

$$(\widehat{\boldsymbol{G}} + s\widehat{\boldsymbol{C}})\widehat{\boldsymbol{x}} = \boldsymbol{b} \tag{12}$$

where the interval-valued conductance matrix $\widehat{\boldsymbol{G}}$, the interval-valued susceptance matrix $\widehat{\boldsymbol{C}}$, and the scalar-valued excitation vector $\boldsymbol{b}$ are defined as

$$\widehat{\boldsymbol{G}} = \begin{bmatrix} \widehat{\boldsymbol{N}} & \boldsymbol{E} \\ -\boldsymbol{E}^T & \boldsymbol{0} \end{bmatrix} \quad \widehat{\boldsymbol{C}} = \begin{bmatrix} \widehat{\boldsymbol{Q}} & \boldsymbol{0} \\ \boldsymbol{0} & \widehat{\boldsymbol{H}} \end{bmatrix} \quad \boldsymbol{b} = -\begin{bmatrix} \boldsymbol{i} \\ \boldsymbol{v} \end{bmatrix}. \tag{13}$$

$\widehat{\boldsymbol{N}}$, $\widehat{\boldsymbol{Q}}$, and $\widehat{\boldsymbol{H}}$ are the interval-valued matrices containing the stamps for variational resistors, capacitors, and inductors, respectively. $\boldsymbol{E}$ consists of constants ones, negative ones, and zeros, which correspond to the variables of current induced by inductance and voltage source. The interval-valued solution vector $\widehat{\boldsymbol{x}}$ contains node voltages appended by inductance and voltage source current. The output-selecting vector is denoted as $\boldsymbol{l}$.

Analogous to AWE, interval-valued AWE assumes a moment expansion for $\widehat{\boldsymbol{x}}$ and matches its first $2q$ interval-valued moments by *Padé approximation*, where q is the order of model reduction. The $2q$ interval-valued moments are used to construct a qth-order interval-valued transfer function and

1. solve $\widehat{\boldsymbol{G}}\widehat{\boldsymbol{x}}_0 = \boldsymbol{b}$ for $\widehat{\boldsymbol{x}}_0$;
2. **for** $k = 1, 2, \cdots, 2q\text{–}1+s$
   solve $\widehat{\boldsymbol{G}}\widehat{\boldsymbol{x}}_k = -\widehat{\boldsymbol{C}}\widehat{\boldsymbol{x}}_{k-1}$ for $\widehat{\boldsymbol{x}}_k$
   end
3. obtain moments at output node of interest;
4. set up Hankel matrix and vector;
5. coefficients of transfer function's denominator = – (Hankel matrix)$^{-1}$·(Hankel vector);
6. find roots of denominator polynomial as poles;
7. set up Vandemonde matrix;
8. residues = – (Vandemonde matrix)$^{-1}$·(moments);

Fig. 2. Interval-valued AWE algorithm.

compute the corresponding interval-valued poles and residues. Fig. 2 gives a detailed description of the interval-valued AWE algorithm.

In Fig. 2, the interval-valued matrix $\widehat{\boldsymbol{G}}$ is first LU factorized using a standard Markowitz pivoting scheme, such that the subsequent matrix solves can be conveniently performed by only changing the right-hand-side (RHS) vectors and then one forward and backward substitution. LU decomposition is also used to find the inverse of an interval-valued matrix column by column, each time given the RHS vector as the column vector of identity matrix. Steps 1, 2, 5, and 8 use the interval-valued LU solve; step 6 uses the interval-valued root finding from [31] and [32], whose algorithm is shown in Fig. 1. As in the scalar case, we use *moment shifting*, which only adds s iterative loops to step 2, where s is the order of moment shifting. We also adopt *frequency shifting*, which scales interval-valued poles and residues by a scalar constant to reduce the instability of our results.[2]

C. Interval-Valued PRIMA

Due to precision loss that occurs during moment calculations and the inherent instability of Padé approximation, AWE may produce inaccurate results or fictitious unstable poles, although the original interconnect circuit is actually stable. PRIMA is a projection-based algorithm that is able to generate provably passive/stable reduced-order models in the scalar case. Again, we use a standard version of the PRIMA "recipe" and replace it with interval-valued computations. The MNA formulation is the same as that in Section III-B.

Interval-valued PRIMA constructs an orthonormal interval-valued matrix $\widehat{\boldsymbol{X}}$ that spans the interval-valued Krylov subspace defined as

$$Kr(\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{r}}, q) = colsp(\widehat{\boldsymbol{r}}, \widehat{\boldsymbol{A}}\widehat{\boldsymbol{r}}, \widehat{\boldsymbol{A}}^2\widehat{\boldsymbol{r}}, \ldots, \widehat{\boldsymbol{A}}^{q-1}\widehat{\boldsymbol{r}}) \tag{14}$$

where $\widehat{\boldsymbol{A}} = -\widehat{\boldsymbol{G}}^{-1}\widehat{\boldsymbol{C}}$ and $\widehat{\boldsymbol{r}} = -\widehat{\boldsymbol{G}}^{-1}\widehat{\boldsymbol{b}}$ by successively generating q interval-valued moment vectors $\widehat{\boldsymbol{x}}_k$ $(k = 0, 1, \ldots, q-1)$, like interval-valued AWE, and filling in the columns of $\widehat{\boldsymbol{X}}$ while maintaining its orthonormality (with respect to its central values). As a result, the central points of interval quantities in the basis of $\widehat{\boldsymbol{X}}$ are numerically better conditioned than the central points of the interval-valued moment space in interval-valued AWE. Interval-valued PRIMA then projects the original variational interconnect system into a reduced-order interval-valued system in the interval-valued Krylov subspace spanned by $\widehat{\boldsymbol{X}}$ via congruence transformation [8]. Finally, the interval-valued poles and residues of the reduced-order model are computed through eigendecomposition. Fig. 3 gives the details of the interval-valued PRIMA algorithm.

