A Fast Time-Step Selection Method for Explicit Solver-Based Simulation of High Frequency Low Loss Circuit and Its Application on EMI Filter

Zhen Huang, *Student Member, IEEE*, Ruimin Zhu, *Member, IEEE*, Chengcheng Tang, *Student Member, IEEE*, and Venkata Dinavahi, *Fellow, IEEE*

***Abstract*—Electromagnetic interference (EMI) modeling and prediction are essential for the design of most power electronics apparatuses. This article aims at finding a fast method to select time step for explicit solver-based simulation of high frequency low loss (HFLL) circuits like EMI filter. The state-space model of HFLL circuit is constructed and its eigenvalues are proved to be very close to the imaginary axis. Both the nondegenerate and degenerate circuit cases are discussed. During the analysis, a circuit lemma is summarized on how to transform degenerate circuit into nondegenerate circuit and the corresponding inversion of its coefficient matrix is derived based on Sherman–Morrison's formula. Then the Laguerre–Samuelson's inequality is employed to find the upper bound of HFLL circuit's eigenvalues. This process only requires two matrix multiplications and traces of the matrix operation results, thus keeping the computational complexity retaining in $O(N^2)$. A typical EMI filter is constructed and its equivalent circuit including the parasitic effects is extracted from ANSYS. This filter is simulated in application between a dc/ac converter and the grid using the fourth-order Runge–Kutta (RK4) solver with a time step selected by the proposed method. Numerical test shows that the spectrum results are very close to those obtained by experiment while being much more efficient than traditional methods, which demonstrates that this time-step selection method could benefit the analysis and time-domain simulation of HFLL circuits.**

***Index Terms*—Circuit simulation, eigenvalues and eigenfunctions, electromagnetic interference, numerical stability, state space methods.**

I. Introduction

TIME-DOMAIN simulation of electromagnetic interference (EMI) circuit model in power electronics has received more and more attention in recent years [1]–[3]. Compared with frequency-domain simulation, time-domain simulation is more straightforward in terms of performance evaluation but suffer from longer execution time because of much higher amount of computations needed [4]. Therefore, if possible, the circuit simulation users always want to use larger time step for efficient analysis and design, which makes time-step selection one of the most critical procedures in time-domain simulation. However, the time step cannot be set arbitrarily large. There are two constraints that determine the upper bound of the time step. One is that the simulation results should meet some accuracy requirement. For example, if the 40th harmonic of a current/voltage is to be analyzed, then the time step should be at most half the reciprocal of 40 times the base (modulation) frequency. The other constraint is the numerical stability of the solver adopted in computation. This is especially important to explicit solvers because they are always conditionally stable.

Numerical stability is requisite for digital simulation to ensure the results do not diverge during the computation process. It's a property requires that the errors in one step do not grow along the succeeding steps [5]. The numerical stability of a given problem is determined by two factors: the numerical solver, and the simulation model. There are two types of numerical solvers that are available for circuit simulation: Implicit and explicit solver. The implicit solvers usually have large stability region in the complex plane. Some even include the whole left half plane, which means they are always numerically stable as long as the simulation model itself is stable in continuous domain (real parts of all eigenvalues are negative). Examples of implicit solver include the backward Euler (BE) and the trapezoidal rule (TR). However, BE and TR methods are deemed to be inefficient when dealing with circuits that contain high-frequency oscillatory phenomenon [6]. Inefficiency comes from the fact that matrix inversion or its equivalent alternative (like Gaussian elimination or some iterative algorithms) is inevitable when utilizing the implicit solver. The computational complexity of such process increases as a cubic function $O(N^3)$ of the system size. Moreover, the implicit solver is not suitable for parallel computation, which is the trend of future high performance computing.

The explicit solvers, on the other hand, usually have smaller stability regions in the complex plane. But they can get rid of matrix inversion and are very suitable for parallel computation. The corresponding computational complexity increases only as quadratic function $O(N^2)$ of the system size and can be

Manuscript received March 31, 2020; revised May 24, 2020; accepted July 21, 2020. Date of publication July 27, 2020; date of current version September 22, 2020. This work was supported in part by the Natural Science and Engineering Research Council of Canada (NSERC) and in part by the China Scholarship Council (CSC). Recommended for publication by Associate Editor F. Luo. *(Corresponding author: Ruimin Zhu.)*

Zhen Huang, Chengcheng Tang, and Venkata Dinavahi are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2R3, Canada (e-mail: zh2@ualberta.ca; ctang8@ualberta.ca; dinavahi@ualberta.ca).

Ruimin Zhu is with the School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China, and also with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta T6G 2R3, Canada (e-mail: zruimin@ualberta.ca).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPEL.2020.3011948

apportioned on distributed computational resources. It should be noted that the numerical stability of a digital simulation problem is assessed by the numerical solver selected and the simulation model together [7], so the product ($z = h\lambda$) is chosen for stability evaluation, where h is the time step and λ is the eigenvalue of the model. The difficulty of applying explicit solver is how to find the proper time-step h so that every z of the simulation model lies within the stability region of the solver.

It is worth mentioning that although explicit solvers generally have smaller stability regions, it does not mean they are less accurate than implicit solvers. The accuracy of a solver is usually quantified by local truncation error (LTE) and it only relates to the numerical order of that solver. In general, a method with $O(h^{n+1})$ LTE is said to be of nth order. For example, the Heun's method (explicit) and Tustin method (implicit) offer the same accuracy results at the same time step as long as they are numerical stable because they are both order-2 methods and their LTEs are in $O(h^3)$.

Although the stability region of a given solver in the complex z plane can be found beforehand, the eigenvalues of the circuit model may distribute randomly. To determine the suitable time step, all the eigenvalues have to be found because usually the stability region has different lengths (from the edge to the origin) along different directions. However, the computational complex of eigenvalue problem is also $O(N^3)$. That creates a dilemma: If a $O(N^3)$ problem has to be solved before using explicit solver, why not just use implicit solver directly? Not to mention the effort that has to made to construct the state-space model of the circuit.

The time-step selection process of simulating a general circuit using explicit solver is in no way easy. In many cases this is done by estimation, then trial and error iteration, which is very time consuming. Some variable time-step methods, like the Runge–Kutta–Fehlberg (RKF45) method [8], estimate the numerical error by comparing two different order (fourth and fifth order Runge–Kutta) solutions and adaptively modify the time step according to the comparison results [9]. However, the RKF45 method still possesses the risk of generating inappropriate time step and producing fundamentally incorrect solutions, especially when it is applied to solve stiff ordinary differential equations (ODEs) [10]. Failure of such time-step control scheme can be explained by the lack of taking the information from circuit itself into consideration.

Nevertheless, for some special classes of circuits, the time-step selection process can be expedited by leveraging the circuit properties. The high-frequency low loss (HFLL) circuit is one of such type. High frequency means there are oscillatory phenomenon in the circuit. Typically, the influences of parasitic capacitance and inductance have to be considered so that small time step has to be utilized. Low loss in this article means the energy dissipated in the circuit is negligible so that the eigenvalues of the circuit are very close to the imaginary axis. In such cases, only the maximum or upper bound of the eigenvalues needs to be found to determine the suitable time step.

