Efficient Per-Element Distortion Contribution Analysis via Harmonic Balance Adjoints

Bichen Wu and Jaijeet Roychowdhury
Department of Electrical Engineering and Computer Science, University of California, Berkeley
Emails: {bichen,jr}@berkeley.edu

***Abstract*—We propose a new metric for quantifying per-element distortion that is simple, intuitive and well-defined for both small- and large-signal excitations. Traditional distortion concepts, based on polynomial expansions and Volterra series, can be viewed as an approximation of our new metric. Although computing this metric exactly is quadratic in circuit size, we devise a novel approximation that (unlike polynomial/Volterra) also makes sense for large distortions and can be computed efficiently using adjoint Harmonic Balance. We validate our new approximate metric on a differential pair and the 741 op-amp, comparing it against the full metric and demonstrating order-of-magnitude speedups. Unlike polynomial/Volterra based methods, our proposed per-element distortion computation technique is easy to implement in any modern simulator that features Harmonic Balance or similar steady-state/RF analyses.**

I. Introduction

In analog and RF design, linearity is usually of paramount importance. Even small nonlinearities can lead to distortion that, together with noise and other non-idealities, can impair performance (*e.g.*, SNR, BER) to an un-acceptable extent. Locating the sources of nonlinear distortion in a circuit accurately and efficiently has, therefore, long been of interest to designers. More precisely, what is often desired is an ordered list of devices that are "top distortion contributors"; such a list is typically the starting point for optimizing a circuit for distortion performance.

A key question, of course, is what precisely is meant by the "distortion contribution" of an individual device. In all past work we are aware of, this question has been approached by first linearizing the entire circuit around a DC operating point to obtain an "ideal linear circuit" that has no distortion. Nonlinearities are then incorporated by expanding device equations in Taylor series expansions around the DC operating point, keeping a few higher-order polynomial terms (typically upto the cubic). The effect of these nonlinear terms on distortion at the circuit's output is calculated using an iterative approach, based on Volterra series theory [1]–[5], that applies repeated linear circuit analyses ("AC analyses") using "distortion source inputs" that calculated from each device's polynomial expansion via complex formulæ based on Volterra series [6].

One advantage of this approach is that adjoint computation [7] can be leveraged to find the contribution of each nonlinearity to the total output distortion efficiently [6, 8], even for circuits with many nonlinear elements. However, polynomial/Volterra based approaches suffer from disadvantages that limit applicability as well as implementability. Because they rely on truncated Taylor expansions which lose accuracy rapidly with increasing input strength, they are not well suited for today's analog and RF-IC design styles, which often use strongly nonlinear operations (such as switching) internally while still maintaining a relatively linear I/O relationship. Representing higher-order Taylor terms requires coding/computing higher derivatives (*i.e.*, tensor derivative) terms, which can be tedious and error-prone to implement. Moreover, increasing the number of polynomial terms to much larger than three is usually impractical because the complexity of distortion formulæ that need to be derived and implemented for each nonlinear device model increases very rapidly with the number of polynomial terms. Indeed, implementing Volterra-based distortion analysis poses such significant coding and maintenance challenges that prominent commercial simulators today simply do not offer it at all, leading end-users to devise improvisations such as [9] that, essentially, implement polynomial-based distortion formulæ by scripting the simulator. Such approaches cannot, of course, easily take advantage of the efficiency of adjoint computations.

In this paper, we consider the per-element distortion contribution question from a fresh perspective, one that does not rely on polynomial approximations or Volterra series. We first propose a simple new metric for per-element distortion that is well-defined and makes intuitive sense even if the device or circuit is in strongly nonlinear operation. The core idea behind this metric is to replace a given nonlinear device by its linearization, but *without modifying any other nonlinear devices* at the same time. The difference made to the circuit's output distortion due to this replacement represents its distortion contribution. Proceeding further, we parameterize the replacement of the device using a parameter $\beta \in [0, 1]$[1], with $\beta = 0$ corresponding to full replacement by the linearization, $\beta = 1$ corresponding to no change, and intermediate values representing a transition from the fully nonlinear to a purely linearized device. The manner in which the element's distortion contribution changes with respect to β provides considerable insight into existing polynomial/Volterra based analyses, which (it turns out) compute the derivative of output distortion with respect to β at $\beta = 0$ for *all nonlinear devices*. In other words, polynomial/Volterra approaches compute a *particular approximation* to our new distortion metric.

Computing all per-element distortions using our new metric requires replacing each nonlinear device individually, a procedure that scales roughly quadratically with the size of the circuit (*i.e.*, requires the same order of computation as [9]). To alleviate this and bring the complexity down to linear, we propose a *different approximation* to our distortion metric, namely, the derivative of output distortion with respect to β at $\beta = 1$ for all nonlinear devices. We argue that this approximation better captures per-element distortion than the polynomial/Volterra one because it represents the impact of moving each device towards linearity without changing the other devices, which remain in fully nonlinear operation.