[2]For clarity, frequency scaling is omitted in Fig. 2.

1. solve $\widehat{\boldsymbol{G}}\widehat{\boldsymbol{x}}_0 = \boldsymbol{b}$ for $\widehat{\boldsymbol{x}}_0$;
2. $\widehat{\boldsymbol{X}}^{(0)} = \widehat{\boldsymbol{x}}_0$ and orthonormalize $\widehat{\boldsymbol{X}}^{(0)}$;
3. **for** $k = 1, 2, \cdots, q\text{–}1$
   solve $\widehat{\boldsymbol{G}}\widehat{\boldsymbol{x}}_k = -\widehat{\boldsymbol{C}}\widehat{\boldsymbol{x}}_{k-1}$ for $\widehat{\boldsymbol{x}}_k$;
   append $\widehat{\boldsymbol{x}}_k$ to $\widehat{\boldsymbol{X}}^{(k-1)}$;
   orthonormalize $\widehat{\boldsymbol{X}}^{(k)}$;
   end
4. compute $\widehat{\boldsymbol{G}}' = \widehat{\boldsymbol{X}}^T\widehat{\boldsymbol{G}}\widehat{\boldsymbol{X}}$ and $\widehat{\boldsymbol{C}}' = \widehat{\boldsymbol{X}}^T\widehat{\boldsymbol{C}}\widehat{\boldsymbol{X}}$; compute $\widehat{\boldsymbol{b}}' = \widehat{\boldsymbol{X}}^T\widehat{\boldsymbol{b}}$ and $\widehat{\boldsymbol{l}}' = \widehat{\boldsymbol{X}}^T\boldsymbol{l}$;
5. $\widehat{\boldsymbol{A}}' = -\widehat{\boldsymbol{G}}'^{-1}\widehat{\boldsymbol{C}}'$ and eigendecompose $\widehat{\boldsymbol{A}}'$;
6. let column vectors of $\widehat{\boldsymbol{S}}$ be eigenvectors; $\widehat{\boldsymbol{\Lambda}} = diag(\widehat{\lambda}_1, \widehat{\lambda}_2, \cdots, \widehat{\lambda}_q)$;
7. invert $\widehat{\lambda}_1, \widehat{\lambda}_2, \cdots, \widehat{\lambda}_q$ to obtain poles;
8. solve $\widehat{\boldsymbol{G}}'\widehat{\boldsymbol{w}} = \widehat{\boldsymbol{b}}'$ for $\widehat{\boldsymbol{w}}$;
9. residues = $-\widehat{\boldsymbol{S}}^T\widehat{\boldsymbol{l}}'\widehat{\boldsymbol{S}}^{-1}\widehat{\boldsymbol{w}}$·(poles);

Fig. 3. Interval-valued PRIMA algorithm.

Interval-valued PRIMA again makes heavy use of standard interval-valued matrix–vector operations, orthonormalization, and LU decomposition (see steps 1–5 and 7–9 in Fig. 3). We implement an interval-valued version of modified Gram–Schmidt orthonormalization procedure to orthonormalize $\widehat{\boldsymbol{x}}_k$ to all interval-valued vectors previously filled into $\widehat{\boldsymbol{X}}$. However, the new challenge is the eigendecomposition in step 5. $\widehat{\boldsymbol{A}}'$ is an interval-valued upper Hessenberg matrix, and we implement an interval-valued QR algorithm with implicit shifts to compute its all interval-valued eigenvalues $(\widehat{\lambda}_1, \widehat{\lambda}_2, \ldots, \widehat{\lambda}_q)$ after balancing $\widehat{\boldsymbol{A}}'$ to improve its numerical condition. All the interval-valued eigenvectors are then found by interval-valued inverse iteration. Again, interval-valued eigendecomposition ends up in practice as a standard sequence of iterative matrix–vector operations, which are interspersed with some critical decision steps, based on central values. We use as templates the scalar versions of these algorithms from [31] and [35].

D. Sampling Interval-Valued Poles and Residues

Interval-valued AWE and PRIMA produce q pairs of interval-valued poles and residues in the following affine form:

$$\widehat{p_i} = p_{i,0} + \sum_{j=1}^{n} p_{i,j}\varepsilon_j + \sum_{l=1}^{m} p_{i,l}\zeta_l \qquad (i = 1, 2, \ldots, q-1) \tag{15}$$

$$\widehat{k_i} = k_{i,0} + \sum_{j=1}^{n} k_{i,j}\varepsilon_j + \sum_{l=1}^{m} k_{i,l}\zeta_l \qquad (i = 1, 2, \ldots, q-1) \tag{16}$$

where $p_{i,0}$ and $k_{i,0}$ are the same nominal poles and residues as those computed by deterministic AWE and/or PRIMA. ε_j's

$(j = 1, 2, \ldots, n)$ are uncertainty symbols originally associated with the resistance, capacitance, and inductance in (9)–(11). ζ_l's $(l = 1, 2, \ldots, m)$ are *new* uncertainty symbols created during affine approximations and assumed to be independent of ε_j's. At this point, we sample these interval-valued poles and residues, which are made up of weighted linear sums of both original and new uncertainty symbols with zero mean and unit variance. The result is a larger set of random samples of the distribution of the final *scalar-valued* poles and residues. Then for each set of samples, the scalar-valued interconnect circuit delay can be obtained via time-domain transient analysis. Enough random samples produce a delay distribution for the variational interconnect. Monte Carlo sampling over pole and residue intervals is very efficient since it primarily involves very simple scalar computations over the *reduced* set of quantities. We will also illustrate the effect of sampling over the interval-valued poles in Section IV.