HFLL circuit is not rare in electrical application. The EMI filter is one typical example. It is commonly equipped in power converting applications like motor drive systems, renewable energy generations, electric vehicle charging infrastructures, and modern more electric aircrafts, etc [11]. There are many papers in literature discussing derivation of the equivalent circuit model of EMI network [12]–[15]. These models are obtained either by impedance measurement and then parameter extraction or by some advanced modeling methodologies like finite element method and partial element equivalent circuit (PEEC) method. Few of them pay attention to the time-domain simulation of the equivalent circuit model, which is not easy for the existing simulation tools. For example, the time needed for a 55-μs transient simulation of a multiconductor cable EMI equivalent circuit using SPICE is 3.6 s on a standard PC, as reported in [16], that is almost 65 thousand times slower than real time.

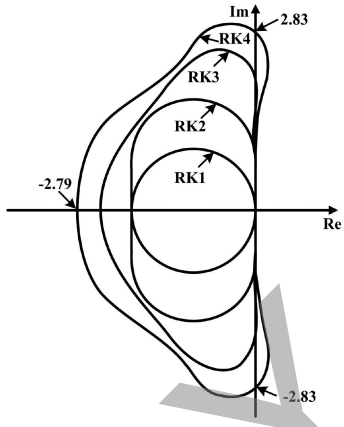


Fig. 1. Stability regions of first to fourth-order Runge–Kutta methods in z plane.

This article presents a quick method to select time step for explicit solver-based simulation of HFLL circuit. The proposed method is easy to implement and remains $O(N^2)$ in computation. Such property preserves the efficiency superiority of explicit solver over implicit solver. The validity of this method is verified by time-domain simulation of an EMI filter in a dc/ac converter compared with experimental results and its strength in efficiency is confirmed by comparison with the same simulation conducted in commercial software.

II. Stability Region of Explicit Solver

Stability region is an important property of a numerical solver. It is evaluated by applying the solver to the Dahlquist test equation

$$\dot{x} = \lambda x, x(0) = x_0 \tag{1}$$

where $\dot{x}$ means derivative of x, λ is eigenvalue of the problem, and x_0 is the initial value. The stability region is quantified in the complex z plane ($z = \lambda h$), where h is the time step.

The region that makes $x(k)$ converging to its analytic solution as $k \to \infty$ is called stability region, where $x(k)$ represents the kth step solution. Fig. 1 illustrates the stability regions of first to fourth-order Runge–Kutta (RK) methods.

In actual applications, it is more common to solve the multivariable state-space equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \mathbf{x}(0) = \mathbf{x}_0 \tag{2}$$

where $\mathbf{x}$ is state vector and $\mathbf{A}$ is system matrix. It may be obscure how the numerical stability results derived from (1) can be applied to analyze (2). A simple matrix transformation can make it clear.

If $\mathbf{A}$ is diagonalizable (which is often the case in engineering problems), then there exists a matrix $\mathbf{P}$ that satisfies

$$\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1} \tag{3}$$

where $\mathbf{\Lambda}$ is a diagonal matrix. It also can be proved that the diagonals of $\mathbf{\Lambda}$ are the eigenvalues of $\mathbf{A}$ and the columns of $\mathbf{P}$ are the corresponding eigenvectors. If $\mathbf{y} = \mathbf{P}^{-1}\mathbf{x}$ is selected as the new state vector, then (2) can be transformed into

$$\dot{\mathbf{y}} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}\mathbf{y} = \mathbf{\Lambda}\mathbf{y}. \tag{4}$$

Solving (4) is like solving n individual Dahlquist equations, where n is the dimension of $\mathbf{y}$. Once $\mathbf{y}$ is obtained, $\mathbf{x}$ can be obtained by simple linear transformation.

The above analysis implies that as long as all the eigenvalues of $\mathbf{A}$ times time-step h locate in the stability region of a given solver, then it is numerically safe to use that solver in simulation. Generally speaking, it is difficult to find the complete set of eigenvalues of matrix $\mathbf{A}$. However, for HFLL circuit, its eigenvalues are very close to the imaginary axis, which makes only the maximum eigenvalue matters in quantifying the time step.

This article chooses the classical fourth-order Runge–Kutta (RK4) method as the solver because it offers a good balance between accuracy and computation effort. It is the highest order method in RK family that uses the same number of function evaluations as with the order of accuracy at each time step. Higher order RK methods need much more computation efforts but do not necessarily bring better accuracy results or significant increase in stability region. Such properties make RK4 the preferred choice for many scientific users on numerical integration methods [17].

The stability region of RK4 method can be found in Fig. 1. Its intersections with imaginary axis are $\pm j2.83$. Therefore, the time step that is used for solving HFLL circuit should satisfy

$$h \leq \frac{2.83}{\bar{\lambda}} \tag{5}$$

where $\bar{\lambda}$ is the upper bound of eigenvalues of HFLL circuit.

III. Finding the Upper Bound of Eigenvalues of HFLL Circuit

There are various ways to find the upper bound of eigenvalues of a given matrix. The most straightforward way must be finding all the eigenvalues first, then the maximum eigenvalue can be identified. This is definitely not an efficient choice because most information are wasted and only the maximum value is of concern. There are also some iterative algorithms available that can be used to find the maximum eigenvalue, like the power iteration, the Lanczos algorithm and the Arnoldi algorithm. However, these algorithms are either prone to convergence problem or subject to some restrictions. Most importantly, they are not computationally efficient as they require $O(N^2)$ calculations in each iteration. In the worst case, N iterations are needed to finish the process. Therefore, they are still $O(N^3)$ algorithms in complexity.

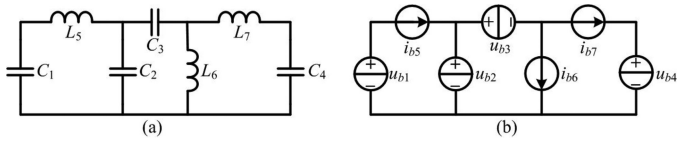


Fig. 2. Nondegenerate circuit consisting of only capacitors and inductors.

Unlike the above methods, this article utilizes Laguerre–Samuelson's inequality to find the upper bound of eigenvalues of HFLL circuit. This inequality can be expressed as the following [18].

Let $\sum_{k=0}^{n} c_k s^k$ be a polynomial with all real roots. Then, all roots of this polynomial are bounded by

$$-\frac{c_{n-1}}{nc_n} \pm \frac{n-1}{nc_n}\sqrt{c_{n-1}^2 - \frac{2n}{n-1}c_n c_{n-2}}. \tag{6}$$

It is known that the eigenvalues of a matrix are the roots of its characteristic polynomial. When applying this inequality to the characteristic polynomial of a matrix, the above bounds can be even simplified because $c_n = 1$. Then only two scalar values, c_{n-1} and c_{n-2}, need to be computed to find the upper bound. The following two subsections explain how to compute these two scalar values for nondegenerate and degenerate HFLL circuits' state-space system matrix, respectively. A circuit is called degenerate when there exists at least one loop in the circuit formed by only capacitive and voltage source type branches, or cutset formed by only inductive and current source type branches [19]. Nondegenerate circuits are those do not contain such loops and cutsets. The handling of these two cases are slightly different.