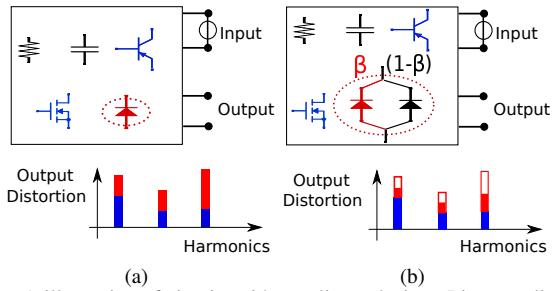
Importantly, we show that our proposed approximation around $\beta = 1$ can be computed efficiently (*i.e.*, in computation/memory linear in the size of the circuit) using adjoint harmonic balance[2] [11, 12]. Another important advantage is that – unlike for polynomial/Volterra based methods – implementing our method need involve no modification to existing device model code in any modern, well-structured simulator; the parameters β can be introduced outside the device model, and will typically involve only a one-time change of a few lines in the code responsible for building circuit equations. Our technique thus combines desirable features that have eluded previous approaches: it computes per-element distortions that are meaningful for large- as well as small-signal operation, it does so in a scalable and efficient manner, and it can be implemented easily in modern simulators without touching device models.

We present results on a differential pair and the 741 op-amp from an implementation in our simulator MAPP[3]. We compare the proposed derivative approximation around $\beta = 1$ with our full distortion metric and demonstrate that the former serves as an excellent proxy for rank-ordering distortion contributors. We obtain order-of-magnitude speedups for our examples; these speedups grow roughly linearly with the number of nonlinear devices in the circuit.

The remainder of the paper is organized as follows. In Section II-A, we provide necessary background on formulating circuit equation as DAEs and describe how to introduce the β parameters noted above. Next, in Section II-B, we briefly recapitulate adjoint harmonic balance. Our main contributions (the new per-element distortion metric and the new approximation around $\beta = 1$) are developed in Section II-C. Finally, in Section III, we demonstrate our method on representative circuits and also illustrate how incorporating feedback can help reduce per-element distortion contributions as calculated by our method.

II. Defining, Approximating and Computing Distortion

A. Incorporating β parameters into circuit DAEs



(a) (b)

Fig. 1. A illustration of circuits with non-linear devices. Linear or linearized devices are colored black, while non-linear devices are colored red or blue. (a) Original circuit with non-linear devices. (b)Assign the parameter β to the circled device.

A circuit with non-linear devices, as illustrated by Fig. 1(a), can be described by differential algebraic equations (DAEs) (1) [13]

[1]Each nonlinear device has its own separate β parameter.

[2]Although outside this paper's scope, shooting [10] can also be used.

[3]MAPP (Model and Algorithm Prototyping Platform) is a well-structured, MATLAB-based electronic and multi-physics simulator that we expect to release freely as open-source by Dec 2014.

$$\frac{\vec{q}(\vec{x}(t))}{dt} + \vec{f}(\vec{x}(t), \vec{u}(t)) = \vec{0}. \quad (1)$$

$\vec{x}(t)$ is the unknown of the circuit and $\vec{u}(t)$ is the input. The derivative term $\vec{q}(\vec{x}(t))$ describes memory characteristics of the circuit, such as charge or flux. The non-derivative term $\vec{f}(\vec{x}(t))$ describes memoryless characteristics of the circuit, such as voltage or current. Output of the circuit $y(t)$ is usually a linear combination of the unknown $\vec{x}(t)$ (2)

$$y(t) = \vec{l}^{\top}\vec{x}(t). \quad (2)$$

Given a periodic input, the output of the circuit will eventually become periodic with the same frequency. And because non-linear devices in the circuit will generate higher harmonics, the output of the circuit will also contain distortions on higher harmonics. Distortion contribution from a non-linear device can be illustrated in Fig. 1(a). The red bars on the output distortion represent the distortion contribution from the red non-linear device.

A non-linear device in a circuit can be described by (3)

$$\vec{y}_D(t) = \overrightarrow{NL}(\vec{x}_D(t)) = \frac{d\vec{q}_{NL}(\vec{x}_D(t))}{dt} + \vec{f}_{NL}(\vec{x}_D(t)), \quad (3)$$

where $\vec{x}_D(t)$ is the input of the device and $\vec{y}_D(t)$ is the output. $\vec{q}_{NL}$ and $\vec{f}_{NL}$ respectively describe memory and memoryless characteristics of the device.

