An Efficient Time Step Control Method in Transient Simulation for DAE System

Jian Yao, Zuochang Ye and Yan Wang
Institute of Microelectronics
Tsinghua University
Beijing 100084, China
Email: {yao-j10, zuochang, wangy46}@mails.tsinghua.edu.cn

Tianshi Wang and Jaijeet Roychowdhury
Department of Electrical Engineering and Computer Science
University of California, Berkeley
California 94720, USA
Email: {tianshi, jr}@berkeley.edu

***Abstract*—Adaptive time step control is very important or even crucial in transient simulation for the efficiency of a circuit simulator. Existing methods are mainly based on the formula of the local truncation error (LTE) for solving ordinary differential equations (ODEs), which is an approximation for the circuit simulator solving a system of nonlinear differential algebraic equations (DAEs). In this work, we derived the formula of LTE for DAE system and proposed a new time step control method. Experimental results show that the proposed method works well for industrial circuits.**

I. INTRODUCTION

Modern circuits are typically stiff systems characterized by a wide range of time constants. During a transient simulation, it is desirable to change the step size dynamically by taking dense steps when the circuit is under going fast transitions in order to preserve accuracy and loose steps when there is little activity [1]. Adaptive time step control is therefore very important for the efficiency of the circuit simulator. Such method should change the time step automatically without human intervention, and ensure both the accuracy and efficiency.

In 1972, Gear [2] proposed an adaptive time step control method for solving ordinary differential equations or ODE. After that, a number of improvements [3], [4], [5] have been developed. However, all those methods are essentially based on the derivation of the local truncation error (LTE) for ODE system. In fact, the circuit system should be modeled by vector differential-algebraic equations (DAEs) [6], which is more complicated than ODEs. Therefore, the traditional time step control methods are just an approximation for the circuit simulator.

In this work, we proposed a new time step control method in transient simulation for the circuit simulator described by DAEs. The key of the method is that the formula of LTE for the DAE system has been derived, which can be directly used for time step control in transient simulation. Experimental results show that the proposed method works well for both linear and non-linear circuits. Both accuracy and efficiency have been achieved.

The rest of this paper is organized as follows. In Section II, the relevant background material will be introduced. And then, the proposed time step control method will be presented in Section III. Next, experimental results to validate the proposed method will be discussed in Section IV. Finally, conclusions will be given in Section V.

II. BACKGROUND

In general, the nonlinear circuit system can be modeled as vector differential-algebraic equations (DAEs) [6]:

$$\frac{d}{dt}q(x(t)) + f(x(t)) + b(t) = 0 \tag{1}$$

where $x(t)$ is a vector of node voltages and branch currents; $q()$ and $f()$ are nonlinear functions describing the charge/flux and resistive terms, respectively; $b(t)$ is a vector forcing term consisting of inputs, usually voltage or current sources. For description convenience, equation (1) can be rewrote as:

$$\frac{d}{dt}q(x(t)) = g(x,t) \tag{2}$$

A. Time Integration

Linear Multi-Step (LMS) method is widely used to calculate the numerical solution of transient simulation [7]. In general, LMS can be expressed as:

$$\sum_{i=0}^{p}\alpha_i q(t_{n-i}) = \sum_{i=0}^{p}\beta_i g(x_{n-i}, t_{n-i}) \tag{3}$$

where n is the current time point, p is the order of LMS, α_i and β_i are coefficients and $\alpha_0 \equiv 1$. With the given p, α_i and β_i, the specific LMS method, e.g. Trapezoidal (TRAP) method and Gear methods can be obtained.

B. Truncation Error and Adaptive Time Step Control

Truncation error occurs when the differential term is replaced by a discrete-time approximation, i.e., LMS methods. Local truncation error ε_T or LTE is the error made in a single step assuming all previous steps are accurate. Such LTE can be defined precisely as:

$$\varepsilon_T(t_n) = x_n - x_n^* \qquad x_{n-i} = x_{n-i}^*, i = 1, \ldots, p \tag{4}$$

where x_n and x_n^* are the numerical solution (i.e., solution obtained by LMS method) and exact solution of equation (1) respectively at n^{th} time point.