A. Nondegenerate Circuit Case

As indicated previously, the energy dissipation in HFLL circuit is negligible, which means the values of resistors in the circuit are either very large or very small so that the current or voltage of the resistor can be neglected. In the first step, these resistors are removed from the circuit to expedite the analysis. The large value resistor branch is viewed as open circuit while the small value resistor branch is viewed as short circuit. After removing these resistors, the remaining part contains only capacitors and inductors. To proceed with the analysis, it is necessary to show that a circuit consisting of only capacitors and inductors has all its eigenvalues on the imaginary axis.

It is proved in [20] that for a given circuit that has b branches and n nodes, there exist $n-1$ branches to be viewed as voltage sources and the remaining $b-n+1$ branches to be viewed as current sources, so that the voltage/current relationship of these branches can be expressed in form of an antisymmetric matrix.

For a nondegenerate circuit consisting of only capacitors and inductors, the capacitors are suitable to be viewed as voltage sources while the inductors are suitable to be reviewed as current sources. Take the circuit in Fig. 2 as an example. If the branches of C_1, C_2, C_3, C_4 are viewed as voltage sources and the branches of L_5, L_6, L_7 are viewed as current sources, then

their voltage/current relationship can be express as

$$\begin{bmatrix} i_{b1} \\ i_{b2} \\ i_{b3} \\ i_{b4} \\ u_{b5} \\ u_{b6} \\ u_{b7} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{R}} \begin{bmatrix} u_{b1} \\ u_{b2} \\ u_{b3} \\ u_{b4} \\ i_{b5} \\ i_{b6} \\ i_{b7} \end{bmatrix}. \tag{7}$$

As can be seen, the relation matrix $\mathbf{R}$ in (7) is an antisymmetric matrix. More importantly, simply replacing the left-hand-side vector in (7) with the derivative of right-hand-side vector times the corresponding capacitor or inductor values, the state-space model of this circuit can be obtained, as shown in

$$\begin{bmatrix} C_1 \frac{du_{b1}}{dt} \\ C_2 \frac{du_{b2}}{dt} \\ C_3 \frac{du_{b3}}{dt} \\ C_4 \frac{du_{b4}}{dt} \\ L_5 \frac{di_{b5}}{dt} \\ L_6 \frac{di_{b6}}{dt} \\ L_7 \frac{di_{b7}}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{b1} \\ u_{b2} \\ u_{b3} \\ u_{b4} \\ i_{b5} \\ i_{b6} \\ i_{b7} \end{bmatrix}. \tag{8}$$

It is known that the eigenvalues of an antisymmetric matrix are either 0 or purely imaginary, i.e., they all locate on the imaginary axis. Although the state-space system matrix ($\mathbf{A}$) of a nondegenerate pure capacitor and inductor circuit is not an antisymmetric matrix, it can be expressed as the product of a diagonal matrix ($\mathbf{D}$) and an antisymmetric matrix ($\mathbf{R}$)

$$\mathbf{A} = \mathbf{DR} \tag{9}$$

where the diagonals of $\mathbf{D}$ are the reciprocals of the capacitor and inductor values. For the model in (8)

$$\mathbf{D} = \begin{bmatrix} \frac{1}{C_1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{C_2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{C_3} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{C_4} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{L_5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{L_6} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{L_7} \end{bmatrix}. \tag{10}$$

The characteristic polynomial of $\mathbf{A}$ can be expressed as

$$\begin{aligned} \det(\lambda \mathbf{I} - \mathbf{A}) &= \det(\lambda \mathbf{I} - \mathbf{DR}) \\ &= \det(\mathbf{D}^{\frac{1}{2}}(\lambda \mathbf{I} - \mathbf{D}^{\frac{1}{2}}\mathbf{R}\mathbf{D}^{\frac{1}{2}})\mathbf{D}^{-\frac{1}{2}}) \end{aligned} \tag{11}$$

where $\mathbf{I}$ is identity matrix, $\mathbf{D}^{\frac{1}{2}}$ and $\mathbf{D}^{-\frac{1}{2}}$ are square roots of $\mathbf{D}$ and $\mathbf{D}^{-1}$, respectively. Equation (11) shows that the eigenvalues of $\mathbf{A}$ are the same with $\mathbf{D}^{\frac{1}{2}}\mathbf{R}\mathbf{D}^{\frac{1}{2}}$. On the other hand, $\mathbf{D}^{\frac{1}{2}}\mathbf{R}\mathbf{D}^{\frac{1}{2}}$ is an antisymmetric matrix with all its eigenvalues on the imaginary axis because $\mathbf{D}^{\frac{1}{2}}$ is a diagonal matrix. Therefore, the eigenvalues of $\mathbf{A}$ all locate on the imaginary axis, as well.

Another issue that has to be aware of is that the Laguerre–Samuelson's inequality is valid only when the roots of the polynomial are all real. Thus it can not be applied to the characteristic polynomial of $\mathbf{A}$ directly. To fix this, recalling that the imaginary parts of eigenvalues of a real matrix always appear in conjugate pairs, thus it is more appropriate to calculate the square of eigenvalues of $\mathbf{A}$ to make the roots of the characteristics polynomial all real. Doing so is equal to calculating the eigenvalues of $\mathbf{A}^2$ (denoted as $\bar{\mathbf{A}}$ henceforth). Then the question is transformed into computing coefficients (c_{n-1} and c_{n-2}) of characteristic polynomial of $\bar{\mathbf{A}}$ accordingly.

Factoring $\bar{\mathbf{A}}$'s characteristic polynomial as follows:

$$\sum_{k=0}^{n} c_k s^k = \coprod_{i=1}^{n} (s - s_i) = s^n - \sum_{i=1}^{n} s_i s^{n-1} + \cdots \tag{12}$$

where s_i are the roots of the polynomial. Because the sum of eigenvalues of a matrix is equal to the trace of it. It can be easily found that

$$c_{n-1} = -\sum_{i=1}^{n} s_i = -\mathrm{trace}(\bar{\mathbf{A}}) = -\sum_{i=1}^{n} \bar{a}_{ii} \tag{13}$$

where $\bar{a}_{ii}$ are the diagonals of $\bar{\mathbf{A}}$.

The computation of c_{n-2}, however, is not that explicit. One possible way is to find its relationship with the traces of powers of $\bar{\mathbf{A}}$ is by leveraging Newtons identities [21]. $\bar{\mathbf{A}}^k$, then the following formula holds:

$$\begin{aligned} &t_k + c_{n-1}t_{k-1} + \cdots + c_0 t_{k-n} = 0, (k > n) \\ &t_k + c_{n-1}t_{k-1} + \cdots + c_{n-k+1}t_1 = -k c_{n-k}, (1 \le k \le n). \end{aligned} \tag{14}$$

Setting $k = 1$ and 2 in (14), respectively, the expression of c_{n-2} can be found by

$$c_{n-2} = \frac{1}{2}\left(t_1^2 - t_2\right). \tag{15}$$

Combining (6), (13), and (15), the upper bound (in absolute value) of eigenvalues of $\bar{\mathbf{A}}$ can be written as

$$\bar{s} = \frac{t_1}{n} - \sqrt{n-1}\sqrt{\frac{t_2}{n} - \left(\frac{t_1}{n}\right)^2}. \tag{16}$$

Note that $\bar{s} < 0$ because the eigenvalues of $\bar{\mathbf{A}}$ are all negative. The expression (16) is exactly the same with the bound presented in [22] and is claimed as the "tightest" when all the eigenvalues are real.

