A Survey of Fast Analog Circuit Analysis Algorithm using SPICE

T.Murugajothi
Assistant Professor, PSNA College of Engineering & Technology, Tamilnadu, India
jotbyece@gmail.com

Abstract—This paper presents a fast analog circuit analysis algorithm, fundamental circuit-based circuit analysis, for circuits being repeatedly modified and verified in product development. The algorithm reduces previous circuit simulation result on successive changed circuit analysis to achieve simulation operation reduction. The algorithm is implemented with SPICE simulator on linear and nonlinear circuit applications with the proposed device delta models. The experiments show that the algorithm increases the speed of the circuit simulation five to ten times over directly simulations under the same simulation accuracy.

Index Terms—Analog circuit simulation, behavior modeling, CAD, circuit optimization design and verification, SPICE.

I. INTRODUCTION

Computer-aided simulation technique is widely used in circuit design and verification. Increasing simulation speed is always a hot topic of important economic significance. For several decades, scientists and engineers have been working on simulation algorithms to increase operational efficiency. Various effective algorithms and techniques are used in simulation program with integrated circuit emphasis (SPICE) simulation engines [6], [7]. The SPICE simulation engines have stabilized and the improvement in simulation speed has slowed. In this paper, we focus our work on how to use an existing SPICE simulator effectively to achieve higher simulation speed.

In circuit design and verification, circuit element parameters are tuned constantly and various working environments and manufacture processes are evaluated. As a result, hundreds of circuit simulations are performed to obtain a reliable product. Traditionally, simulation operations in the above processes are executed independently. A simulator does not recognize any relationships among similar circuits. The total simulation time is straightly calculated as one circuit simulation time multiplied by the number of simulations. It is easily accumulated to a considerable amount and consumes a large portion of the resources in the product development.

For the above scenarios, we present a new circuit simulation algorithm for analog circuit design modification and verification. When simulation operations need to be executed on a circuit over and over again with varying parameters, the algorithm applies the previous circuit’s simulation results on subsequent assessed circuits to reduce the total simulation time.

In this paper, the basic theory of the proposed algorithm is discussed and the algorithm is implemented with a SPICE simulator in Section II. In Section III, the approach is applied to nonlinear circuits with the device delta models. In Section IV, the experiments show the algorithm’s efficiency and accuracy.

II. FUNDAMENTAL CIRCUIT-BASED CIRCUIT ANALYSIS

In this section, upon to the relationship between a given circuit and its modified circuits, we develop a new algorithm in SPICE simulation and discuss its applications. We assume that the proposed circuits are linear circuits and discuss nonlinear circuits in Section III.

A. Fundamental Circuit, Derivative Circuit, and Delta Circuit

A circuit is built up with components. Component parameter value changes will create a new circuit.

Definition 1: For a given circuit, a modified circuit is created from component parameter value changes. The given circuit is called the fundamental circuit of the modified circuit. The modified circuit is called the derivative circuit of the given circuit.

For a given linear circuit, the circuit nodal matrix equation in SPICE is written as follows:

$$Y_v \cdot V_i = I$$

where $Y_v$ is the circuit nodal admittance matrix, $V_i$ is the circuit node voltage vector, and $I$ is the input vector.

Similarly, the circuit nodal matrix equation of a derivative circuit of (1) is written as follows:

$$Y_{v_{d}} \cdot V_{i} = I$$

where $Y_{v_{d}}$ is the derivative circuit nodal admittance matrix and $V_{i}$ is the circuit node voltage vector. The circuit holds the same circuit input vector $I$ as its fundamental circuit (1).

By Definition 1, a derivative circuit has the same circuit topology as its fundamental circuit. Every variable in a derivative circuit has a corresponding variable in its fundamental circuit. The two circuits mirror each other, and so do $V_{i}$ and $V_{i_{d}}$, and $V_{o}$ and $V_{o_{d}}$.

Now let

$$\Delta V = V_{o_{d}} - V_o$$

and

$$\Delta Y = Y_{o_{d}} - Y_o.$$  \hspace{1cm} (3a)

By combining (1) and (2), we have the following:

$$Y_v \cdot \Delta V = -\Delta Y \cdot V_i$$

By comparing (4) with (1) or (2), we imagine a virtual circuit represented by (4).
Definition 2: A virtual circuit defined by (4) is called the delta circuit of the fundamental circuit (1) and its derivative version (2). Circuits (1) and (2) are the original circuits of the delta circuit (4). The delta circuit nodal voltage $\Delta V$ is the difference of the nodal voltage of its two original circuits.