The most common flow for time step control is trial and error methods [3], [4]. For each time point, the adaptive solver starts with calculating the circuit equation and LTE. If the condition for LTE is not satisfied, the step size will be reduced and the circuit equations will be solved again. This process will be repeated until the LTE condition is satisfied. In [1], a more efficient method is proposed by solving the circuit equations together with the condition for LTE as one nonlinear system.

C. The Limitation of Traditional Methods

Time step control methods are based on calculating LTE to adaptively change the time step. In this way, the key is the formula of LTE used in circuit simulator. Traditional methods are essentially based on the derivation of the LTE for ODE system [2], i.e.,

$$\frac{d}{dt}x(t) + f(x(t)) + b(t) = 0 \tag{5}$$

Compared with Equation (1), ODE is without $q()$ term. Therefore, the traditional time step control methods are just an approximation for the circuit simulator and there are problems in certain cases. So in the following, we will derive the LTE for DAE system and propose a new time step control method.

III. The Proposed Method

In this section, the proposed time-step control method will be presented. First, the formula of LTE for DAEs will be derived in detail in Section III-A. And then, according to the formula of LTE, the time-step control flow will be given in Section III-B.

A. Derivation the LTE formula for DAE system

Assume the exact solution of $q(x(t))$ in equation (1) is $q(x^*(t))$, which can be simply denoted as $q(x^*(t)) \equiv q^*(t)$. We also denote $q(x(t_n)) \equiv q(t_n) \equiv q_n$ for description convenience.

To derive LTE, first, the exact solution $q^*(t)$ should be expanded at the point t_{n-p} using Taylor's formula with Lagrange remainder [8]:

$$\begin{aligned} q^*(t) &= q^*(t_{n-p} + t - t_{n-p}) \\ &= q^*(t_{n-p}) + {q^*_{n-p}}'(t-t_{n-p}) + \tfrac{{q^*_{n-p}}''}{2!}(t-t_{n-p})^2 \\ &\quad + ... + \tfrac{{q^*_{n-p}}^{(k)}}{k!}(t-t_{n-p})^k + \tfrac{q^{*(k+1)}(\zeta)}{(k+1)!}(t-t_{n-p})^{k+1} \\ &\qquad \zeta \in [t_{n-p}, t_n] \end{aligned} \tag{6}$$

Equation (6) can be rewritten into a more convenient form by defining the k^{th} degree polynomial $p_k(t)$:

$$\begin{aligned} p_k(t) &= q^*(t_{n-p}) + {q^*_{n-p}}'(t-t_{n-p}) + \tfrac{{q^*_{n-p}}''}{2!}(t-t_{n-p})^2 \\ &\quad + ... + \tfrac{{q^*_{n-p}}^{(k)}}{k!}(t-t_{n-p})^k \end{aligned} \tag{7}$$

note that k is the maximum order of polynomial that LMS method can obtain the exact solution.

So, Equation (6) can now be written as follows:

$$q^*(t) = p_k(t) + \frac{q^{*(k+1)}(\zeta)}{(k+1)!}(t-t_{n-p})^{k+1}, \zeta \in [t_{n-p}, t_n] \tag{8}$$

According to Equation (2), (3), we have:

$$\sum_{i=0}^{p} \alpha_i q(t_{n-i}) = \sum_{i=0}^{p} \beta_i g(x_{n-i}, t_{n-i}) = \sum_{i=0}^{p} \beta_i \dot{q}_{n-i} \tag{9}$$

Based on Equation (9), a new operator $\mathcal{L}_n\{\}$ is defined as follows:

$$\mathcal{L}_n\{y(t)\} \equiv \sum_{i=0}^{p} [\alpha_i y(t_{n-i}) - \beta_i \dot{y}_{n-i}] \tag{10}$$

this operator is used to convert any time function $y(t)$ into a number according to the LMS method.