The corresponding upper bound of eigenvalues of $\mathbf{A}$ is

$$\bar{\lambda} = \sqrt{|\bar{s}|} = \sqrt{-\frac{t_1}{n} + \sqrt{n-1}\sqrt{\frac{t_2}{n} - \left(\frac{t_1}{n}\right)^2}}. \tag{17}$$

As can be seen that only two matrix multiplications (one from $\mathbf{A}$ to $\bar{\mathbf{A}}$ and one from $\bar{\mathbf{A}}$ to $\bar{\mathbf{A}}^2$) is required to obtain the upper bound of the eigenvalues of a HFLL circuit using this method. Besides, the second multiplication (from $\bar{\mathbf{A}}$ to $\bar{\mathbf{A}}^2$)

is not necessarily to be accomplished completely because only the diagonals are of concern. These mathematic operations still remain $O(N^2)$ in computation complexity. Thus it preserves the computation efficiency advantage of explicit solver.

In addition, because of the antisymmetry of $\mathbf{R}$, the calculation of $\mathbf{A}^2$ can be further simplified. The matrix $\mathbf{R}$ and $\mathbf{D}$ can be partitioned into the following form:

$$\mathbf{R} = \begin{bmatrix} \mathbf{0} & \mathbf{R_{12}} \\ -\mathbf{R'_{12}} & \mathbf{0} \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{D_{11}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_{22}}, \end{bmatrix} \tag{18}$$

where $\mathbf{R}'_{12}$ is the transpose of $\mathbf{R}_{12}$. Then $\bar{\mathbf{A}}$ and $\bar{\mathbf{A}}^2$ can be expressed as

$$\bar{\mathbf{A}} = \begin{bmatrix} -\mathbf{D}_{11}\mathbf{R}_{12}\mathbf{D}_{22}\mathbf{R}'_{12} & \mathbf{0} \\ \mathbf{0} & -\mathbf{D}_{22}\mathbf{R}'_{12}\mathbf{D}_{11}\mathbf{R}_{12} \end{bmatrix} \tag{19}$$

$$\bar{\mathbf{A}}^2 = \begin{bmatrix} (\mathbf{D}_{11}\mathbf{R}_{12}\mathbf{D}_{22}\mathbf{R}'_{12})^2 & \mathbf{0} \\ \mathbf{0} & (\mathbf{D}_{22}\mathbf{R}'_{12}\mathbf{D}_{11}\mathbf{R}_{12})^2 \end{bmatrix}. \tag{20}$$

It also can be proved that the traces of $\mathbf{D}_{11}\mathbf{R}_{12}\mathbf{D}_{22}\mathbf{R}'_{12}$ and $\mathbf{D}_{22}\mathbf{R}'_{12}\mathbf{D}_{11}\mathbf{R}_{12}$ are the same and so are the traces of their square. Thus the dimension of matrix multiplication can be reduced by half (that one quarter in mathematic calculation amount).

It is interesting to note that there is a very concise formula for calculating t_1 of $\bar{\mathbf{A}}$

$$t_1 = -\sum_{r_{ij}\neq 0} d_i d_j \tag{21}$$

where r_{ij} are the entries of $\mathbf{R}$, and d_i and d_j are the diagonals of $\mathbf{D}$. For the example in Fig. 2

$$t_1 = -2\left(\frac{1}{L_5C_1} + \frac{1}{L_5C_2} + \frac{1}{L_6C_2} + \frac{1}{L_7C_2} + \frac{1}{L_6C_3} + \frac{1}{L_7C_3} + \frac{1}{L_7C_4}\right). \tag{22}$$

In cases when only very rough estimation is needed, one can simply assume c_{n-2} in (6) is 0, then the upper bound can be selected as

$$\bar{\lambda} = \sqrt{c_{n-1}} = \sqrt{-t_1} \tag{23}$$

which is very convenient to compute.

B. Degenerate Circuit Case

Although nondegenerate case has covered a lot of circuits in engineering application, there are still circumstances when degenerate circuit may occur, especially in three-phase power systems. Fig. 3(a) is a typical example of degenerate circuit in which C_1, C_2, and C_3 form a loop while L_4, L_5, and L_6 connect to a common node. In such circumstances, if all capacitors are simply viewed as voltage sources and all inductors as current sources, as shown in Fig. 3(b), their voltage-current relation can no longer be represented by an antisymmetric matrix $\mathbf{R}$ because of the singularity of the intermediary matrix during derivation [20].

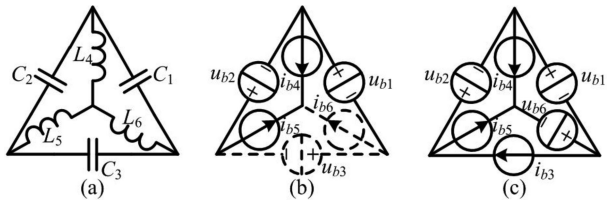


Fig. 3. Degenerate circuit consisting of only capacitors and inductors.

To fix this problem, one branch of the capacitor loop should be viewed as current source and one branch of the inductor cutset should be viewed as voltage source, as illustrated in Fig. 3(c). Then their voltage/current relation can be developed following the same routine as for nondegenerate case, as shown in

$$\begin{bmatrix} i_{b1} \\ i_{b2} \\ u_{b3} \\ u_{b4} \\ u_{b5} \\ i_{b6} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \boxed{1} & -1 & -1 & 0 \\ 0 & 0 & \boxed{1} & 0 & -1 & 0 \\ \boxed{-1} & \boxed{-1} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & \boxed{1} \\ 1 & 1 & 0 & 0 & 0 & \boxed{1} \\ 0 & 0 & 0 & \boxed{-1} & \boxed{-1} & 0 \end{bmatrix} \begin{bmatrix} u_{b1} \\ u_{b2} \\ i_{b3} \\ i_{b4} \\ i_{b5} \\ u_{b6} \end{bmatrix}. \tag{24}$$

Special attention should be given to the entries that are boxed. For the branches that are altered types (C_3, L_6), their voltage/current are the opposite of summation of all remaining branches' voltage/currents in the loop/cutset that make the circuit degenerate. Thus the entries (r_{ij}) in the rows that corresponding to these branches' voltage/current relations are either 1 or 0. If branch i and branch j are in the same degenerate loop/cutset, then $r_{ij} = -1$, otherwise $r_{ij} = 0$. In addition, because of the antisymmetry of matrix $\mathbf{R}$, every 1 entry in these rows has an opposite entry at the symmetric location in matrix $\mathbf{R}$. Just as the boxed entries are shown in (24).