Now, we can immediately see why questions of passivity and stability for interval-valued models are so difficult. In its original form, AWE did not guarantee either passive or stable reduction; PRIMA remedied this difficulty. This corresponding variational version of the problem for the matrix perturbation approach of Liu *et al.* [12] was resolved in [36] and [37] via a careful coupling of the reduced variational model with a fast custom transistor-level timing engine. Unfortunately, we cannot yet say anything similarly formal for our interval-valued results. The center points of our resulting intervals behave, by construction, as the nominal versions of each algorithm. However, the difficulty of the linearized approximations for higher order uncertainties, the conservative nature of range estimation, and the normal sampling procedure mean that any particular sample from these interval ranges may not yield a stable/passive result.

In practice, we use some simple heuristics to remove obviously erroneous results, e.g., unstable poles are simply omitted when they are sampled, as are pole values that differ in sign from their central value. The reason for this latter rule is more subtle. Affine intervals are, by construction, symmetric about their center points. Conservative bounds sometimes create cases where the overall interval range straddles both the positive and negative real lines, even when one or the other of these signs for the results is impossible (see [20] and [30] for a more careful treatment of the problem). This is another source of error for us in extracting useful interval-valued pole/residue pairs. We note that, while some theory for the stability of interval-valued transfer functions exists [33], it does not target the more useful correlated affine model. For simple classical intervals, even the decision problem for stability is sometimes unsolvable. Thus, the real question we wish to answer is, empirically, how well the affine intervals yield usable results.

IV. Experimental Results

The interval-valued AWE and PRIMA algorithms have been implemented as *int*AWE and *int*PRIMA, respectively. The pair of tools, together with the underlying affine arithmetic library, were implemented in C/C++ and benchmarked on a 1.0-GHz UNIX machine, using four $RC(L)$ interconnect circuits of general topology: design1, design2, design3, and design4. The numbers of RLC elements of the four circuits are given in Table I. We choose a number of uncertainty symbols ranging from 4 to 20 for (9)–(11). Among these symbols, one is assumed to originate from global variation and is shared by all RLC elements. The rest of the symbols are assumed for local wire-width variations, linearized by first-order Taylor series expansion, and only shared by a cluster of RLC elements that are "close enough" to one another. In other words, for the case of, e.g., four uncertainty terms, we partition each netlist into four groups and assign the same uncertainty term to every element in one group. Our algorithms and implementation can accommodate any number of global and local uncertainty symbols that originate from most types of variation sources. In addition, note that these are only the *initial* sets of uncertainty symbols, and more new distinct symbols will be created during the interval-valued model-order reduction process. Furthermore, we assume three combinations for the relative σ of global and local variations: 15%/5%, 5%/15%, and 5%/30%, which are given in the first columns of Tables II and III. We use eighth-order reduction for the RLC circuits design1 to design3 and fourth-order reduction for the RC circuit design4 and compute 50% delay distribution for the variational interconnect circuits.

TABLE I
Numbers of R, C, and L Elements for the Four Circuits

	R	C	L
design1	42	41	40
design2	186	185	185
design3	248	247	247
design4	638	637	0

We compare our proposed approaches with complete straightforward Monte Carlo simulations (denoted as MC-AWE and MC-PRIMA), i.e., randomly varying RLC circuit elements according to the specified variations at the very beginning and repeating scalar deterministic model reductions all the way through. To allow maximum trust in the various numerical steps in each algorithm, we implemented the scalar deterministic AWE and PRIMA algorithms using MATLAB. For simplicity, we use 10 000 samples, in both the complete straightforward MATLAB-based Monte Carlo simulations and the Monte Carlo sampling steps in our proposed approaches, for each combination of design case, global variation, local variation, and number of uncertainty terms. The overall run times, mean, and standard deviation (std) of the resulting delay distributions are measured and shown in Tables II and III. In columns (9)–(11) of Tables II and III, the run-time speedup and the errors of mean delay and std are defined as the difference between the result of our proposed approach and that of complete straightforward MATLAB-based Monte Carlo simulation, which is normalized to the latter.

It is easy to see that both the mean delay and std errors for all the experiments are less than 10%, even with variations up to 35%. On average, the mean delay error is 4.1% for *int*AWE and 4.9% for *int*PRIMA. Empirically, all the mean delay errors are positive; *int*AWE and *int*PRIMA are providing conservative approximations. On average, the std errors are 5.6% and 5.9% for *int*AWE and *int*PRIMA, respectively.