Now, we can apply this operator to Equation (8). Note that $\mathcal{L}_n\{\}$ is linear in $y(t)$, so we can obtain the following equation:

$$\mathcal{L}_n\{q^*(t)\} = \mathcal{L}_n\{p_k(t)\} + \mathcal{L}_n\{\frac{q^{*(k+1)}(\zeta)}{(k+1)!}(t-t_{n-p})^{k+1}\} \tag{11}$$

Note that $p_k(t)$ is k^{th}-order polynomial. According to exactness constraints [9], the solution of $p_k(t)$ is the exact solution, so:

$$\mathcal{L}_n\{p_k(t)\} \equiv 0 \tag{12}$$

In this way, we can obtain the following equation:

$$\begin{aligned} \mathcal{L}_n\{q^*(t)\} &= \mathcal{L}_n\{\tfrac{q^{*(k+1)}(\zeta)}{(k+1)!}(t-t_{n-p})^{k+1}\} \\ &= \tfrac{q^{*(k+1)}(\zeta)}{(k+1)!}\mathcal{L}_n\{(t-t_{n-p})^{k+1}\} \end{aligned} \tag{13}$$

Next, we need to find the relationship between $q_n - q_n^*$ and $\mathcal{L}_n\{q^*(t)\}$.

According to Equation (9), q_n can be solved as:

$$q_n - \beta_0 \dot{q}_n + \sum_{i=1}^{p} (a_i q^*_{n-i} - \beta_i \dot{q}^*_{n-i}) = 0 \tag{14}$$

Meanwhile, after applying operator $\mathcal{L}_n\{\}$ to $q^*(t)$, we can obtain:

$$\mathcal{L}_n\{q^*(t)\} = q^*_n - \beta_0 \dot{q}^*_n + \sum_{i=1}^{p} (a_i q^*_{n-i} - \beta_i \dot{q}^*_{n-i}) \tag{15}$$

We can subtract Equation (14) from Equation (15), and obtain the relationship between $q_n - q_n^*$ and $\mathcal{L}_n\{q^*(t)\}$:

$$q_n - q_n^* - \beta_0(\dot{q}_n - \dot{q}_n^*) = \mathcal{L}_n\{q^*(t)\} \tag{16}$$

Next, we need to express $q_n - q_n^*$ and $\dot{q}_n - \dot{q}_n^*$ in terms of LTE ε_T. This should be done in two steps.

Firstly, let us expand $q(x_n)$ at the point x_n^* using Taylor's formula:

$$\begin{aligned} q(x_n) &= q(x_n^*) + \dot{q}_{x_n^*}(x_n - x_n^*) + \text{h.o.t}(x_n - x_n^*) \\ &\approx q(x_n^*) + \dot{q}_{x_n^*}\varepsilon_T \end{aligned} \tag{17}$$

Secondly, we expand $g(x_n, t_n)$ at the point x_n^* using Taylor's formula:

$$\begin{aligned} g(x_n, t_n) &= g(x_n^*, t_n) + \dot{g}_{x_n^*}\varepsilon_T + \text{h.o.t}(\varepsilon_T) \\ &\approx g(x_n^*, t_n) + \dot{g}_{x_n^*}\varepsilon_T \end{aligned} \tag{18}$$

According to Equation (2), we have:

$$\dot{q}_n - \dot{q}_n^* = g(x_n, t_n) - g(x_n^*, t_n) \tag{19}$$

So, according to Equation (16) - (19), we can obtain:

$$\begin{aligned}\dot{q}_{x_n^*}\varepsilon_T - \beta_0\dot{g}_{x_n^*}\varepsilon_T &\approx \mathcal{L}_n\{q^*(t)\} \\ &= \tfrac{q^{*(k+1)}(\zeta)}{(k+1)!}\mathcal{L}_n\{(t-t_{n-p})^{k+1}\}\end{aligned} \tag{20}$$

So now, we can obtain

$$\varepsilon_T = (\dot{q}_{x_n^*} - \beta_0\dot{g}_{x_n^*})^{-1}\frac{q^{*(k+1)}(\zeta)}{(k+1)!}\mathcal{L}_n\{(t-t_{n-p})^{k+1}\} \tag{21}$$

Next, we need to find the formula of $\mathcal{L}_n\{(t-t_{n-p})^{k+1}\}$. According to the operator $\mathcal{L}_n\{\}$, the formula can be easily obtained. So finally, the formula of LTE can be expressed as:

$$\begin{aligned}\varepsilon_T = \sum_{i=0}^{p}&[\alpha_i(t_{n-i}-t_{n-p})^{k+1} - \beta_i(k+1)(t_{n-i}-t_{n-p})^k] \\ &(\dot{q}_{x_n^*} - \beta_0\dot{g}_{x_n^*})^{-1}\tfrac{q^{*(k+1)}(\zeta)}{(k+1)!} \qquad \zeta \in [t_{n-p}, t_n]\end{aligned} \tag{22}$$

Based on Equation (22) and finite difference approximation, we can easily obtain the corresponding formula of LTE for specific LMS methods, which is summarized in Table I.