When writing state-space equation of the degenerate circuit, the branches that are altered types cannot be viewed as independent branches because their states can be represented by other branch's states. For the example in Fig. 3

$$\begin{cases} i_{b3} = C_3\dfrac{du_{b3}}{dt} = C_3\dfrac{d(-u_{b1}-u_{b2})}{dt} = C_3\left(-\dfrac{du_{b1}}{dt} - \dfrac{du_{b2}}{dt}\right) \\ u_{b6} = L_6\dfrac{di_{b6}}{dt} = L_6\dfrac{d(-i_{b4}-i_{b5})}{dt} = L_6\left(-\dfrac{di_{b4}}{dt} - \dfrac{di_{b5}}{dt}\right) \end{cases} \tag{25}$$

while the other state equation can be written as usual

$$\begin{cases} C_1\dfrac{du_{b1}}{dt} = i_{b3} - i_{b4} - i_{b5} \\ C_2\dfrac{du_{b2}}{dt} = i_{b3} - i_{b5} \\ L_4\dfrac{di_{b4}}{dt} = u_{b1} + u_{b6} \\ L_5\dfrac{di_{b5}}{dt} = u_{b1} + u_{b2} + u_{b6}. \end{cases} \tag{26}$$

Substituting (25) into (26), the state-space representation of Fig. 3 can be obtained

$$\underbrace{\begin{bmatrix} C_1+C_3 & C_3 & 0 & 0 \\ C_3 & C_2+C_3 & 0 & 0 \\ 0 & 0 & L_4+L_6 & L_6 \\ 0 & 0 & L_6 & L_5+L_6 \end{bmatrix}}_{=\mathbf{M}} \begin{bmatrix} \frac{du_{b1}}{dt} \\ \frac{du_{b2}}{dt} \\ \frac{di_{b4}}{dt} \\ \frac{di_{b5}}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{b1} \\ u_{b2} \\ i_{b4} \\ i_{b5} \end{bmatrix}. \tag{27}$$

Compared with (8), the matrix $\mathbf{M}$ on the left hand side is no longer a diagonal matrix, thus its inversion is not that explicit. However, there are two features of $\mathbf{M}$ that can be leveraged to simplify the solving process of its inversion. First, every degenerate loop/cutset is decoupled from each other, thus $\mathbf{M}$ can be partitioned into several square blocks on the diagonal so that the inversion of $\mathbf{M}$ is transformed into the inversion of these square blocks. Second, although these blocks are not diagonal matrix, they can be expressed as the summation of a diagonal matrix and a scalar times an all-ones matrix. For example

$$\begin{bmatrix} C_1+C_3 & C_3 \\ C_3 & C_2+C_3 \end{bmatrix} = \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} + C_3 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}. \tag{28}$$

This is actually a rank-1 modification of the original diagonal matrix. According to Sherman–Morrison's formula [23], the inversion of a rank-1 modification of the original matrix can be expressed as

$$\mathbf{M}^{-1} = (\mathbf{D}+\mathbf{uv})^{-1} = \mathbf{D}^{-1} - \sigma \mathbf{D}^{-1}\mathbf{uv}\mathbf{D}^{-1} \tag{29}$$

where $\sigma = 1/(1+\mathbf{v}\mathbf{D}^{-1}\mathbf{u})$ is a scalar.

Without loss of generality, assuming $\mathbf{M}$ has $n-1$ individual diagonals (m_1 to $m_{n-1}, n \geq 2$) and the same off-diagonal entries (m_n), then $\mathbf{D}$, $\mathbf{u}$, and $\mathbf{v}$ in (29) can be selected as

$$\mathbf{D} = \begin{bmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & m_{n-1} \end{bmatrix}$$

$$\mathbf{u} = m_n \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}. \tag{30}$$

The inversion of $\mathbf{M}$ can be derived as follows:

$$\mathbf{M}^{-1} = \begin{bmatrix} \frac{1}{m_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{m_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{m_{n-1}} \end{bmatrix} - \frac{m_n}{1+m_n\sum_{i=1}^{n-1}\frac{1}{m_i}} \begin{bmatrix} \frac{1}{m_1 m_1} & \frac{1}{m_1 m_2} & \cdots & \frac{1}{m_1 m_{n-1}} \\ \frac{1}{m_2 m_1} & \frac{1}{m_2 m_2} & \cdots & \frac{1}{m_2 m_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m_{n-1} m_1} & \frac{1}{m_{n-1} m_2} & \cdots & \frac{1}{m_{n-1} m_{n-1}} \end{bmatrix}$$

$$= \frac{1}{\sum_{j=1}^{n}\prod_{\substack{i=1\\ i\neq j}}^{n} m_i} \begin{bmatrix} \sum_{j=1}^{n}\prod_{\substack{i=1\\ i\neq j}}^{n} m_i & -\prod_{\substack{i=1\\ i\neq 2}}^{n} m_i & \cdots & -\prod_{\substack{i=1\\ i\neq n-1}}^{n} m_i \\ -\prod_{\substack{i=2\\ i\neq 1}}^{n} m_i & \sum_{j=1}^{n}\prod_{\substack{i=1\\ i\neq j}}^{n} m_i & \cdots & -\prod_{\substack{i=2\\ i\neq n-1}}^{n} m_i \\ \vdots & \vdots & \ddots & \vdots \\ -\prod_{\substack{i=n-1\\ i\neq 1}}^{n} m_i & -\prod_{\substack{i=n-1\\ i\neq 2}}^{n} m_i & \cdots & \sum_{j=n-1}^{n}\prod_{\substack{i=1\\ i\neq j}}^{n} m_i \end{bmatrix}. \tag{31}$$

The above process of dealing with degenerate circuit is not hard to be promoted to more general case and a circuit lemma can be summarized.

Lemma 1: A degenerate circuit can be transformed into nondegenerate by the following steps if the loops/cutsets that make the circuit degenerate contain at least one capacitive/inductive branch. First, set the reference direction of the branches in the degenerate loops/cutsets identical*. Second, select one capacitive/inductive branch in the degenerate loops/cutsets and remove it by viewing it as open/short circuit. Third, add self-capacitance/inductance to every branch and mutual-capacitance/inductance to every two branches in these loops/cutsets. The added self- and mutual-capacitance/inductance value is the same with the capacitance/inductance value of the branch that is removed.

When applying the above lemma to the most simple case, i.e., two capacitors connect in parallel or two inductors connect in series, it will result into one capacitor whose value is the sum of these two capacitors and one inductor whose value is the sum of these two inductors, which exactly coincide with the common sense.

Once the degenerate circuit has been transformed into nondegenerate circuit and the inversion of coefficient matrix $\mathbf{M}$ has been found by (31), the upper bound of eigenvalues of the degenerate circuit can be obtained following the same routine in Section III-A.

*Setting the reference direction identical means the branch voltages in the loop form either a clockwise or anticlockwise circle from their positive ends to negative ends and the positive directions of branch currents in the cutset either all flow into or all flow out the cutset. This is to make sure that the removed branch's voltage/current is the opposite of summation of all remaining branches' voltage/currents and to provide a reference direction to the added mutual-capacitance/inductance.