Based on its fundamental circuit solution, a derivative circuit can be indirectly resolved from its delta circuit solution through (3a). This procedure is called fundamental circuit-based circuit analysis (FCBCA).

### B. Signal Speed Versus Simulation Speed

Signal speed represents how fast a signal changes. In circuit transient simulation, high speed signals, or fast changing signals, take longer simulation time than low speed signals. It is because under the same simulation accuracy, high speed signals require smaller time-step than low speed signals.

Literally, simulation time-step is controlled by signal’s derivatives. A signal with a smaller derivative is changing slower and needs shorter simulation time than a signal with larger derivative.

**Theorem 1:** Suppose two time domain signals, $x(t)$ and $y(t)$, have Fourier transforms $x(w)$ and $y(w)$, with their high-end cutoff frequency $w_a$ and $w_b$, respectively. Signal $x(t)$ is changing slower than signal $y(t)$, if

1. $w_a < w_b$ and $|x(w)| \approx |y(w)|$
   
   or

2. $w_a \approx w_b$ and $|x(w)| < |y(w)|$

where $| \cdot |$ represents the magnitude of the signal.

**Proof:**

1. Based on the theorem’s condition 1, we have the following:

   $$y(w) = [f(t)e^{jw\Delta t}]$$
   $$y(w) = [f(t)e^{jw\Delta t}]$$

   where $\Delta \theta$ is the phase difference of $x(w)$ and $y(w)$.

   Applying derivative operation on $x(t)$ and $y(t)$, we have the Fourier transformers of $dx/dt$ and $dy/dt$, $x_d(w)$ and $y_d(w)$, with the following:

   $$|x_d(w)| = |x(w)|$$
   $$y_d(w) = [x_d(w) + i\omega |x(w)|] e^{j\omega t}.$$ 

   Apparently, $|x_d(w)| > |x(w)|$ since $|x(w)| > 0$ and $y(w) \neq 0$ when $w = (w_a, w_b)$, i.e., $y(t)$ is changing faster than $x(t)$.

2. According to condition 2, suppose $|\omega| < 0$ and $|y(\omega)| = |x(\omega)| + |\omega| |a(\omega)|$. Then, $y(\omega)$ can be expressed as follows:

   $$y(\omega) = |x(\omega)| + |\omega| |a(\omega)|$$

   where $\theta$ is the phase of $y(\omega)$.

   The Fourier transformers of $dx/dt$ and $dy/dt$, $x_d(w)$ and $y_d(w)$, have the relationship as follows:

   $$|y_d(w)| = |x_d(w)| + \omega |\omega| |a(\omega)|.$$ 

   Since $|\omega| |\omega| > 0$, $|y_d(w)| > |x_d(w)|$. $y(t)$ is changing faster than $x(t)$. A slower changing signal is simply called a slower signal.

---

**Fig. 1.** Constraints of a delta signal being a slower signal are satisfied in the shadowed area, where $\eta = \frac{|x_d(w)|}{|x(w)|}$ and $\Delta \theta(w) = \theta_x(w) - \theta_y(w)$.

**C. Guideline of Using Delta Circuit in FCBCA Simulation**

In this section, we deduce the conditions that the FCBCA runs faster than independent individual simulation. Based on Theorem 1, we propose the following.

**Theorem 2:** Suppose two signals $x_d(w)$ and $y_d(w)$ are the corresponding signals from a fundamental circuit and its derivative circuit. Their high-end cutoff frequencies are $w_a$ and $w_b$, respectively. If

$$w_a \approx w_b$$

and

$$\cos(\Delta \theta) \geq \eta/2$$

and

$$\cos(\Delta \theta) \geq 1/2\eta$$

where $\eta = \frac{|x_d(w)|}{|x(w)|}$ and $\Delta \theta$ is the phase difference of $x_d(w)$ and $x(w)$, then the delta signal is a slower signal compared to the two corresponding signals in the original circuits.