The distortion contribution of a device is generated by its non-linearity. So if (3) is linearized at its operating point $\vec{x}_D^*$ and $\vec{y}_D^*$ as (4)

$$\begin{aligned}\vec{y}_D(t) = \overrightarrow{LIN}(\vec{x}_D(t)) = &\frac{d\vec{q}_{NL}(\vec{x}_D^*)}{d\vec{x}_D}\frac{d\vec{x}_D(t)}{dt} \\ &+ \vec{y}_D^* + \frac{d\vec{f}_{NL}(\vec{x}_D^*)}{d\vec{x}_D}(\vec{x}_D(t) - \vec{x}_D^*),\end{aligned} \quad (4)$$

the device will no longer generate higher harmonics, and it will not contribute distortions to the output.

In order to observe non-linear devices' individual distortion contribution to the output, we can respectively replace each non-linear device with its linearized counterpart and compare the difference of the output distortion. To do this, we assign a new parameter β to each non-linear device. As shown in (5)

$$\vec{y}_D(t) = (1-\beta)\overrightarrow{LIN}(\vec{x}_D(t)) + \beta\overrightarrow{NL}(\vec{x}_D(t)), \quad (5)$$

β is the combinational factor of the original non-linear device's output and the linearized device's output. If β is 0, the device becomes the purely linearized model . If β is 1, the device is still the original non-linear model By adjusting β between 0 and 1, we will be able to observe the non-linear device's individual effect on the output, as is illustrated in Fig. 1(b).

Now the DAE (1) can be re-formulated as (6)

$$\frac{\vec{q}(\vec{x}(t), \vec{\beta})}{dt} + \vec{f}(\vec{x}(t), \vec{u}(t), \vec{\beta}) = \vec{0}, \quad (6)$$

where $\vec{\beta}$ is the vector whose components are β_i for each non-linear device. This work has been implemented in Berkeley Model & Algorithm Prototyping Platform (MAPP), a multi-physics simulator for compact model and analysis algorithm development. In this platform, this implementation can be done by adding a few lines of code completely on DAE level, and no modifications on device models are needed.

B. Harmonic Balance adjoint sensitivity analysis

To compute the distortion of the circuit, Harmonic Balance (HB) analysis is applied to obtain the periodic steady state solution of (6). In HB, the periodic unknown $\vec{x}(t)$ and input $\vec{u}(t)$ are expanded into truncated Fourier series as (7),

$$\vec{u}(t) = \sum_{k=-M}^{M} \vec{U}_k e^{j2\pi k f_0 t}\ ,\ \vec{x}(t) = \sum_{k=-M}^{M} \vec{X}_k e^{j2\pi k f_0 t}, \quad (7)$$

where M is the highest order of harmonics, f_0 is the fundamental frequency, $\{\vec{U}_k\}$ and $\{\vec{X}_k\}$ are the Fourier coefficients of $\vec{x}(t)$ and $\vec{u}(t)$.

The unknown $\vec{x}(t)$ and the input $\vec{u}(t)$ are both periodic, so $\vec{q}(\vec{x}(t), \vec{\beta})$ and $\vec{f}(\vec{x}(t), \vec{u}(t), \vec{\beta})$ are also periodic with the same frequency. Thus, we can sample the periodic waveform of $\vec{q}$ and $\vec{f}$ and transform them to frequency domain as truncated Fourier series as (8)

$$\begin{aligned}\vec{q}(\vec{x}(t), \vec{\beta}) &= \sum_{k=-M}^{M} \vec{Q}_k e^{j2\pi k f_0 t}, \\ \vec{f}(\vec{x}(t), \vec{u}(t), \vec{\beta}) &= \sum_{k=-M}^{M} \vec{F}_k e^{j2\pi k f_0 t}.\end{aligned} \quad (8)$$

$\{\vec{Q}_k\}$ and $\{\vec{F}_k\}$ are Fourier coefficients of $\vec{q}$ and $\vec{f}$.

It's not hard to be convinced that $\{\vec{F}_k\}$ and $\{\vec{Q}_k\}$ can be regarded as functions of $\{\vec{X}_k\}$ and $\{\vec{U}_k\}$. Besides, on each harmonic $\vec{Q}_k e^{j2\pi k f_0 t}$ and $\vec{F}_k e^{j2\pi k f_0 t}$ still satisfy the equation (6). So, we can derive Harmonic Balance equations from DAE (6) as (9)

$$\vec{H}_{HB}(\vec{X}_N, \vec{\beta}) = \sum_{k=-M}^{M} j2\pi k f_0 \vec{Q}_k(\vec{X}_N, \vec{\beta}) + \vec{F}_k(\vec{U}_N, \vec{X}_N, \vec{\beta}) \approx \vec{0}. \quad (9)$$