TABLE I. THE FORMULA OF LTE FOR SPECIFIC LMS METHODS

Method	LTE Formula
TRAP	$\varepsilon_T = -\frac{1}{12}(\dot{q}_{x_n} + 0.5h\dot{f}_{x_n})^{-1}$ $(g(x_n,t_n) - 2g(x_{n-1},t_{n-1}) + g(x_{n-2},t_{n-2}))h$
GEAR1	$\varepsilon_T = -\frac{1}{2}(\dot{q}_{x_n} + h\dot{f}_{x_n})^{-1}(g(x_n,t_n) - g(x_{n-1},t_{n-1}))h$
GEAR2	$\varepsilon_T = -\frac{1}{6}(\dot{q}_{x_n} + \frac{h_1h_2(h_1+h_2)}{h_2(2h_1+h_2)}\dot{f}_{x_n})^{-1}\frac{(h_1+h_2)^2}{(2h_1+h_2)h_2}$ $[h_2g(x_n,t_n) - (h_1+h_2)g(x_{n-1},t_{n-1}) + h_1g(x_{n-2},t_{n-2})]$

*$h = t_n - t_{n-1}$. $h_1 = t_n - t_{n-1}$ and $h_2 = t_{n-1} - t_{n-2}$.

B. LTE-based time step control method

After we obtain the formula of LTE for DAEs, the time step of transient simulation can be changed adaptively according to the LTE value. The flow of the proposed method is shown in Fig. 1. This strategy can be easily explained. If the LTE is smaller than a given specification ($\varepsilon_{\text{spec}}$), the next time step will be increased. Otherwise, the next time step will be reduced. Particularly, if the LTE is too large (larger than $\text{F}_{\text{redo}}\cdot\varepsilon_{\text{spec}}$), the results of the current time step are not reliable. So, the current time step needs to be redone with a smaller time step until the condition of LTE is satisfied.

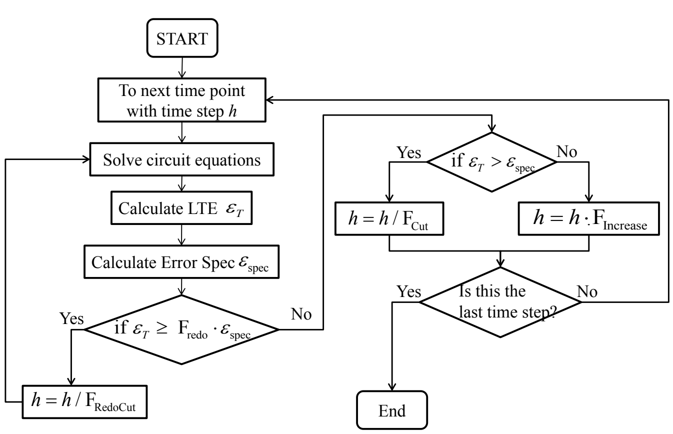


Fig. 1. The flow diagram of the proposed time step control method.

IV. EXPERIMENTAL VALIDATION

In this section, the proposed method will be verified by industrial circuits, including a relaxation oscillator [10], a RC line with 3 segments [11] and a suite of performance benchmarks. Results shown in this section were generated using 2nd-order Gear method (GEAR2), which is the default time integration method for lots of simulators. It must be noted that the proposed method also works well for other commonly used time integration formula such as Backward Euler method (i.e, GEAR1) and Trapezoidal method. The accuracy reference is the analytical solution or numerical solution with extremely dense time-step.

A. Example 1: Relaxation Oscillator

In the first example, we simulate a Schmitt Trigger Relaxation Oscillator. Besides the proposed adaptive time-step control method, the uniformly dense and loose time-step strategy are also used for comparison. The simulation results are shown in Fig. 2.