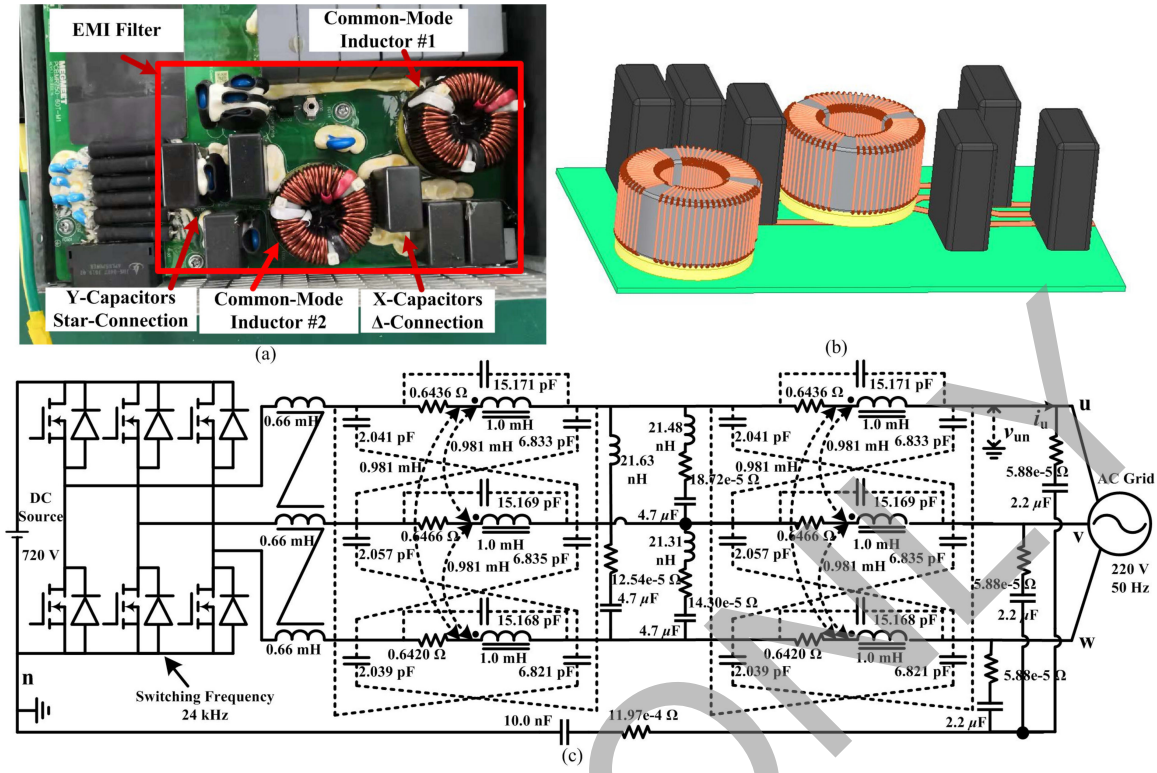


Fig. 4. (a) Structure of the EMI filter prototype. (b) Filter's 3-D model in ANSYS. (c) Equivalent circuit of the test-bench.

IV. Numerical Validation on EMI Filter

To validate the effectiveness of the proposed time-step selection method, a three-phase EMI attenuation filter prototype is constructed and its 3-D model is developed in ANSYS, as shown in Fig. 4(a) and (b). The filter is composed of a first-stage common-mode inductor, followed by three X-capacitors in Δ-connection and the second-stage common-mode inductor, then three Y-capacitors in star-connection and finally a grounding capacitor. This filter is placed between a three-phase two-level converter and the ac grid to test its performance. The equivalent circuit of the experimental setup is shown in Fig. 4(c) where the parasitic resistors, capacitors, and inductors in the windings and wires of the EMI filter are taken into account. The parasitic network of common-mode inductor is adopted from [11] and the parameters of these parasitic elements are extracted by ANSYS Q3D tool using finite element method.

As can be seen that, this is a typical HFLL circuit where the resistor values are very small. By neglecting the effects of these resistors, the remaining parts become a lossless degenerate circuit. As a consequence, some manual derivations are needed to transform it into nondegenerate by following the procedures presented in Section III-B. Once the nondegenerate form is obtained, its state-space system matrix can be generated automatically and the upper bound of this circuit's eigenvalues can be found by following the routine presented in Section III-A, which is 3.3622×10^9. According to (5), the upper bound of the selected time step using RK4 solver is 8.4171×10^{-10} s.

The corresponding numerical simulation is conducted on Matlab platform using the modeling method proposed in [20] with RK4 solver. The results are illustrated in Fig. 5(a) and (b), respectively. It has to be mentioned that although the resistors are neglected in the time-step selection analysis, they are not neglected when conducting the modeling and simulation because the dissipative effect is very important with regard to attenuating the high-frequency noise. The authors in [20] and [24] explained in detail how to take care of these resistors.

The voltage $v_{\rm un}$ (grid side phase-u line to reference ground n at dc side) and current $i_{\rm u}$ are selected for frequency domain analysis. To make a complete evaluation, the high-frequency spectrum (100 kHz to 30 MHz) of both simulation and experimental results are also displayed in Fig. 5. Noted that unlike the typical EMI measurement setup through line impedance stabilization network (LISN) [25], this article chooses to measure converter current/voltage directly and then conducts FFT to obtain the corresponding frequency spectrum. This unavoidably brings background noise in high-frequency range but manages to keep the circuit HFLL to implement the proposed time-step selection method.

As can be seen that, there are several spikes distributed along the left half axis in the frequency spectrum. These spikes are aroused by the switching devices working at 24 kHz, which is a recognized phenomenon in power electronics analysis.

Most part of the high-frequency spectrum are below 100 dBμV or 100 dBμA, except for a resonant peak at 1.8909 MHz in simulation spectrum and 1.8846 MHz in experimental spectrum. According to the analysis in [26], this can be explained by the resonance between the parasitic