$N = 2M + 1$ is the total number of harmonics. $\vec{X}_N$ is the vector built by stacking $\{\vec{X}_k\}$ together. $\vec{U}_N$ is stacked by $\{\vec{U}_k\}$. (2) is modified as (10)

$$\vec{Y}_N = (I_N \otimes \vec{l}^{\top})\vec{X}_N. \quad (10)$$

$\vec{Y}_N$ is stacked by Fourier coefficients of the output $\vec{y}(t)$. $\vec{l}^{\top}$ is modified as $I_N \otimes \vec{l}^{\top}$ where I_N is an identity matrix of size N and $\otimes$ is the Kronecker product operation. For example, if $N = 3$, $I_N \otimes \vec{l}^{\top}$ can be expressed as the matrix in (11).

$$I_3 \otimes \vec{l}^{\top} = \begin{pmatrix} \vec{l}^{\top} & & \\ & \vec{l}^{\top} & \\ & & \vec{l}^{\top} \end{pmatrix} \quad (11)$$

Distortion of the output can be obtained by solving (9). Furthermore, we can compute sensitivity of the output $\vec{Y}_N$ with respect to the parameter $\vec{\beta}$ by adjoint sensitivity analysis. Consider a small perturbation on the parameter $\Delta\vec{\beta}$ causes a small change on the unknown $\Delta\vec{X}_N$, their relationship can be described by (12)

$$\begin{aligned}&\vec{H}_{HB}(\vec{X}_N^* + \Delta\vec{X}_N, \vec{\beta}^* + \Delta\vec{\beta}) = \vec{0}, \\ &\frac{\partial\vec{H}_{HB}}{\partial\vec{X}_N}\Delta\vec{X}_N + \frac{\partial\vec{H}_{HB}}{\partial\vec{\beta}}\Delta\vec{\beta} \approx \vec{0}, \\ &\Delta\vec{X}_N \approx -(\frac{\partial\vec{H}_{HB}}{\partial\vec{X}_N})^{-1}\frac{\partial\vec{H}_{HB}}{\partial\vec{\beta}}\Delta\vec{\beta}.\end{aligned} \quad (12)$$

$\vec{X}_N^*$ and $\vec{\beta}^*$ are the solutions of (9).

For simplicity, we will refer to the Jacobian of $\vec{H}_{HB}$ with respect to $\vec{X}_N$ as matrix H, and the Jacobian with respect to $\vec{\beta}$ as matrix B. Sensitivity of the unknowns $\vec{X}_N$ can be described as its derivative with respect to the parameter $\vec{\beta}$ as (13)

$$\frac{\partial\vec{X}_N}{\partial\vec{\beta}} = \lim_{\Delta\vec{\beta}\to\vec{0}}\frac{\Delta\vec{X}_N}{\Delta\vec{\beta}} = -H^{-1}B. \quad (13)$$

And sensitivity of the output can then be expressed as (14)

$$\frac{\partial\vec{Y}_N}{\partial\vec{\beta}} = (I_N \otimes \vec{l}^{\top})\frac{\partial\vec{X}_N}{\partial\vec{\beta}} = -(I_N \otimes \vec{l}^{\top})H^{-1}B. \quad (14)$$

Computing the inverse of the Jacobian H in (14) directly is very expensive especially for Harmonic Balance equations. However in the adjoint system (15)

$$\frac{\partial\vec{Y}_N}{\partial\vec{\beta}}^{\top} = -B^{\top}H^{-\top}(I_N \otimes \vec{l}), \quad (15)$$

$H^{-\top}(I_N \otimes \vec{l})$ can be computed altogether by sparse matrix techniques with linear time cost. And the following sparse matrix multiplication with matrix $B^{\top}$ also takes linear time cost. Thus, the overall time cost for HB adjoint sensitivity analysis to compute the sensitivity of the output is still linear.

C. Quantification and computation of device distortion contribution

1) Quantification of device distortion contribution: First of all, set all $\vec{\beta}$ in (9) to be $\vec{1}$ and apply HB to compute the output $\vec{Y}_N$. The total output distortion on each harmonic is illustrated in Fig. 1(a). To exclude the non-linear effect of non-linear device i on the output, we can replace it with its linearized counterpart by setting its β_i to 0. Apply HB again to compute the new output $\vec{Y}_N^{(i)}$. This output doesn't include distortion contributed by device i. Then, the distortion contribution from device i, or $\overrightarrow{Y}_N^{(i)}$ can be justifiably quantified as (16)

$$\overrightarrow{Y}_N^{(i)} \triangleq \vec{Y}_N - \vec{Y}_N^{(i)}. \quad (16)$$

2) Rigorous but brute force method: This quantification naturally leads to a rigorous but brute force method to compute distortion contribution from each non-linear device. First set $\vec{\beta}$ as $\vec{1}$ and apply HB to compute the original output $\vec{Y}_N$. Then for each device i, set β_i to 0. Apply HB on the modified circuit to compute the new output $\vec{Y}_N^{(i)}$. Then use (16) to compute $\overrightarrow{Y}_N^{(i)}$, the distortion contribution from device i. To obtain distortion contribution of every device with this method, HB needs to be applied for every non-linear device in the circuit, which is computationally expensive. Therefore this brute force method doesn't scale as the number of non-linear devices in the circuit grows.