Interested readers may see [11] for the proof of Theorem 2. Fig. 1 shows the constraints in (5) with the shadowed area, where the delta signal is a slower signal. The horizontal axis is the magnitude ratio of the two corresponding signals, $\eta = \frac{|x_d(w)|}{|x(w)|}$, and the vertical axis is the signal phase difference, $\Delta \theta(w) = \theta_x(w) - \theta_y(w)$. When the delta signal is a slower signal, $\eta$ is limited between 0.5 and 2, i.e., each corresponding signal’s magnitude cannot go beyond the other’s two times. Also the phase difference is always under $60^\circ$.

In general, the more similar the two corresponding signals are, the slower the delta signal is. When the two corresponding signals are similar, the delta signal contains fewer new frequency components and most of frequency components are stored in the fundamental circuit signals. The FCBCA solves the delta circuit and adds the slower delta signal on the fundamental circuit solution to form the derivative circuit solution. Thus, the FCBCA runs faster. If, however, the derivative signal has little similarity to its corresponding fundamental signal, the delta signal is not a slower signal.

Yet, the above phenomenon does not prevent the FCBCA from being used in circuit verification and modification if the circuit has fairly robust performance. For instance, to check a circuit’s performance, 100 circuit simulations are needed for different parameter changes. Suppose we have 10 bad derivative circuits with their performances highly different from that of the fundamental circuit and 90 good derivative circuits similar to that of the fundamental circuit. Suppose the FCBCA uses $\frac{1}{10}$ of the original circuit simulation time for a good derivative circuit and the same simulation time for a bad derivative circuit. Consequently, for the 100 derivative circuit simulations, the FCBCA needs 190 min. while conventional
device characteristic curve changes from $V_{th}$, and the diode’s characteristic curve changes from $V_{th}$ to $V_{th} - \Delta V$ and $i_{d} = i_{d} - \Delta i$ as we expected. $\Delta G$ in (7) is the linear slope between points A and B. The delta circuit is drawn in Fig. 3(b).

Now we find $\Delta G(t)$ and solve the derivative circuit by the secant algorithm [10] in the following iterations in Fig. 4. Suppose the simulation error tolerance is $\epsilon$.

Step 1.1: Initiate the diode delta model as $G(t) = G_{d}(E(t) - G_{a}v)$ and the common point of (6a), and curve $i = g(t)$.

When the conductor changes from $G_{a}$ to $G_{b}$ and the diode’s characteristic curve changes from $i = g(t)$ to $i = g(t)$, the dash curve in Fig. 4, a derivative circuit forms. The reformed derivative circuit KVL equation is as follows:

$$i = G_{b} E(t) - G_{a} v.$$  

The derivative circuit solution is $R_{b} - G_{a}$, the common point of curve $i = g(t)$ and (6b), as shown in Fig. 4.

Now if we model the diode in the delta circuit as a resistor with conductance

$$\Delta G(t) = \left(\frac{ib}{\Delta t}\right)$$

and (7), the model reinforces the relationship of $v_b = v_a - \Delta V$ and $i_b = i_a - \Delta i$ as we expected. $\Delta G$ in (7) is the linear slope between points A and B. The delta circuit is drawn in Fig. 3(b).

Now we find $\Delta G(t)$ and solve the derivative circuit by the secant algorithm [10] in the following iterations in Fig. 4.

Suppose the simulation error tolerance is $\epsilon$.

Step 1.1: Initiate the diode delta model as $G(t) = G_{d}(E(t) - G_{a}v)$ and simulate the delta circuit. The guessed solution of the derivative circuit is formed by the solutions of the delta circuit and the fundamental circuit at point 1.1 in Fig. 4. Point 1.1 is the intersection of line $i = G_{a} E(t) - G_{a}v$ and $i = g(t)$ tangent line at point A.

Step 1.2: Check error with $g(t)$ as $|\text{Err}| = |g(t_{1.1}) - g(t_{1.1})|/|g(t)|$.

Since $\text{Err} > \epsilon$, modify $\Delta G(t)$; draw the chord from point A to point 1.2. At point 1.2, $v(t_{1.2}) = (t_{1.1}, t_{1.2}) = g(t_{1.1})$, and $\Delta G(t)$ is as the slope of the chord.