TABLE II
COMPARISON BETWEEN MC-AWE AND *int*AWE

1	2	3	4	5	6	7	8	9	10	11
global/local variation	number of ε	MC-AWE			*int*AWE			error		run time speed-up
		mean (ps)	std (ps)	run time (s)	mean (ps)	std (ps)	run time (s)	mean	std	
design1										
15% / 5%	4	271.4	37.8	418	275.8	38.8	23	1.6%	2.6%	17
	8	271.9	37.7	419	275.9	38.6	28	1.5%	2.4%	14
	16	271.7	37.3	419	275.2	38.1	36	1.3%	2.1%	12
5% / 15%	4	272.6	37.7	417	274.0	38.6	22	0.5%	2.7%	18
	8	272.6	37.6	418	273.7	38.5	29	0.4%	2.4%	13
	16	270.5	37.2	419	274.1	38.1	36	1.3%	3.5%	11
5% / 30%	4	270.4	37.9	419	274.7	38.9	22	1.6%	2.6%	18
	8	270.2	37.8	420	274.0	38.8	29	1.4%	2.9%	14
	16	269.3	37.8	422	273.8	38.5	39	1.7%	1.9%	11
design2										
15% / 5%	5	555.6	78.3	3884	588.9	83.9	171	6.0%	7.2%	22
	9	550.6	76.9	3912	575.4	83.2	225	4.5%	8.2%	16
	18	551.9	75.4	3927	579.2	81.6	321	4.9%	8.2%	11
5% / 15%	5	564.1	76.2	3859	582.6	81.0	170	3.3%	6.3%	22
	9	557.3	75.6	3905	584.5	78.8	236	4.9%	4.2%	16
	18	561.4	74.0	3918	576.7	78.4	314	2.7%	5.9%	11
5% / 30%	5	565.7	79.6	3962	587.8	85.6	159	3.9%	7.5%	25
	9	560.0	78.5	3997	579.4	83.7	231	3.3%	6.6%	16
	18	560.8	76.2	3984	581.0	80.5	329	3.6%	5.6%	11
design3										
15% / 5%	5	694.2	98.2	7432	723.2	106.5	305	4.2%	8.5%	23
	10	702.8	97.6	7459	723.5	104.9	392	2.9%	7.5%	18
	20	700.3	97.5	7454	723.9	103.3	509	3.4%	5.9%	14
5% / 15%	5	708.7	97.8	7447	735.3	103.6	303	4.6%	5.9%	24
	10	699.1	96.4	7478	726.7	102.9	388	3.9%	6.7%	18
	20	709.0	95.2	7480	729.1	100.4	519	2.8%	5.5%	13
5% / 30%	5	705.3	99.1	7475	734.0	107.1	302	4.1%	8.1%	24
	10	704.4	98.3	7451	733.2	106.0	381	4.1%	7.8%	19
	20	707.2	98.2	7467	733.8	103.9	508	3.8%	5.8%	14
design4										
15% / 5%	4	1319.4	185.7	14794	1431.0	197.6	512	8.5%	6.6%	28
	8	1320.3	184.2	14815	1424.6	196.5	650	7.9%	6.7%	22
	16	1319.4	182.6	14823	1428.5	194.0	824	8.3%	6.2%	17
5% / 15%	4	1324.0	184.0	14682	1437.7	196.4	507	8.6%	6.7%	28
	8	1314.0	183.5	14797	1435.4	194.9	645	9.2%	6.2%	22
	16	1320.8	182.1	14823	1436.2	192.1	820	8.7%	5.5%	17
5% / 30%	4	1322.6	186.1	14851	1413.2	198.5	506	6.9%	6.7%	28
	8	1322.9	184.8	14942	1425.4	197.3	643	7.7%	6.8%	22
	16	1327.9	183.6	14915	1421.8	195.3	829	7.1%	6.4%	17

*int*AWE achieves an average of 18× run-time speedup over MC-AWE, whereas *int*PRIMA achieves 19× over MC-PRIMA. In general, more speedup is observed for larger circuits. The run time of *int*AWE and *int*PRIMA increases with the number of uncertainty symbols, whereas the run time of the complete straightforward MATLAB-based Monte Carlo simulations almost remains the same because 10 000 vectors were used to sample any number of uncertainty symbols (ranging from 4 to 20).

Two observed trends are worth mentioning here. First, we do not observe more error for the test cases with more uncertainty terms. One explanation is that the affine intervals effectively capture the main correlations and allow for significant empirical cancellation among the uncertainty terms. Second, as expected, we see more error as the size of the interconnect circuit increases. Just as with basic floating-point calculations, a longer chain of calculations produces a less accurate result. The same is true for all interval-based approaches. An important component of ongoing work is trying to pin down more carefully where the most serious sources of misestimation occur and what we might do about them. Nevertheless, we regard the results of Tables II and III as a very satisfactory outcome for the first *fully* interval-valued implementation of modern model-order reduction methods.

Figs. 4 and 5 pictorially demonstrate the accuracy of our interval-valued approaches. First, recall that the affine interval model is fundamentally a *linear* model of correlated range uncertainties, defined by a central point and a set of symmetric excursions about that point. Higher order terms are replaced with newly created uncertainty symbols that capture (in a conservative sense) the linear range implications but sever the perfect connection to the problem's original, possibly nonlinear, uncertainty properties. The soundness of these approximations can be examined more closely through, e.g., sampling the interval-valued poles obtained by interval-valued nonlinear root finding of *int*AWE. Fig. 4 plots the distributions of the four dominant poles of the RLC circuit design1 on the complex plane, assuming 5% global variation, 30% local variation, eight uncertainty terms, and eighth-order AWE reduction. False color shading indicates greater pole density with a darker color. We compare the pole distributions extracted by MATLAB-based Monte Carlo simulation [see Fig. 4(a)] with those by *int*AWE