3) Proposed method using adjoint sensitivity analysis: Here we propose a more efficient method to replace the brute force one. It also starts with applying HB on the original circuit to compute $\vec{Y}_N$. But next instead of computing the exact distortion contribution, HB adjoint sensitivity analysis is applied to compute the derivative of the output with respect to β_i of each non-linear device. This derivative is then used as an approximation of the device distortion contribution computed by brute force method.

Now we will explain why this is a reasonable approximation. The distortion of the output can be treated as a function of β_i as $\overrightarrow{Y}_N^{(i)}(\beta_i)$.

Alter the parameter β_i from 1 to 0, the output $\overrightarrow{Y}_N^{(i)}(\beta_i)$ will also change from $\vec{Y}_N$ to $\vec{Y}_N^{(i)}$. By definition, device distortion contribution defined in (16) can be expressed as (17)

$$\overrightarrow{Y}_N^{(i)} \triangleq \vec{Y}_N - \vec{Y}_N^{(i)} \triangleq \overrightarrow{Y}_N^{(i)}(1) - \overrightarrow{Y}_N^{(i)}(0). \quad (17)$$

This can be computed by the brute force method precisely. And the derivative of the outputwith respect to each β_i can be expressed as $\frac{\partial \overrightarrow{Y}_N^{(i)}(1)}{\partial \beta_i}$. This derivative can be computed efficiently by HB adjoint sensitivity analysis

The relationship between the device distortion contribution and the derivative approximation is illustrated in Fig. 2. As an example, the distortion-β curve of a illustrative device is plotted as the blue solid line in Fig. 2. The exact distortion contribution $\overrightarrow{Y}_N^{(i)}$ of the device, which can be computed by the brute force method, is just the difference between the starting and ending value of the distortion-β curve. Tangent line of the curve is also plotted as the red dash line in Fig. 2. The derivative approximation $\frac{\partial \overrightarrow{Y}_N^{(i)}(1)}{\partial \beta_i}$, which can be computed by the proposed method, is just the difference between the starting and ending value of the tangent line. Generally, Distortion contribution from different devices in a circuit usually have orders of difference. But for each device, the derivative $\frac{\partial \overrightarrow{Y}_N^{(i)}(1)}{\partial \beta_i}$ is generally of the same order as the exact distortion contribution $\overrightarrow{Y}_N^{(i)}$. Thus, the derivative computed by the HB adjoint sensitivity analysis is a good approximation to identify different devices in the circuit.

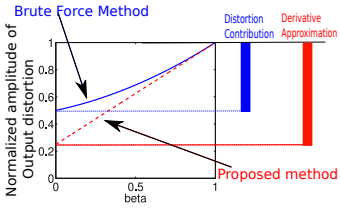


Fig. 2. Distortion-β curve of a illustrative device is plotted as the solid blue line. The tangent line of the curve is plotted as the red dash line. Distortion contribution of the device is measured as the difference between starting and ending value of the distortion-β curve, and the derivative approximation of the device is measured as the difference between starting and ending value of the dash line.

Besides, compared with providing circuit designers the exact value of device distortion contribution, it's more valuable to provide a sorted list of devices ranked by the amplitude of each device's distortion contribution. With this information, circuit designers are able to identify the most significant distortion contributors. In this sense, the derivative approximation is usually good enough to preserve this ranking.

Compared with the brute force method which needs to do HB for every non-linear device and 1 HB on the original circuit, the proposed method requires only 1 HB on the original circuit in addition to ignorable time cost for adjoint sensitivity analysis. So the scalability of this method is much better then the brute force method.

III. Validation on Representative Circuits

In this section, we will apply the proposed algorithm to analyze two sample circuits, a Bipolar Junction Transistor (BJT) differential pair with active load and a 741 Operational Amplifier (OpAmp) with negative feedback. We will first briefly introduce each circuit. Then we will discuss the result of the proposed algorithm and compare it with the rigorous but brute force method. In Section III-B, we will also see how the proposed method help to improve the overall circuit linearity.

A. A BJT differential pair with active load

Fig. 3 is the schematic of a BJT differential pair with active load. In this circuit, Q_7 and Q_6 form a current mirror and Q_7 provides a constant current for Q_5 and Q_4. Q_5 and Q_4 are differential pairs that transfer the differential input to the output. Q_2 and Q_3 are two PNP transistors that form active load, and they are biased by Q_1 such that current through Q_2 and Q_3 are half of the current through Q_7. Output of the circuit is define to be $V_{out+} - V_{out-}$. Supply voltage V_{dd} is $5V$. Common mode voltage source V_1 supply a DC voltage of $1.2V$ to shifts V_{out+} and V_{out-} to the middle such that the output can have largest range. Differential input is provided by V_2, which in our experiment, is a sinusoidal signal. Frequency of the input is $1kHz$ and amplitude is $1.3mV$.