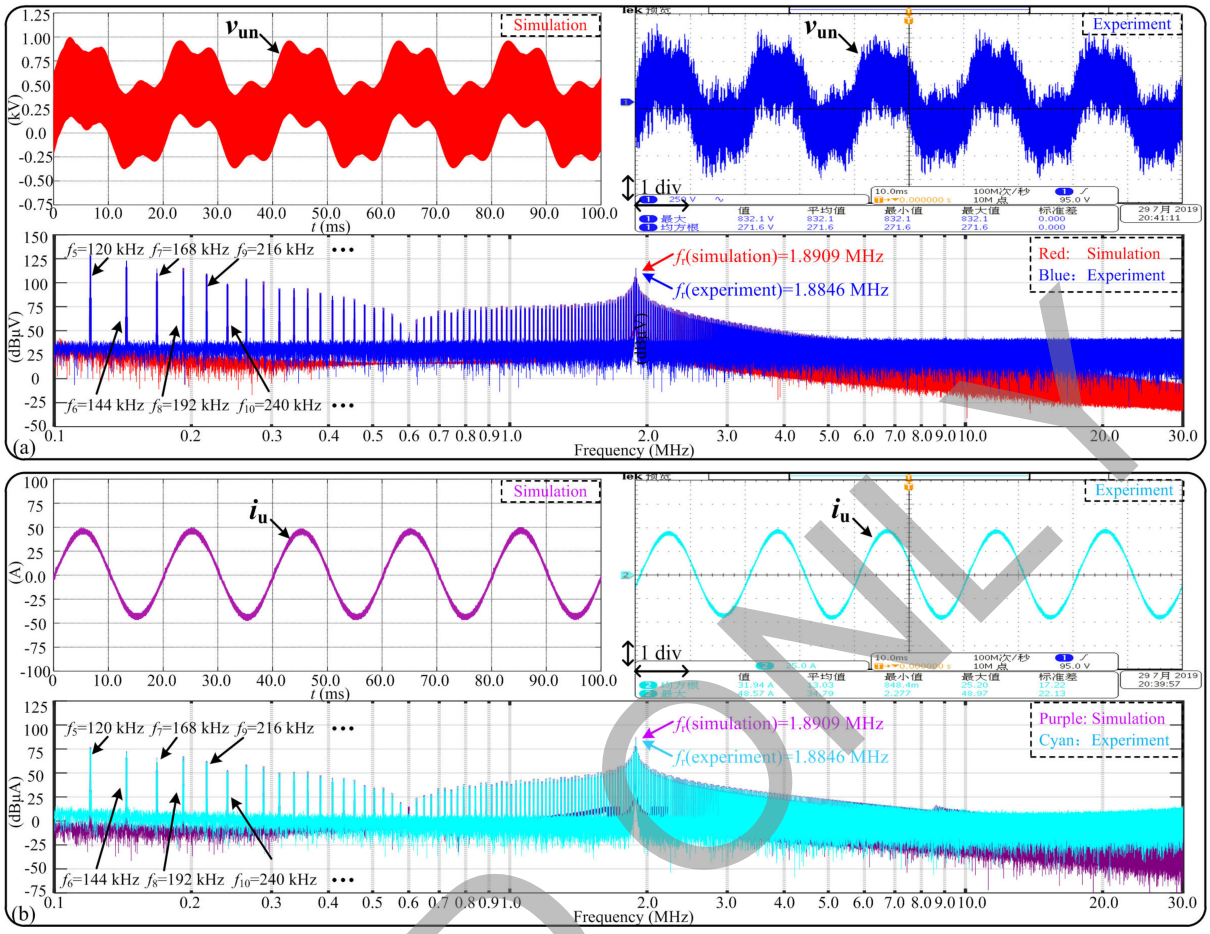


Fig. 5. Time-domain and frequency domain comparison between the proposed time-step selection method-based simulation and experiment. (a) Upper-left: $v_{\rm un}$ from simulation, upper-right: $v_{\rm un}$ from experiment, x-axis: 10 ms/div, y-axis: 0.25 kV/div; lower: frequency spectrum of $v_{\rm un}$ (simulation results in red, experiment results in blue), x-axis: log-scale from 100 kHz to 30 MHz, y-axis 25 dBμV/div. (b) Upper-left: $i_{\rm u}$ from simulation, upper-right: $i_{\rm u}$ from experiment, x-axis: 10 ms/div, y-axis 25 A/div; lower: frequency spectrum of $i_{\rm u}$ (simulation results in purple, experiment results in cyan), x-axis: log-scale from 100 kHz to 30 MHz, y-axis: 25 dBμA/div)

capacitor C_p and the equivalent parallel inductance of common mode inductor L_C and differential mode inductor L_D. Since there are two common mode inductors in the circuit, the resonance frequency can be expressed as

$$f_r = \frac{1}{2\pi\sqrt{\frac{L_C L'_D}{2L_C + L'_D} \times C'_p}} = 1.8824 \text{ MHz} \qquad (32)$$

where $L_C = 1.0$ mH, $L'_D = 0.66/3 = 0.22$ mH, $C'_p = 3 \times (15.171 + 6.833 + 2.041) = 72.135$ pF. The reason L'_D is divided by 3 while C'_p is multiplied by 3 is because the three-phase L_D and C_p are connected in parallel in the common-mode equivalent circuit.

As a comparison, the same test bench is also simulated using MALTAB/Simulink's embedded solver and the results are similar with those in Fig. 5. The selected numerical method is the ODE23tb solver because it proves to be an efficient choice for solving stiff problems when a crude error tolerance is permitted [27]. Even though this is one of the most efficient solver in Matlab/Simulink, it is still much slower than the proposed time selection method with RK4 solver on this test bench. To conduct the same 100.0-ms simulation in Fig. 5, the RK4 solver with 8.4171×10^{-10}-s time step consumes 902.7 s while the ODE23tb solver costs 15128.6 s on the same computer. The former is more than 16 times faster than the latter and also faster than some reported performance on EMI time-domain simulations like in [16] (9000 times versus 65 000 times slower than real time).

The efficiency difference mainly results from the solver's computational complexity. RK4 solver is an explicit method so that no matrix inversion is required at every time-step calculation (the computation complexity remains $O(N^2)$) while ODE23tb is an implementation of trapezoidal rule with the second-order backward difference formula (TR-BDF2), an implicit Runge–Kutta formula with two stages [27], and this makes its computation complexity in $O(N^3)$. Although the average time step of ODE23tb solver (ODE23tb is a variable time-step method) is 1.205×10^{-9} s and larger than the one selected for RK4 solver, the implicit solver still consumes much more time than the explicit solver.

The experimental and simulation results agree with each other in very high degree, yet there are still some discrepancies between them and the theoretic analysis. The following factors may help explain the discrepancy. First, some nonlinear features like the saturation of inductors, the temperature varying phenomenon of capacitors and resistors are omitted. Second, the near-field coupling between inductors, inductors and capacitors, and between capacitors, especially the coupling between the first filter stage and the second filter stage, could contribute to some differences. Third, there are some background noise in the experimental measurement. That's also the reason spectrums from experiment do not attenuate in frequency higher than 5 MHz. Fourth, the above frequency domain results are obtained based on 10-ns sampling period, which is not the integer multiple of the solver's time step. Therefore, linear interpolation has to be adopted to sample the signals at 10-ns interval. This may lead to some frequency spectrum distortions. Nevertheless, it is safe to say that the adopted numerical solver has achieved numerical stability along the whole simulation. The selected time step is valid and superior than some traditional time-step selection schemes.

V. Conclusion

This article presents a fast method to select time step for simulating high-frequency low loss circuit using explicit solver. The state-space model analysis of HFLL circuit shows that all its eigenvalues distribute close to the imaginary-axis, thus transforming the time-step selection problem into finding the upper bound of these eigenvalues. The process of formulating nondegenerate and degenerate circuits' state-space model is elaborated and a transformation lemma from the former case into the latter case is presented. Time step is selected based on the Laguerre–Samuelson's inequality, which involves only two matrix multiplications during the calculation. As a verification, the equivalent circuit of an EMI filter that takes into account the parasitic effects is extracted from ANSYS and a numerical simulation is conducted to compare the performance of the presented time-step selection manner. Numerical results show that the presented method is able to capture all major and parasitic features of the circuit while be much more (16 times) efficient in computation time than the traditional method. Experimental field test of the EMI filter in a dc/ac converter validates the effectiveness of the proposed method.

Acknowledgment

The Authors would like to thank X. Zhang, F. Fang, and K. Zheng in helping us build the experimental setup and conduct EMI measurement.

References

[1] J. Sun and L. Xing, "Parameterization of three-phase electric machine models for EMI simulation," *IEEE Trans. Power Electron.*, vol. 29, no. 1, pp. 36–41, Jan. 2014.

[2] A. Subramanian and U. Govindarajan, "Analysis and mitigation of EMI in DC–DC converters using QR interaction," *IET Circuits Devices Syst.*, vol. 11, no. 4, pp. 371–380, Jan. 2017.