The Relaxation Oscillator is a typical non-linear circuit characterized by a wide range of time constants. It means that in some regions, the time-step should be big to ensure the efficiency and in other regions, the time-step should be small for accuracy requirement. So the uniform time-step strategy is always not satisfied. As shown in Fig. 2(a), the accuracy of dense step is acceptable, but the time-step is too dense in both 0.03s-0.04s and 0.06s-0.07s. On the contrary, when the time-step is selected loosely, the accuracy cannot be ensured. As shown in Fig. 2(b)(Zoom in), the time-step is too loose around 0.047s and the period of the oscillator isn't right (Fig. 2(b)(Zoom out)). As the oscillating period is a key parameter for the relaxation oscillator, such simulation result cannot be accepted.

Fig. 2(c) shows the simulation result of the proposed method. We can find that the time step is adaptively changed. Both accuracy and efficiency have been achieved by the proposed time-step control method.

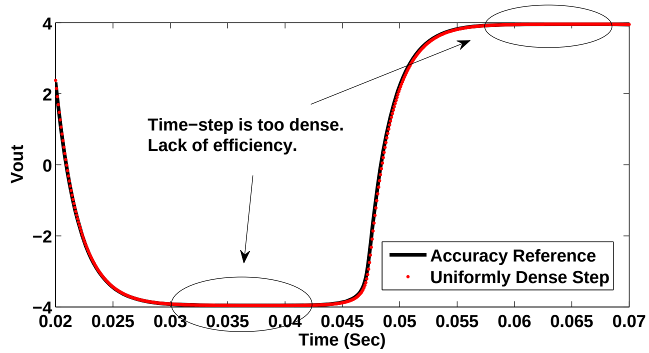
B. Example 2: RC Line with 3 Segments

In this example, we use a linear example, RC Line with 3 segments, to show the efficiency of the proposed method. The Fig. 3(a) is the simulation result based on uniform time-step. As the error will be accumulated, we can conclude that the accuracy is not satisfied. The reason is that in each peak and valley of Vout (i.e., fast transitions), the number of time-steps is not enough.

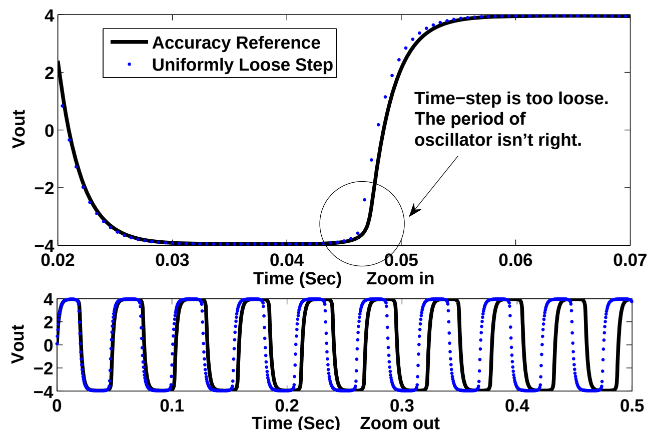
We further run the circuit using the proposed time-step control method, which is shown in Fig. 3(b). More time points are added adaptively when the circuit is under going fast transitions. In this way, the accuracy is satisfied over the whole time points.

C. Performance Benchmarks

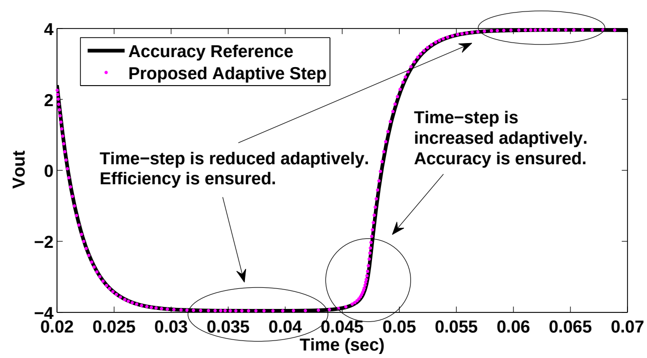
We simulate a performance benchmark suite of 12 industrial circuits listed in Table II. Similar as Example 1 and Example 2, the proposed method works well for these test circuits. Both accuracy and efficiency can be achieved. In this way, the robustness of the proposed time-step control method can be verified.