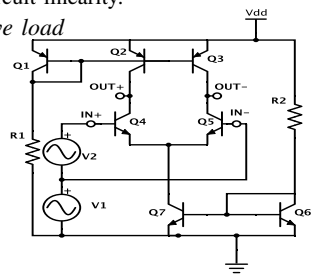


Fig. 3. Schematic of a BJT differential pair

Apply Harmonic Balance analysis on the original circuit to compute the output of the circuit. Frequency domain results are shown in Fig. 4. Distortions on even harmonics are low due to differential output cancellation. But on odd harmonics, for example the 3rd harmonic, distortion can be measured as large as -36dB.

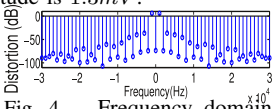


Fig. 4. Frequency domain output of the differential pair

To analyze the distortion contribution to the 3rd harmonic from each device, adjoint sensitivity analysis is applied to the circuit and its result is shown as the red bars in Fig. 5. Amplitude of distortion contribution of each device is plotted along y-axis, and devices are sorted by the amplitude of distortion contribution along x-axis. It can be observed that Q_7 and Q_6 contributed least distortion. As current mirror, they are working in at the DC operating point such that their non-linear effect can be ignored. Same is true for $Q1$. $Q2$ and $Q3$ as active load contribute some amount of distortion. But because their V_{BE} are fixed by Q_1, they are not contributing as much distortion as $Q4$ and $Q5$. Inputs to $Q4$ and $Q5$ change dramatically that their non-linear effect can no longer be ignored.

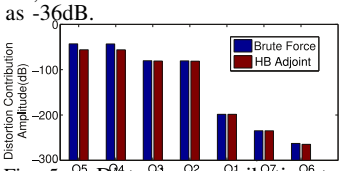


Fig. 5. Distortion contribution to the 3rd harmonic of the output from each transistor

The brute force method is also applied to compute each device's exact distortion contribution and the result is shown as blue bars in Fig. 5. For devices such as Q_5, brute force method computes a similar yet different amplitude of distortion contribution. However the ranking provided by brute force method is the same as the proposed method. A more precise comparison between the values computed by both method can be found in Table.I.

TABLE I
Device Distortion Contribution Analysis on the Differential Pair

HB adjoint		Brute Force	
Q_5:	$-2.560e{-}3-j5.223e{-}6$	Q_5:	$-7.007e{-}3-j1.599e{-}5$
Q_4:	$-1.548e{-}3-j5.383e{-}6$	Q_4:	$-6.825e{-}3-j1.570e{-}5$
Q_3:	$-8.766e{-}5-j1.690e{-}5$	Q_3:	$-9.512e{-}5-j1.706e{-}5$
Q_2:	$-8.488e{-}5-j1.705e{-}5$	Q_2:	$-9.183e{-}5-j1.720e{-}5$
Q_1:	$-1.184e{-}10-j4.622e{-}12$	Q_1:	$-1.184e{-}10-j4.604e{-}12$
Q_7:	$-2.255e{-}14+j1.806e{-}12$	Q_7:	$-2.095e{-}14+j1.837e{-}12$
Q_6:	$+5.737e{-}14+j1.023e{-}14$	Q_6:	$+6.911e{-}14+j2.372e{-}14$
CPU time: 344s		CPU time: 1148s	

From Table.I we can also learn that CPU time of the proposed algorithm is about 3.3X faster than the brute force method. The circuit contains 7 non-linear devices, and by rough estimation, the brute force method should be about 8X slower than the proposed method because it requires 1 HB on the original circuit and 7 HB for every non-linear device in the circuit while the proposed method requires only 1 HB in addition to trivial extra amount of time cost for adjoint sensitivity analysis. The reason that brute force method is not 8X slower is because the solution of the first HB on the original circuit is used as initial guess for successive HB. After linearizing a single device, the new solution is still very close to the original one. So Newton-Raphson takes less iterations to converge and therefore successive HB also takes less time than the original HB. But even so, applying HB for every non-linear device is still not scalable for circuits with large number of non-linear devices.