TABLE III
COMPARISON BETWEEN MC-PRIMA AND *int*PRIMA

1	2	3	4	5	6	7	8	9	10	11
global/local variation	number of ε	MC-PRIMA			*int*PRIMA			error		run time speed-up
		mean (ps)	std (ps)	run time (s)	mean (ps)	std (ps)	run time (s)	mean	std	
design1										
15% / 5%	4	239.8	32.8	403	243.9	34.4	22	1.7%	4.9%	17
	8	240.6	32.4	405	243.7	34.3	28	1.3%	5.9%	14
	16	239.8	32.1	400	244.1	33.7	38	1.8%	5.0%	10
5% / 15%	4	240.2	32.4	411	244.9	33.9	22	2.0%	4.6%	18
	8	239.4	32.0	404	245.1	33.8	28	2.4%	5.6%	13
	16	240.5	30.9	416	244.9	32.4	39	1.8%	4.9%	10
5% / 30%	4	240.1	33.5	407	245.3	35.0	22	2.2%	4.5%	18
	8	242.0	33.1	410	245.4	35.0	29	1.4%	4.5%	13
	16	240.4	33.1	407	244.9	34.2	38	1.9%	3.3%	10
design2										
15% / 5%	5	561.4	78.0	3387	588.3	82.9	151	4.8%	6.3%	21
	9	554.3	77.5	3420	579.5	81.6	197	4.5%	5.3%	16
	18	560.8	76.2	3411	585.2	80.0	272	4.4%	5.0%	12
5% / 15%	5	569.0	77.6	3405	580.8	82.2	164	2.1%	5.9%	20
	9	562.8	76.9	3391	585.6	81.0	190	4.1%	6.2%	17
	18	563.1	76.0	3402	583.5	80.8	257	3.6%	6.3%	12
5% / 30%	5	556.5	78.3	3402	582.8	83.4	147	4.7%	6.5%	22
	9	554.7	78.0	3424	578.6	82.8	189	4.3%	6.2%	17
	18	554.6	77.5	3418	579.2	81.1	276	4.3%	4.6%	11
design3										
15% / 5%	5	691.5	97.6	6510	723.7	103.4	246	4.7%	5.9%	26
	10	700.2	97.1	6497	723.2	103.1	305	3.3%	6.2%	21
	20	699.4	96.0	6521	724.3	103.1	392	3.6%	7.3%	16
5% / 15%	5	696.9	96.3	6484	732.1	101.9	256	5.1%	5.8%	24
	10	704.3	96.0	6506	734.2	101.8	312	4.2%	6.0%	20
	20	696.2	94.8	6545	731.5	100.1	374	5.1%	6.6%	17
5% / 30%	5	697.7	98.0	6495	730.2	104.2	248	4.7%	6.3%	25
	10	698.1	96.9	6512	725.6	103.5	313	3.9%	6.8%	20
	20	696.0	96.4	6510	726.9	102.9	388	4.4%	6.7%	16
design4										
15% / 5%	4	1321.4	185.2	12538	1418.2	196.0	427	7.3%	5.8%	28
	8	1314.8	184.3	12607	1409.6	195.1	504	7.2%	5.4%	24
	16	1321.0	181.6	12663	1411.8	193.3	611	7.4%	6.4%	20
5% / 15%	4	1319.0	184.1	12475	1422.3	195.0	410	7.8%	5.9%	29
	8	1320.2	183.4	12513	1413.7	193.8	506	7.1%	5.7%	24
	16	1322.7	181.9	12745	1420.4	192.0	602	7.4%	5.6%	20
5% / 30%	4	1315.9	185.2	12600	1400.2	196.9	425	6.4%	6.3%	29
	8	1316.9	184.2	12794	1397.0	195.4	521	6.1%	6.1%	24
	16	1315.6	184.0	12682	1398.5	194.1	601	6.3%	5.5%	20

[see Fig. 4(b)]. The correspondence is good. To further quantify the correspondence, we first define the "distribution error" for a grid as the difference between the number of poles in the grid obtained by *int*AWE and that obtained by Monte Carlo simulation. Then, the average error is the square root of the sum of the squared distribution errors over all the grids, divided by the total number of samples. The average errors of pole distributions predicted by *int*AWE with respect to MC-AWE are 6.9%, 9.3%, 9.5%, and 7.4%, respectively, for the four poles (from top to bottom). Similar observations hold for PRIMA reduction. In other words, our first-order correlated affine interval model can produce scalar-valued pole samples that approximate real nonlinear pole distributions quite accurately in practice.

Fig. 5 plots the probability distribution function (pdf) and cumulative distribution function (cdf) for design4 with 5% global variation, 30% local variation, and 16 initial uncertainty symbols. In Fig. 5, the pdfs and cdfs given by our interval-valued approaches match very well with those given by straightforward Monte Carlo simulation, although on average, the estimation error for design4 is greater than that for other designs. For this test case, we also quantify the accuracy of our approaches in the following sense: For each Monte Carlo sample over ε's, the relative error is the difference between the delay result of our approach and that of straightforward Monte Carlo simulation, which is normalized to the latter. Then, the average error of the delay distribution predicted by our approach is defined as the square root of the sum of the squared relative errors over all the Monte Carlo samples, divided by the number of samples. The average error is 9.7% for *int*AWE [Fig. 5(a)] and 11.1% for *int*PRIMA [Fig. 5(c)]. Interestingly enough, the distribution of 50% delay is asymmetrical with a positive skew (mean $>$ median) and a long tail at the far end of distribution histogram [see Fig. 5(a) and (c)], although the parameter uncertainties do follow normal distributions.

Finally, it is worth addressing, at least partially, the more general question of where this particular set of heuristic approximations may be viable, i.e., where our simple statistical interpretation on the affine interval model is a workable surrogate for the real statistics. Fig. 6 gives a first characterization of the answer here. We know that the addition, subtraction, and scaling by scalar constant need no new uncertainty symbols and lose no accuracy even when we interpret the interval-valued

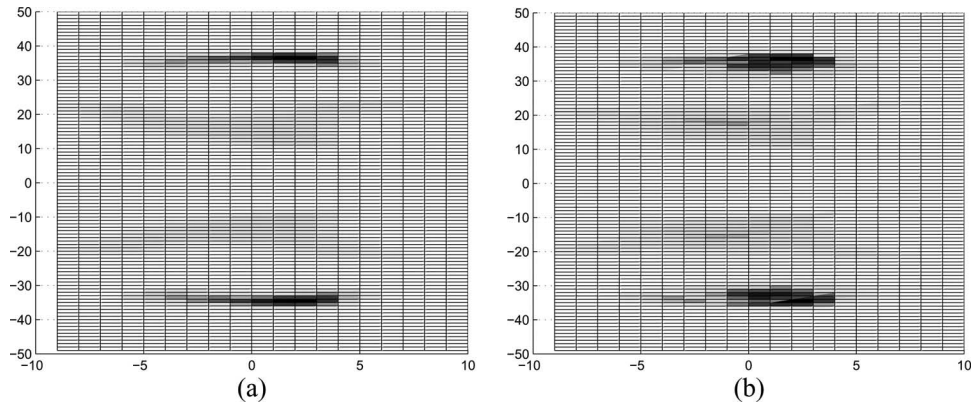


Fig. 4. Comparison of pole distribution on the complex plane between MC-AWE and *int*AWE for design1. For both figuras, the x-axis is the real axis and the y-axis is the imaginary axis. The average errors are 6.9%, 9.3%, 9.5%, and 7.4%, respectively, for the four poles (from top to bottom). (a) Pole distribution of MC-AWE. (b) Pole distribution of *int*AWE.