(a) Uniformly dense step.



(b) Uniformly loose step.



(c) Proposed adaptive step.

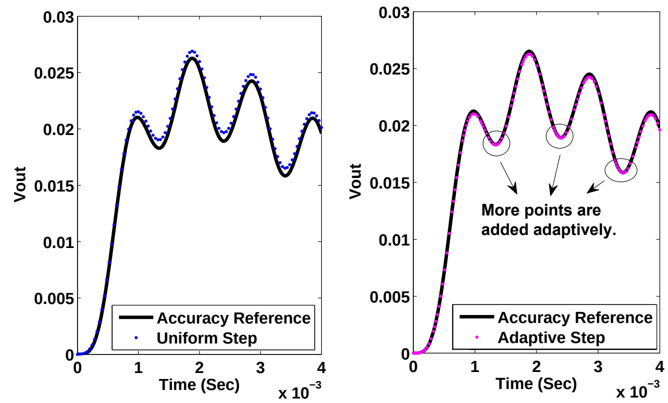
Fig. 2. Transient simulation of Relaxation Oscillator with different strategies.

V. Conclusion

In this work, we proposed a new time step control method for circuit simulators described as a system of DAEs. The formula of LTE for the DAE system has been derived. Experimental results show that the proposed method works well for industrial circuits. Both accuracy and efficiency have been achieved. The proposed method is especially suitable for stiff system characterized by a wide range of time constants. Meanwhile, such method can also be used in transient simulation of other domains, e.g., mechanics and biology.

Acknowledgment

This author's work was supported by the Chinese Scholarship Council during his visit to Department of EECS, University of California, Berkeley.



(a) Uniform time-step. (b) Proposed adaptive time-step.

Fig. 3. Transient simulation of RC Line circuit with 3 segments.

TABLE II. The Test Circuit Used In This Work

Benchmark circuits	Benchmark circuits
Charge Pump	Delay Line
Ring Oscillator	Parallel RLC Diode Circuit
Parallel LC Circuit	Parallel LRC Circuit
Differential Pair Circuit	Inverter
Resistive Divider	Two Reaction Chain
Inverter Chain	RC Line

References

[1] G. Fang, "A new time-stepping method for circuit simulation," in *IEEE Design Automation Conference (DAC)*, May 2013, pp. 1–10.

[2] C. W. Gear and C. W. Gear, *Numerical initial value problems in ordinary differential equations*. Prentice-Hall Englewood Cliffs, NJ, 1971, vol. 59.

[3] K. Gustafsson, "Control-theoretic techniques for stepsize selection in implicit runge-kutta methods," *ACM Transactions on Mathematical Software (TOMS)*, vol. 20, no. 4, pp. 496–517, 1994.

[4] G. Söderlind and L. Wang, "Adaptive time-stepping and computational stability," *Journal of Computational and Applied Mathematics*, vol. 185, no. 2, pp. 225–243, 2006.

[5] A. G. Gheorghe, F. Constantinescu, and M. Nitescu, "Error control in circuit transient analysis," in *Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th IEEE International Conference on*, Dec 2009, pp. 207–210.

[6] O. Narayan and J. Roychowdhury, "Multi-time simulation of voltage-controlled oscillators," in *Design Automation Conference, 1999. Proceedings. 36th*, 1999, pp. 629–634.

[7] C. T. Baker, G. Monegato, and G. V. Berghe, *Ordinary Differential Equations and Integral Equations*. Gulf Professional Publishing, 2001, vol. 6.

[8] J. A. Riestra, *A generalized Taylor's formula for functions of several variables and certain of its applications*. CRC Press, 1995, vol. 329.

[9] M. Crow, *Computational methods for electric power systems*. Crc Press, 2002.

[10] A. Srinivasulu, "A novel current conveyor-based schmitt trigger and its application as a relaxation oscillator," *International Journal of Circuit Theory and Applications*, vol. 39, no. 6, pp. 679–686, 2010.

[11] P. Penfield Jr and J. Rubinstein, "Signal delay in rc tree networks," in *Proceedings of the 18th design automation conference*. IEEE Press, 1981, pp. 613–617.