B. An Operational Amplifier with negative feedback

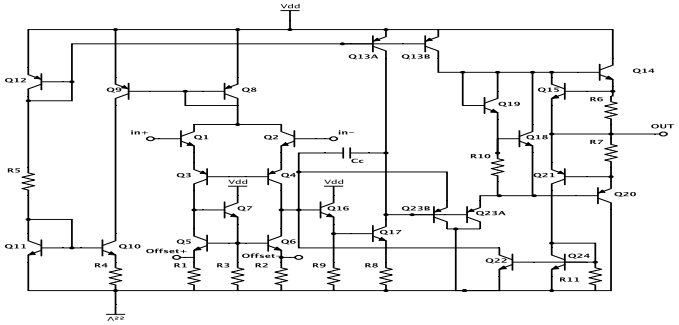


Fig. 6. Schematic of a 741 Operational Amplifier

741 Operational Amplifier [14] in Fig. 6 is a more complicated example which contains 26 BJTs. Transistors such as Q_{11}, Q_{12}, Q_{10} *etc.*act as current mirrors and determine the reference current of the circuit. Q_1 to Q_6 form a monolithic amplifier that drives following stages. Q_{16}, Q_{17} *etc.*form the intermediate stage and Q_{14}, Q_{20} *etc.*form output stage of the circuit. Q_{15}, Q_{21}, Q_{22}, Q_{24} *etc.*perform short-circuit protection and are normally off. Q_{23B} is designed to prevent excessive current flowing into base of Q_{16}. As shown in Fig. 7, negative feedback is used to obtain a closed-loop gain of $-2X$. In the experiment, supply voltage V_{dd} and V_{ee} are respectively $15V$ and

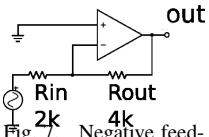


Fig. 7. Negative feedback configuration

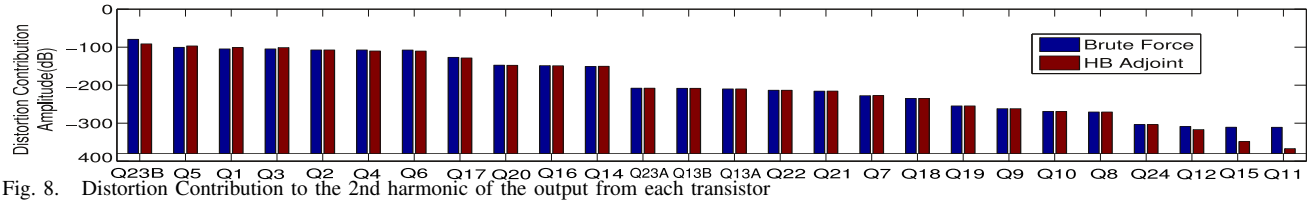
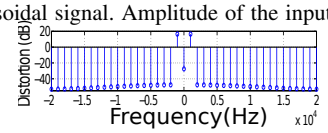


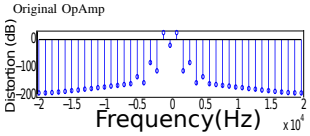
Fig. 8. Distortion Contribution to the 2nd harmonic of the output from each transistor

$-15V$. Input to this circuit is a sinusoidal signal. Amplitude of the input is $6.7V$ and frequency is $1kHz$.

Harmonic Balance is applied to this circuit to obtain output distortion. Frequency domain results are shown in Fig. 9(a). Distortion can be observed on each harmonic, descending slowly as order of harmonic grows. On the 2nd harmonic, for example, the distortion is about $-47dB$.



(a) Frequency domain and time domain output of the Original OpAmp



(b) Frequency domain and time domain output of the OpAmp after removing Q_{23B}

Fig. 9. Frequency and time domain output of the OpAmp

To analyze the distortion contribution to the 2nd harmonic from each device, HB adjoint analysis is then applied and result is shown as the red bars in Fig. 8. Amplitude of distortion contribution of each device is plotted along y-axis and devices are sorted by the amplitude of distortion contribution in descending order along x-axis. We can learn from Fig. 8 that contribution from Q_{11}, Q_{12} *etc.*are below $-200dB$. This is because they basically act as current mirrors and are working at the DC operating point. Therefore their non-linear effect can be ignored. Besides, transistors such as Q_{15}, Q_{21}, Q_{22}, Q_{24} *etc.*are normally off, thus contribute little distortion as well. It's no surprising that contribution by Q_1 to Q_6 are significant. They by themselves form a monolithic amplifier at the input stage thus suffer from violent shift of operating point. What's more, their contributions are also amplified by successive stages. Q_{23B} is identified as the most significant contributor, which we will talk about later.

Brute force method is also applied and the result is shown as blue bars in Fig. 8. Again, for devices such as Q_{23B}, brute force method provides a similar yet different amplitude of distortion contribution. It's worth noting that along x-axis, the device ranking provided by brute force method is slightly different from the proposed method by placing Q_6 after Q_2 and Q_4. This is because exact distortion contribution from Q_6, Q_2 and Q_4 are very close to each other, as listed in the right column of Table.II. For devices whose contribution have orders of difference, approximation computed by adjoint analysis is still good enough to distinguish them. A more precise comparison can be found in Table.II. It also shows that proposed algorithm is about 9X faster than the brute force method.