Step 2.1: Simulate the delta circuit with the new $\Delta G(t)$ and obtain the guessed solution of the derivative circuit
Step 3.2: Check error as follows:

\[ Err = |y_{2,1} - y_{3,1}|. \]

Since \( Err > \varepsilon \), modify \( \Delta G(t) \); draw the chord from point A to point 2.2. At point 2.2, \((v_{t,2,2}) = (v_{3,2,2}) = y_{3,2,1} \), and \( \Delta G(t) \) is as the slope of the chord.

Step 3.1: Simulate the delta circuit with the new \( \Delta G(t) \) and obtain the guessed solution of the derivative circuit at point 3.1. It is the common point of the line \( \text{from } G_y(t) - G_y(t) \) and the chord of between point 2.2 and point A.

Step 3.2: Check error as follows:

\[ Err = |y_{3,1} - y_{3,1}|. \]

Since \( Err < \varepsilon \), point 3.1 is the final solution. \( v* = v(3.1) \) and \( i* = i(3.1) \) as is shown in Fig. 4.

B. Transistor Circuit with FCBCA

Transistors are the most common nonlinear devices in integrated circuits (ICs). Transistor’s performance can be affected by manufacturing process variations or the ambient working environment. Let us use a simple metal-oxide-semiconductor field-effect transistor (MOSFET) transistor circuit (Fig. 5) as an example to briefly discuss how the FCBCA is used on a transistor circuit.

In Fig. 5, suppose the circuit is altered by both linear and nonlinear components, i.e., the bias resistors are changed from \( R_a \) and \( R_b \) to \( R_a \) and \( R_b \) and transistor parameters \( V_{th} \) and \( I_0 \) from \( V_{th} \) and \( I_0 \) to \( V_{th} \) and \( I_0 \), respectively.

First, we build the delta circuit with Procedure 1 for the changed resistances and initiate the MOSFET transistor delta model with the solution of the fundamental circuit with \( V_{th} \) and \( I_0 \). Then we solve the delta circuit and get the first guessed solution \((V_1, I_1)\) for the altered circuit.

Next, we check if the guessed solution \((V_1, I_1)\) is the real solution. We apply the guessed voltage \( V_1 \) on the SPICE model card with the new parameters, \( V_{th} \) and \( I_0 \), and get \( I_1^* \). The error is calculated as follows:

\[ Err = |I_1 - I_1^*|. \]

If \( Err \) is less than the given error tolerance, \((V_1, I_1)\) is the circuit solution. Otherwise, we rebuild the delta circuit by adjusting the transistor delta model. For example, the new drain-gate capacitance, \( C_{DG} \), in the MOSFET is calculated as follows:

\[ C_{DG} = \frac{Q_{d}(V_{th} - I_0) - Q_{d}(V_{th} - I_1^*)}{V_{th}(V_{th} - I_0) - V_{th}(V_{th} - I_1^*)}. \]

where \( Q_{d}(V_{th}, I_0) \) and \( V_{th}(V_{th} - I_0) \) are the MOSFET channel charge and gate-drain voltage in the fundamental circuit, respectively. \( Q_{d}(V_{th} - I_1^*) \) and \( V_{th}(V_{th} - I_1^*) \) are the MOSFET channel charge and gate-drain voltage, respectively, from the SPICE model card when \((V_1, I_1^*)\) is applied.

Then we solve the new delta circuit and get the new guessed solution, \((V_2, I_2)\), for the changed circuit.

Similarly, apply \( V_2 \) to the SPICE model card with the new parameters, \( V_{th} \) and \( I_0 \), and get \( I_2^* \). The new error is as follows:

\[ Err = |I_2 - I_2^*|. \]

If \( Err \) is less than the given error tolerance, \((V_2, I_2)\) is the circuit solution. Otherwise, rebuild the delta circuit by adjusting the MOSFET transistor delta model again with \((V_1, I_1^*)\). The iteration is continued until the real solution is found.

C. General Nonlinear Device Delta Models

In general, a nonlinear device is defined on an \( x-y \) plane, where \( x \) and \( y \) could be voltage and current, or voltage/current’s derivatives, and so on.

Definition 3: For a device defined on an \( x-y \) plane, the device’s delta model is defined as follows:

\[ \Delta \xi(t) = \frac{\partial \xi}{\partial x} \]

where the device works at \( (x_0(t), y_0(t)) \) in a fundamental circuit and \( (x_3(t), y_3(t)) \) in its derivative circuit.

Procedure 2: For a given nonlinear circuit and simulation error tolerance, \( \varepsilon \), the following procedure is built to find the derivative circuit solution with the FCBCA.