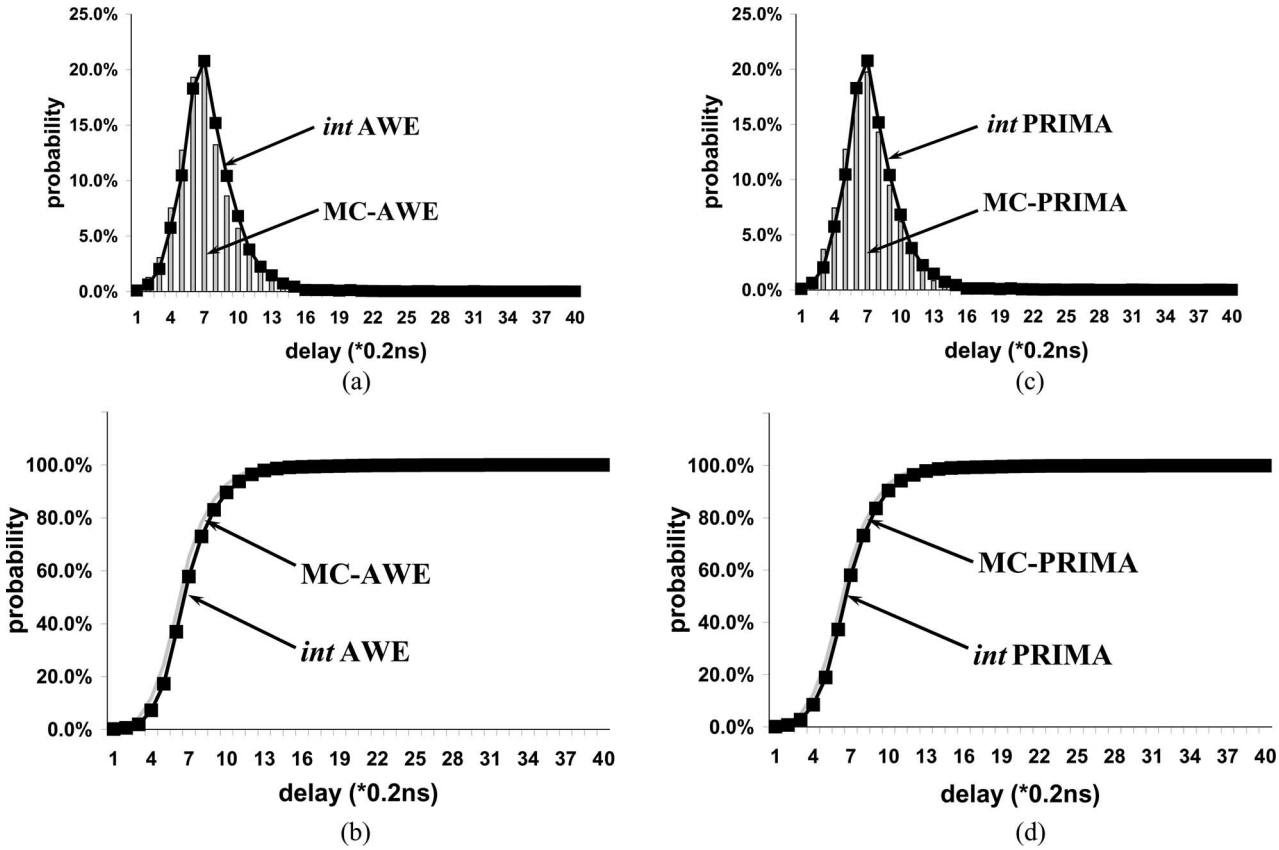


Fig. 5. Comparison of pdfs and cdfs for design4. (a) pdf: MC-AWE versus *int*AWE (average error: 9.5%). (b) cdf: MC-AWE versus *int*AWE. (c) pdf: MC-PRIMA versus *int*PRIMA (average error: 11.1%). (d) cdf: MC-PRIMA versus *int*PRIMA.

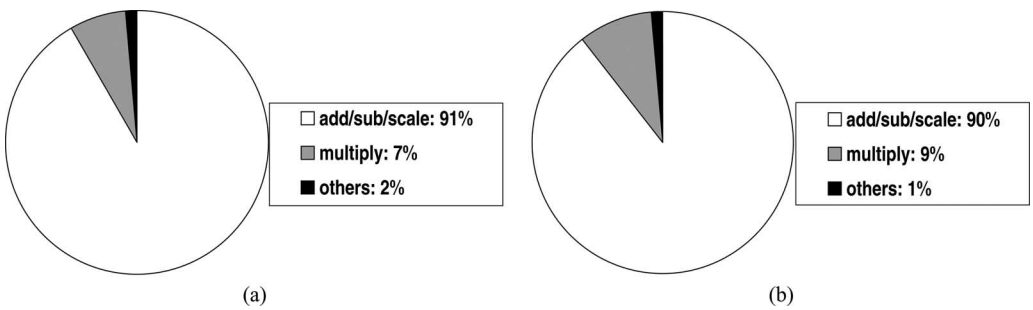


Fig. 6. Percentages of affine operators in (a) Interval-valued AWE and (b) Interval-valued PRIMA for design2.