TABLE II
DEVICE DISTORTION CONTRIBUTION ANALYSIS ON THE OPAMP

$V_{in} = 6.7V \times \sin(2\pi 10^3 t)$

HB adjoint		Brute Force	
Q_{23B}:	$-2.685\text{e}{-5} - j2.411\text{e}{-9}$	Q_{23B}:	$-1.071\text{e}{-4} - j1.167\text{e}{-6}$
Q_5:	$-1.451\text{e}{-5} + j6.578\text{e}{-8}$	Q_5:	$-9.513\text{e}{-6} - j5.851\text{e}{-8}$
Q_1:	$+8.828\text{e}{-6} - j3.494\text{e}{-8}$	Q_1:	$+5.897\text{e}{-6} - j2.553\text{e}{-8}$
Q_3:	$+8.798\text{e}{-6} - j3.483\text{e}{-8}$	Q_3:	$+5.879\text{e}{-6} - j2.545\text{e}{-8}$
Q_2:	$-3.103\text{e}{-6} + j1.992\text{e}{-8}$	Q_2:	$-4.292\text{e}{-6} - j2.735\text{e}{-8}$
Q_4:	$-3.094\text{e}{-6} + j1.987\text{e}{-8}$	Q_4:	$-4.278\text{e}{-6} + j2.726\text{e}{-8}$
Q_6:	$+4.277\text{e}{-6} - j9.374\text{e}{-8}$	Q_6:	$+4.175\text{e}{-6} + j1.066\text{e}{-7}$
CPU time: 830s		CPU time: 7489s	

Both methods in Fig. 8 reveal that Q_{23B} is the most significant distortion contributor, which is quite surprising. Because by design, Q_{23B} should just prevent excessive current flowing into the base of Q_{16} and shouldn't effect the output too much. With this clue, one possible option to improve the overall circuit linearity is to remove Q_{23} from the circuit. Experiment proves this option to be very effective. Distortion on second harmonic falls dramatically to $-119dB$ after removing Q_{23B}, as shown in Fig. 9(b).

IV. CONCLUSION

In this paper we proposed a new quantification of device distortion contribution and an efficient method to compute its approximation. The proposed method is easy to implement and requires no modification on device models. It is computationally low cost, requiring only one Harmonic Balance and one adjoint sensitivity analysis. Therefore, it is scalable for circuits with large number of non-linear devices. This work has been implemented in Berkeley Model & Algorithm Prototyping Platform (MAPP), and will be released as open source.

Acknowledgments

We thank N. Krishnapura for pointing us to his paper [9]. This work is partially supported by the NSF NEEDS project.

REFERENCES

[1] V. Volterra. *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover, 1959.
[2] W. Rugh. *Nonlinear System Theory – The Volterra-Wiener Approach*. Johns Hopkins Univ Press, 1981.
[3] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley, 1980.
[4] S. Narayanan. Transistor distortion analysis using Volterra series representation. *Bell System Technical Journal*, May-June 1967.
[5] Y.L. Kuo. Distortion Analysis of Bipolar Transistor Circuits. *IEEE Trans. Ckt. Theory*, CT-20(6), November 1973.
[6] J.S. Roychowdhury. SPICE3 Distortion Analysis. Master's thesis, EECS department, University of California, Berkeley, Electronics Research Laboratory, April 1989. Memorandum no. UCB/ERL M89/48.
[7] S.W. Director and R.A Rohrer. The Generalized Adjoint Network and Network Sensitivities. *IEEE Trans. Ckt. Theory*, 16:318–323, August 1969.
[8] S H. Chisholm and L.W. Nagel. Efficient Computer Simulation of Distortion in Electronic Circuits. *IEEE Trans. Ckt. Theory*, CT-20:742–745, November 1973.
[9] N. Krishnapura and K.S. Rakshitdatta. A Model-Agnostic Technique for Simulating Per-Element Distortion Contributions. In *Proc. IEEE CICC*, pages 1–4, September 2013.
[10] S. Skelboe. Computation of the periodic steady-state response of nonlinear networks by extrapolation methods. *IEEE Trans. Ckts. Syst.*, CAS-27:161–175, 1980.
[11] K.S. Kundert, J.K. White, and A. Sangiovanni-Vincentelli. *Steady-state methods for simulating analog and microwave circuits*. Kluwer Academic Publishers, 1990.
[12] J.W. Bandler, Q.J. Zhang, and R.M. Biernacki. Practical high speed gradient computation for harmonicbalance simulators. In *Proc. IEEE MTT Symp.*, volume 1, pages 363–366, 1989.
[13] Jaijeet Roychowdhury. Numerical simulation and modelling of electronic and biochemical systems. *Foundations and Trends in Electronic Design Automation*, 3(2-3):97–303, December 2009.
[14] P.R. Gray, P.J. Hurst, S.H Lewis, and R.G. Meyer. *Analysis and Design of Analog Integrated Circuits*. Wiley, 4th edition, 2001.