RE-INTEGRATE EMT Simulation Software: DAE Solvers and Automation

Phani R V Marthi[1] marthip@ornl.gov
Soumyajit Gangopadhyay[1,2] szg6052@psu.edu
Kuan-Chieh Hsu[3] khsu037@ucr.edu
Jongchan Choi[1] choij1@ornl.gov
Nilanjan Ray Chaudhuri[2] nuc88@psu.edu
Suman Debnath[1] debnaths@ornl.gov

***Abstract*—Existing electromagnetic transient (EMT) simulation tools face challenges in accelerating EMT simulations, especially for very large-scale power networks. To tackle this issue, next generation EMT simulation tools such as RE-INTEGRATE EMT are being researched upon. Such tools should be equipped with automation capabilities and advanced numerical differential-algebraic equation (DAE) solvers. In this paper, the DAE solvers incorporated within the RE-INTEGRATE EMT simulation tool are discussed. In particular, a modified ODEINT-based DAE solver and the ARKODE solver from SUNDIALS are leveraged within RE-INTEGRATE EMT. In addition, the automation implemented within RE-INTEGRATE EMT to automate the DAE generation (replacing the need of manual discretization and assembling DAEs) is discussed. Different use cases were implemented using the RE-INTEGRATE EMT tool and were validated with respect to baseline simulations.**

***Index Terms*—Electromagnetic Transient, EMT, differential-algebraic equations, DAEs, stiff solvers, non-stiff solvers**

I. Introduction

With the increasing integration of power electronics systems and smaller electrical machines into the power grid, faster timescale events such as transient events, oscillations, and control interactions are observed more frequently. Electromagnetic transient (EMT) simulation tools are needed to accurately capture these events [1] since such tools can simulate the electromagnetic behavior of electrical systems that are observed during fast changes spanning a few microseconds or smaller.

Different EMT simulation tools are available today such as PSCAD, EMTP, among others. These tools are based on the nodal approach [2]. However, the major challenges associated with EMT simulation of a large-scale power system equipped with power electronics based systems such as inverter-based resources (IBRs), inverter-based loads, and distributed energy resources (DERs) include: (a) limited scalability of power grid simulation; (b) lack of automation; and (c) speed-up limited by computational constraints. To address these challenges, newer generation of EMT simulation tools have to be researched upon. These newer generation of EMT simulation tools should be infused with state-of-the-art advanced numerical solvers, automation algorithms, high-performance computing (HPC) techniques, and artificial intelligence algorithms.

Different from the existing nodal analysis based EMT simulation tools, this work presents a new software framework for EMT simulations that generates and discretizes the differential-algebraic equations (DAEs) of a power system. This newer generation EMT simulation tool, called RE-INTEGRATE, includes DAE solvers and automation pipelines required to accelerate EMT simulations. In this paper, the DAE solvers that are currently utilized within RE-INTEGRATE are discussed. One of the DAE solvers is developed by modifying ODEINT [3], a library that was originally designed for solving ordinary differential equations (ODEs). The other solver is the Implicit-Explicit (ImEx) solver adopted from the Suite of Nonlinear and Differential/Algebraic Equation Solvers (SUNDIALS) [4], [5]. This paper also outlines the automation processes currently implemented within RE-INTEGRATE in order to automate (a) the process of DAE generation of an electrical network and (b) the process of discretization involved in solving the DAEs. Two different use cases that were implemented using the DAE solvers are documented that confirms the accuracy of the developed EMT software.

II. RE-INTEGRATE EMT Simulation Tool

RE-INTEGRATE EMT simulation tool is the next generation EMT simulation tool that can perform EMT simulation of large-scale power grids with power electronics (PE)-based systems and other fast-acting components. This section describes the main pillars of RE-INTEGRATE EMT that include automation, DAE solvers, linear solvers, and intelligence. Other pillars of this simulation tool such as models/libraries and features extraction will be described in future papers.

A. Automation

Automation consists of five sequential steps. At first, a parser is developed to process a netlist (see Fig. 4b for an example) that characterizes the system under study. For each component in the netlist, the parser first creates an object of the component's class and then initializes it with the component's parameters read from the netlist[1]. Based on the source and the

Research sponsored by Applied Grid Modeling (AGM) program in U. S. Department of Energy's Office of Electricity (OE). The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government. [1]Energy Science & Technology Science Directorate, Oak Ridge National Laboratory, Knoxville, TN, USA. [2]Electrical Engineering and Computer Science, Pennsylvania State University, University Park, PA, USA. [3]Department of Computer Science, University of California Riverside, Riverside, CA. Currently he is with Brookhaven National Laboratory.