operands as combinations of zero-mean unit-variance normal variables and not of finite $[-1,1]$ uncertainties. However, nonaffine operators, such as multiplication, division, square root, transcendentals, etc., introduce new uncertainty symbols and are not so accurate even for the standard affine interval range calculations, and thus, the statistical interpretation is also less accurate. It also becomes less efficient when managing a large number new uncertainty symbols. What saves us from a more macroscopic accuracy and efficiency loss is the fact that linear circuit analysis is dominated ($\geqslant 90\%$ for design2, as shown in Fig. 6) at the level of basic arithmetic operators, by addition, subtraction, and scaling by scalar constant. This is not particularly surprising since it has long been known that, for example, division is relatively rare in the mix of floating-point operations when measured across a wide range of numerical applications [38]. In our application, the operations that create new uncertainty symbols to manage and lose the most accuracy are relatively rare. As a result, the overall interval-valued version of the numerical computation offers a workable level of overall efficiency and accuracy.

V. Conclusion

The affine interval model, which allows us to represent and preserve first-order correlations among intervals, can be applied to classical model-order reduction techniques for linear interconnect, yielding a compact interval-valued circuit model. A simple statistical interpretation of the resulting intervals allows us to statistically sample the reduced model and estimate the distribution of delay for the original variational interconnect circuit. Preliminary results are extremely encouraging and suggest that interval-valued models may be able to bring statistics into a wider range of CAD tools and play a more central role in statistical design and analysis. Our ongoing work focuses on understanding more clearly the sources of interval error in our present approach, the best mapping from intervals to statistics, and the optimal "boundary" between where it is advantageous to compute with intervals, and where it is more efficient to stop, sample the intervals, and continue forward with a set of scalar-valued models (e.g., see [39] for follow-on work in this direction).

Acknowledgment

The authors would like to thank the following people from Carnegie Mellon University: C. Fang, who participated in the development of the affine arithmetic library and provided many useful insights into affine arithmetic; S. Tiwary, who provided MATLAB versions of AWE and PRIMA for the comparisons; and L. Pileggi, X. Li, and Y. Zhan for helpful discussions on model-order reduction methods and process variations.

References

[1] S. Nassif, "Modeling and analysis of manufacturing variations," in *Proc. Custom Integr. Circuits Conf.*, 2001, pp. 223–228.

[2] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," in *Proc. Int. Conf. Comput.-Aided Des.*, 2003, pp. 607–614.

[3] S. Bhardwaj, B. Vrudhula, and D. Blaauw, "τAU: Timing analysis under uncertainty," in *Proc. Int. Conf. Comput.-Aided Des.*, 2003, pp. 615–620.

[4] H. Chang and S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *Proc. Int. Conf. Comput.-Aided Des.*, 2003, pp. 621–625.

[5] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *Proc. Des. Autom. Conf.*, 2004, pp. 331–336.

[6] J. Le, X. Li, and L. T. Pileggi, "STAC: Statistical timing analysis with correlation," in *Proc. Des. Autom. Conf.*, 2004, pp. 343–348.

[7] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. 4, pp. 352–366, Apr. 1990.

[8] K. J. Kerns and A. T. Yang, "Stable and efficient reduction of large, multiport *RC* networks by pole analysis via congruence transformations," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 16, no. 7, pp. 734–744, Jul. 1997.

[9] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645–654, Aug. 1998.

[10] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas, "Impact of interconnect variations on the clock skew of a gigahertz microprocessor," in *Proc. Des. Autom. Conf.*, 2000, pp. 168–171.

[11] C. L. Ratzlaff and L. T. Pillage, "RICE: Rapid interconnect circuit evaluation using AWE," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 13, no. 6, pp. 763–776, Jun. 1994.

[12] Y. Liu, L. T. Pileggi, and A. J. Strojwas, "Model order-reduction of $RC(L)$ interconnect including variational analysis," in *Proc. Des. Autom. Conf.*, 1999, pp. 201–206.

[13] L. Daniel, O. C. Siong, L. S. Chay, K. H. Lee, and J. White, "Multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 5, pp. 678–693, May 2004.

[14] J. M. Wang, P. Ghanta, and S. Vrudhula, "Stochastic analysis of interconnect performance in the presence of process variations," in *Proc. Int. Conf. Comput.-Aided Des.*, 2004, pp. 880–886.

[15] J. R. Phillips, "Variational interconnect analysis via PMTBR," in *Proc. Int. Conf. Comput.-Aided Des.*, 2004, pp. 872–879.

[16] K. Agarwal, D. Sylvester, D. Blaauw, F. Liu, S. Nassif, and S. Vrudhula, "Variational delay metrics for interconnect timing analysis," in *Proc. Des. Autom. Conf.*, 2004, pp. 381–384.

[17] X. Li, J. Le, P. Gopalakrishman, and L. T. Pileggi, "Asymptotic probability extraction for non-normal distributions of circuit performance," in *Proc. Int. Conf. Comput.-Aided Des.*, 2004, pp. 2–9.

[18] C. L. Harkness and D. P. Lopresti, "Interval methods for modeling uncertainty in *RC* timing analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 11, no. 11, pp. 1388–1401, Nov. 1992.

[19] J. Phillips, L. Daniel, and L. M. Silveira, "Guaranteed passive balancing transformations for model order reduction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 8, pp. 1027–1041, Aug. 2003.

[20] J. Stolfi and L. H. de Figueiredo, "Self-validated numerical methods and applications," in *Brazilian Mathematics Colloquium Monograph*. Rio De Janeiro, Brazil: IMPA, 1997. [Online]. Available: http://www.ece.cmu.edu/~dazhuanm/doc/aabook.ps

[21] J. Singh, V. Nookala, Z.-Q. Luo, and S. Sapatnekar, "Robust gate sizing by geometric programming," in *Proc. Des. Autom. Conf.*, 2005, pp. 315–320.