[3] Z. Duan, D. Zhang, T. Fan, and X. Wen, "Prediction of conducted EMI in three phase inverters by simulation method," in *Proc. IEEE Transp. Electrific. Conf. Expo*, Harbin, China, 2017, pp. 1–6.

[4] J. Lai, X. Huang, E. Pepa, S. Chen, and T. W. Nehl, "Inverter EMI modeling and simulation methodologies," *IEEE Trans. Ind. Electron.*, vol. 53, no. 3, pp. 736–744, Jun. 2006.

[5] R. S. Esfandiari, *Numerical Methods for Engineers and Scientists Using MATLAB*. Boca Raton, FL, USA: CRC Press, 2013.

[6] P. Maffezzoni, "A versatile time-domain approach to simulate oscillators in RF circuits," *IEEE Trans. Circuits Sys. I, Regular Papers*, vol. 56, no. 3, pp. 594–603, Mar. 2009.

[7] N. Watson and J. Arrillaga, *Power Systems Electromagnetic Transients Simulation*, Milton Keynes, U.K.: Lighting Source, 2007.

[8] E. Fehlberg, "Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems," *Nat. Aeronaut. and Space Admin., Washington, DC, USA, Tech. Rep. R-315,* 1969.

[9] R. Biswas, *Parallel Computational Fluid Dynamics: Recent Advances and Future Directions*, Lancaster, PA, USA: DEStech, 2009.

[10] J. D. Skufca, "Analysis still matters: A surprising instance of failure of Runge–Kutta–Felberg ODE solvers," *SIAM Rev.*, vol. 46, no. 4, pp. 729–737, 2004.

[11] M. L. Heldwein, L. Dalessandro, and J. K. Kolar, "The three-phase common-mode inductor: modeling and design issues," *IEEE Trans. Ind. Electron.*, vol. 58, no. 8, pp. 3264–3274, Aug. 2011.

[12] W. Tan, C. Cuellar, X. Margueron, and N. Idir, "A high frequency equivalent circuit and parameter extraction procedure for common mode choke in the EMI filter," *IEEE Trans. Power Electron.*, vol. 28, no. 3, pp. 1157–1166, Mar. 2013.

[13] I. F. Kovačević, T. Friedli, A. M. Muesing, and J. K. Kolar, "3-D electromagnetic modeling of EMI input filters," *IEEE Trans. Ind. Electron.*, vol. 61, no. 1, pp. 231–242, Jan. 2014.

[14] M. Illia, and G. Griepentrog, "Finite element method based electromagnetic modeling of three-phase EMI filters," in *Proc. IEEE 2nd Annu. Southern Power Electron. Conf.*, Auckland, New Zealand, 2016, pp. 1–6.

[15] Z. Yu *et al.*, "Simulation analysis on conducted EMD caused by valves in ±800 kV UHVDC converter station," *IEEE Trans. Electromagn. Compat.*, vol. 51, no. 2, pp. 236–244, May 2009.

[16] B. Wunsch, I. Stevanović, and S. Skibin, "Length-scalable multiconductor aable modeling for EMI simulations in power electronics," *IEEE Trans. Power Electron.*, vol. 32, no. 3, pp. 1908–1916, Mar. 2017.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3{rd} ed., Cambridge U.K.: Cambridge Univ. Press, 2007.

[18] S. T. Jensen and G. P. H. Styan, "Some comments and a bibliography on the laguerre-samuelson inequality with extensions and applications in statistics and matrix theory," in *Analytic and Geometric Inequalities and Applications.* Dordrecht, The Netherlands: Springer, pp. 151–181, 1999.

[19] D. L. Skaar, "Using the superposition method to formulate the state variable matrix for linear networks," *IEEE Trans. Educ.*, vol. 44, no. 4, pp. 311–314, Nov. 2001.

[20] Z. Huang and V. Dinavahi, "An efficient hierarchical zonal method for large-scale circuit simulation and its real-time application on more electric aircraft microgrid," *IEEE Trans. Ind. Electron.*, vol. 66, no. 7, pp. 5778–5786, Jul. 2019.

[21] D. Kalman, "A matrix proof of Newton's identities," *Math. Mag.*, vol. 73, no. 4, pp. 313–315, Oct. 2000.

[22] H. Wolkowicz and G. P. H. Styan, "Bounds for eigenvalues using traces," *Linear Algebra Appl.*, no. 29, pp. 471–506, Feb. 1980.

[23] W. W. Hager, "Updating the inverse of a matrix," *SIAM Rev.*, vol. 31, no. 2, pp. 221–239, 1989.

[24] Z. Huang, C. Tang, and V. Dinavahi, "Unified solver based real-time multi-domain simulation of aircraft electro-mechanical-actuator," *IEEE Trans. Energy Convers.*, vol. 34, no. 4, pp. 2148–2157, Dec. 2019.

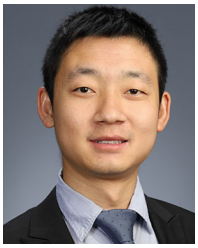
[25] Y. Liu, S. Jiang, W. Liang, H. Wang, and J. Peng, "Modeling and design of the magnetic integration of single- and multi-stage EMI filters," *IEEE Trans. Power Electron.*, vol. 35, no. 1, pp. 276–288, Jan. 2020.

[26] H. Akagi, H. Hasegawa, and T. Doumoto, "Design and performance of a passive EMI filter for use with a voltage-source PWM inverter having sinusoidal output voltage and zero common-mode voltage," *IEEE Trans. Power Electron.*, vol. 19, no. 4, pp. 1069–1076, Jul. 2004.

[27] The Mathworks, Natick, MA, USA, Choose a Solver. (2019). [Online]. Available: https://www.mathworks.com/help/simulink/ug/types-of-solvers.html

Zhen Huang (Student Member, IEEE) received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2012, and the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2015. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada.

His research interests include parallel computing, high-performance modeling and simulation of power electronics and power systems.



Ruimin Zhu (Member, IEEE) received the B.Eng. and Ph.D. degree in electrical engineering from North China Electric Power University, Beijing, China, in 2012 and 2020 respectively.

He is currently a Lecturer with the School of Electrical and Electronic Engineering, North China Electric Power University, Baoding, China. In 2018 and 2019, he was a Joint Ph.D. Student with RTXLAB, University of Alberta, Edmonton, AB, Canada. His research interests include EMI analysis of power systems, power electronics, device-level modeling, and parallel simulation.

Chengcheng Tang (Student Member, IEEE) received the B.S. degree in electrical engineering from North China Electric Power University, Baoding, China, in 2013, and the M.Eng. degree in biomedical engineering from Tsinghua University, Beijing, China, in 2018. She is currently working toward Ph.D. degree with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada.

Her research interests include parallel computing and circuit design.

Venkata Dinavahi (Fellow, IEEE) received the B.Eng. degree in electrical engineering from Visvesvaraya National Institute of Technology, Nagpur, India, in 1993, the M.Tech degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2000.

He is currently a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include real-time simulation of power systems and power electronic systems, electromagnetic transients, device-level modeling, large-scale systems, and parallel and distributed computing.