Step 1: Initialize each nonlinear device model as follows:

\[ \Delta \xi(t) = \frac{\partial \xi}{\partial x} \]

where \( (x_0, y_0) \) is the nonlinear device solution in the fundamental circuit.

Step 2: Build the delta circuit by Procedure 1 with the nonlinear device models \( \Delta \xi(t) \), and solve the circuit, \( (\Delta x, \Delta y) \).

Step 3: Guess derivative circuit solution by the following:

\[ x = x_0 - \Delta x \quad \text{and} \quad y = y_0 - \Delta y. \]

Step 4: Check the error as follows:

\[ Err = |y - f(x)| \]

where \( f(.) \) denotes a nonlinear device characteristic curve or nonlinear device SPICE model card when the device is applied in the derivative circuit.

Step 5: If \( Err > \varepsilon \), modify each nonlinear device model as follows:

\[ \Delta \xi(t) = \frac{\partial \xi}{\partial x}. \]

and repeat the process back to Step 2.

Step 6: Otherwise, the derivative circuit solution is as follows:

\[ x_0 = x \quad \text{and} \quad y_0 = y. \]
**IV. Experiments**

Two examples are analyzed with the FCBCA and compared with individual SPICE simulations. When transferring the technique over to real design activities, such as parametric sweeping, a significant amount of efficiency is attained.

### A. Speech Band Amplifier

Fig. 6 is a speech band amplifier circuit. The frequency response of this amplifier covers the range of the human voice, 250 Hz to 2700 Hz. The two ICs used on the board are OPA347. A derivative circuit is built with 30% change on all linear elements on the board, 20% change on $V_{th}$ for all transistors in the ICs, and temperature change from $27 \degree C$ to $0 \degree C$.

In the experiments, a 1 kHz sine wave and 1 kHz square waveform are used as input signals in two separate analyses. Since the sine wave is in the design frequency specification, it induces a linear system. The square wave is beyond the design frequency specification, so it induces a nonlinear system.

Fig. 7(a) and (b) displays the simulation output waveforms from the sine wave input and square wave input, respectively. The two larger waveforms in each graphic are the outputs of the fundamental and the derivative circuits. The two smaller waveforms in each graphic are the waveform from the delta circuit and the difference waveform of the two original circuits.

**TABLE I**

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Number of Analysis Calculated Points</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fundamental</td>
<td>50 000</td>
<td>67.2</td>
</tr>
<tr>
<td>Derivative</td>
<td>48 210</td>
<td>62.1</td>
</tr>
<tr>
<td>Delta</td>
<td>11 860</td>
<td>13.8</td>
</tr>
</tbody>
</table>

Table I shows the simulation performances when the system acts as a nonlinear system. With error tolerance under $10^{-6}$, the delta circuit simulation took two iterations to reach the solution. The simulation time was 22% of its derivative circuit.

### B. Signal Regulator IC

Fig. 8 is a signal regulation circuit. The IC used in Fig. 8 is a signal comparator designed on the LBC7 process. When the input signal is less than the given reference signal, the output of the comparator turns to zero. Otherwise, the output turns to a high voltage. Most of the transistors in the IC work in the transistor nonlinear region.

A derivative circuit is built with a 10% change in the power supply and a 5% change in the reference signal. Running simulation with SPICE model cards from foundry fab, we have the simulation performances of the three circuits in Table II. With the error control under $10^{-6}$, the delta circuit took two iterations to get the solution and totally used about 17% of the derivative circuit simulation time.

When parametric sweeping is applied on this design, 220 circuit simulations are executed. The FCBCA uses 12 min. while independent simulations use 70 min.

**TABLE II**

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Number of Analysis Calculated Points</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fundamental</td>
<td>48 000</td>
<td>27.4</td>
</tr>
<tr>
<td>Derivative</td>
<td>46 910</td>
<td>26.1</td>
</tr>
<tr>
<td>Delta</td>
<td>10 240</td>
<td>4.6</td>
</tr>
</tbody>
</table>

V. Conclusion

An efficient algorithm for computer-aided circuit simulation was presented. The algorithm was implemented with the SPICE simulator and can be applied to linear and nonlinear circuit applications. Examples demonstrated and confirmed the efficiency attained through the algorithm.