[22] M. Mani, A. Devgan, and M. Orshansky, "An efficient algorithm for statistical minimization of total power under timing yield constraints," in *Proc. Des. Autom. Conf.*, 2005, pp. 309–314.

[23] D. Patil, S. Yun, S.-J. Kim, A. Cheung, M. Horowitz, and S. Boyd, "A new method for design of robust digital circuits," in *Proc. Int. Symp. Quality Electron. Des.*, 2005, pp. 676–681.

[24] B. Eckel, *Thinking in C++*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[25] J. D. Ma and R. A. Rutenbar, "Interval-valued reduced order statistical interconnect modeling," in *Proc. Int. Conf. Comput.-Aided Des.*, 2004, pp. 460–467.

[26] R. E. Moore, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

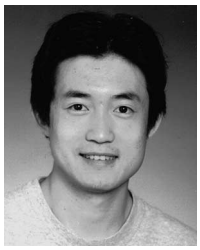
[27] A. Neumaier, *Interval Methods for Systems of Equations*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[28] C. F. Fang, R. A. Rutenbar, M. Püschel, and T. Chen, "Toward efficient static analysis of finite precision effects in DSP applications via affine arithmetic modeling," in *Proc. Des. Autom. Conf.*, 2003, pp. 496–501.

[29] M. R. Spiegel, *Theory and Problems of Probability and Statistics*. New York: McGraw-Hill, 1992.

[30] C. F. Fang, "Probabilistic interval-valued computation: Representing and reasoning about uncertainty in DSP and VLSI designs," Ph.D. dissertation, CMU, Pittsburgh, PA, 2005.

[31] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C*. Cambridge, U.K.: Cambridge Univ. Press, 1992.
[32] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*. Pacific Grove, CA: Brooks/Cole, 2002.
[33] L. V. Kolev, *Interval Methods for Circuit Analysis*. Singapore: World Scientific, 1993.
[34] V. Mehrotra, S. Nassif, D. Boning, and J. Chung, "Modeling the effects of manufacturing variation on high-speed microprocessor interconnect performance," in *IEDM Tech. Dig.*, 1998, pp. 767–770.
[35] G. H. Golub and C. F. Van Loan, *Matrix Computation*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
[36] E. Acar, L. T. Pileggi, and S. R. Nassif, "A linear-centric simulation framework for parametric fluctuations," in *Proc. Des., Autom. and Test Eur. Conf.*, 1999, pp. 568–575.
[37] E. Acar, S. R. Nassif, Y. Liu, and L. T. Pileggi, "Time-domain simulation of variational interconnect models," in *Proc. Int. Symp. Quality Electron. Des.*, 2002, pp. 419–424.
[38] S. F. Anderson, J. G. Earle, R. E. Goldschmidt, and D. M. Powers, "The IBM system/360 model 91: Floating-point execution unit," *IBM. J. Res. Develop*, vol. 11, no. 1, pp. 34–53, Jan. 1967.
[39] J. D. Ma and R. A. Rutenbar, "Fast interval-valued statistical interconnect modeling and reduction," in *Proc. Int. Symp. Phys. Des.*, 2005, pp. 159–166.



James D. Ma received the Ph.D. degree in computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2006.

He joined the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, in 2006. He was a summer intern with Synopsys in 2003 and an intern with Cadence Berkeley Laboratories from 2005 to 2006. His research interests are design-for-manufacturing, logic and physical design automation, and circuit simulation.

Dr. Ma is on the Technical Program Committees of ACM Great Lakes Symposium on VLSI and IEEE International Symposium on Circuits and Systems. In 2006, he was a recipient of the MARCO Inventor Recognition Award and the Chinese Government Award for Outstanding Ph.D.s Abroad.

Rob A. Rutenbar (S'77–M'84–SM'90–F'98) received the Ph.D. degree from the University of Michigan, Ann Arbor, in 1984.

After receiving the Ph.D. degree, he subsequently joined the faculty at Carnegie Mellon University, Pittsburgh, PA, where he is currently the Stephen J. Jatras Professor of Electrical and Computer Engineering and (by courtesy) Computer Science. His research pioneered much of today's commercial analog circuit and layout synthesis technology. He cofounded Neolinear, Inc., in 1997 and served as its Chief Scientist until its acquisition by Cadence in 2004. He is the founding Director of the MARCO Focus Center for Circuit and System Solutions (called "C2S2"), a consortium of American universities chartered by the U.S. semiconductor community and U.S. government to address future circuit design challenges. His research interests focus on circuit and layout synthesis algorithms for mixed-signal ASICs and for large digital ICs.

Dr. Rutenbar is a member of the Association for Computing Machinery. He has been on the program committees for the IEEE International Conference on CAD, the ACM/IEEE Design Automation Conference, the ACM International Symposium on FPGAs, and the ACM International Symposium on Physical Design. He also served on the Editorial Board of *IEEE Spectrum*. He was the General Chair of the 1996 ICCAD. From 1992 to 1996, he chaired the Analog Technical Advisory Board for Cadence Design Systems. He was a recipient of various awards, namely: the Presidential Young Investigator Award from the National Science Foundation in 1987; the Best/Distinguished Paper Awards from the Design Automation Conference (1987 and 2002), the International Conference on CAD (1991), and the Semiconductor Research Corporation TECHCON (1993, 2000, 2003, and 2005); the Semiconductor Research Corporation (SRC) Aristotle Award for excellence in education in 2001 and the SRC Technical Excellence Award for contributions to tools for analog circuit design in 2004; and the University of Michigan Alumni Society Merit Award in 2002.