[1]Currently, the parser can process resistance, inductance, capacitance, voltage source, and current source only. This is being extended to other power system components like generators, exciters, power system stabilizers, etc.

destination nodes of the components stored in the components' objects, an undirected graph is then formed where each node serves as a vertex and a component connected between two nodes serves as an edge of the graph. Since loop identification enables KVL analysis, identifying the components forming a loop in the connectivity graph becomes necessary to obtain a network's DAEs. To this end, a modified Depth First Search (DFS) algorithm is developed to identify the loops such that no two loops lead to the same DAE upon KVL analysis. Post loop identification, KVL analysis is conducted on each of the m identified loops, resulting in m DAEs. Then, KCL analysis is conducted on $n-1$ nodes for a network with n nodes, resulting in $n-1$ DAEs. Consequently, for a system with n nodes and m loops, $m+n-1$ DAEs are formed. Among these $m+n-1$ DAEs, there may exist trivial DAEs discussed further in section III-A. At first, such DAEs (if any) are eliminated. Then, the remaining non-trivial DAEs are separated into the ODEs and the algebraic equations (AEs).

B. DAE Solvers

The RE-INTEGRATE EMT tool employs DAE solvers that are developed by modifying and incorporating the ODEINT and the SUNDIALS libraries. Consequently, RE-INTEGRATE EMT houses different DAE solvers having different orders of discretization, different stiffness-based discretization methods, among others, that can facilitate dynamic simulation of systems defined by DAEs with different properties. The DAE solvers are automatically invoked at the end of the automation step in order to discretize the generated DAEs, leading to a linear system of equations at each time step during the simulation interval obtained as an input from the user. More details on the DAE solvers are available in section III.

C. Linear Solvers

Based on the discretization of the DAEs representing EMT simulation dynamic models, a linear system of equations having the form $\mathrm{Ax} = \mathrm{b}$ (where A is a square matrix) is generated and solved at each time step of the simulation. For a large-scale power system, the matrix A is not only sparse but also has special forms and properties [6]. Within RE-INTEGRATE EMT, several sparse linear solvers that can leverage these special properties and forms of the A matrix are utilized to improve the scalability and the speed of the EMT simulation of a large-scale power system. Some of the linear solvers incorporated within RE-INTEGRATE EMT include KLU [7] and SuperLU. This work does not discuss the linear solvers in detail that will be presented in future papers.

D. Intelligence

Intelligence is one of the key pillars within RE-INTEGRATE EMT. The DAEs obtained for power grids with fast-acting devices have properties that can be explored to use different types of discretization methods [8]. Moreover, the matrices resulting from large-scale power system EMT simulation models are often sparse with specific structural configurations. Intelligence is also being embedded in the RE-INTEGRATE EMT simulation tool to identify the specific structure of the matrices. These intelligent methods help in re-ordering the matrices that lend themselves to a variety of linear solvers that are appropriate for enhancing speed-up and achieving scalability in EMT simulation of large-scale power grids [6]. Intelligence is also utilized within RE-INTEGRATE for automation of large-scale systems [8].

The rest of this paper solely focuses on the DAE solvers and the automations developed within RE-INTEGRATE EMT.

III. DAE Solvers

This section outlines (a) the modifications incorporated within ODEINT and (b) the capabilities utilized from SUNDIALS to solve the DAEs generated within RE-INTEGRATE EMT.

A. ODEINT

ODEINT includes several numerical integrators (such as Euler, implicit Euler, adams-bashforth, adams-moulton, rosenbrock, and explicit runge-kutta methods) that can solve a set of ODEs, provided by the user as an input, over a pre-specified time interval set by the user. Since ODEINT can solve only ODEs, ODEINT is appropriately modified to interface it with RE-INTEGRATE EMT. In particular, ODEINT is modified to (i) automatically accept the set of DAEs generated within RE-INTEGRATE EMT as input instead of user-defined ODEs only, (ii) handle AEs in addition to ODEs, and (iii) delink the linear solvers from the DAE solvers at each time step of the simulation. The proposed modifications are as follows:

1) Automated DAE generation within RE-INTEGRATE EMT: To generate a network's DAEs, identifying the components (or edges) constituting a loop becomes necessary. This is because, for each component, the voltage-current ($v-i$) dynamics is described by a DAE that differs across components. The traditional DFS algorithm forms and stores the adjacency list of a graph that indicates whether a pair of vertices is directly connected or not. While this helps in identifying the vertices forming form a loop, the algorithm cannot identify all the loops when multiple edges connect a pair of nodes. In an electrical network, multiple components can be connected between a pair of nodes, leading to multiple loops involving the same pair of nodes. Inability to detect and identify such loops may underestimate the number of DAEs required to characterize the network. To address this issue, the DFS algorithm is modified within RE-INTEGRATE EMT. In particular, besides storing the adjacency list of the connectivity graph, RE-INTEGRATE EMT stores (a) the number of edges between any two nodes and (b) the information of the component corresponding to each edge in a form similar to that of the adjacency list. The resulting advantages are two-fold. First, multiple loops involving the same pair of nodes can be identified. Second, upon detection of a loop, the components forming the loop can be back-tracked simultaneously while back-tracking the nodes forming the loop.

It is worth noting that with the proposed modification, the DFS algorithm may identify two different loops that yield the same DAE upon KVL analysis as shown in Fig. 1a. Starting from node 0, the algorithm may identify two loops - Loop 1 and Loop 2. Since (i) a capacitor's $v-i$ dynamics is *uniquely* determined by the capacitor's voltage and (ii) C_1 and C_2 are connected in parallel, the DAEs resulting from KVL analysis of Loop 1 and Loop 2 are identical. Identical DAEs may also be encountered when two independent current sources are connected in parallel. Identical DAEs can pose singularity problems for the linear solvers[2]. To circumvent this

[2]Identical DAEs are not formed when resistors/inductors are connected in parallel since their $v-i$ dynamics are *uniquely* determined by their currents.

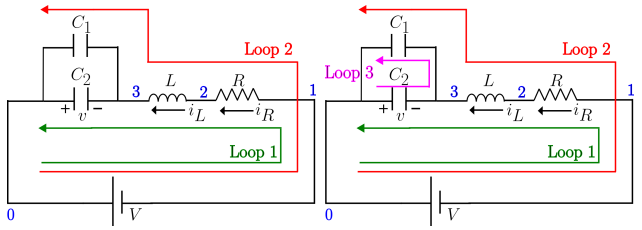


Fig. 1. (a) Identified loops: Loop 1 and Loop 2 (KVL: $-V - i_R R - L\frac{di_L}{dt} + v = 0$) (b) Identified loops: Loop 1/Loop 2 (KVL: $-V - i_R R - L\frac{di_L}{dt} + v = 0$) and Loop 3 (KVL: $-v + v = 0$).

issue, the DFS algorithm is further modified to keep track of the "reverse-parent" nodes for each node in the connectivity graph. In particular, upon a loop detection, if node X directly leads to node Y during back-tracking, then node X is assigned as a "reverse-parent" node of node Y. Instead of identifying Loop 1 and Loop 2 as the two loops, the modified algorithm identifies either Loop 1 and Loop 3 or Loop 2 and Loop 3, thus preventing the generation of identical DAEs from KVL analysis. However, as shown in Fig. 1b, KVL analysis of Loop 3 yields $v = v$ which is trivial. Such trivial DAEs are eliminated to form the final set of N non-trivial DAEs.

2) Modifying ODEINT to solve DAEs: ODEINT is capable of solving a set of ODEs of the form:

$$\dot{x_i} = f_i(\boldsymbol{x}) \qquad \forall i \in \{1, 2, \ldots, m\} \tag{1}$$

where m is a positive integer and $\boldsymbol{x} \in \mathbb{R}^m$ having coordinates $x_i \; \forall i \in \{1, \ldots, m\}$. To solve the DAEs generated within RE-INTEGRATE EMT, the implicit Euler class within ODEINT is modified to tackle linear/affine DAEs of the form (2a)-(2b).

$$\sum_{j=1}^{N} l_{ij}\dot{x_j} = \sum_{j=1}^{N} r_{ij}x_j + c_i \qquad \forall i \in \{1, 2, \ldots, N_1\} \tag{2a}$$

$$0 = \sum_{j=1}^{N} r_{ij}x_j + c_i \qquad \forall i \in \{N_1 + 1, \ldots, N\} \tag{2b}$$

where N is the total number of DAEs, out of which N_1 are ODEs and the rest are AEs. The coefficients l_{ij}, r_{ij} and c_i are real numbers $\forall i, j \in \{1, 2, \ldots, N\}$. At each time step of the simulation, the objective is to generate the matrix A_{DAE} and the vector b_{DAE} such that the solution at the current time step can be obtained by solving the linear system $A_{DAE}x = b_{DAE}$. To realize this objective, at first, the matrices J_L, $J_R \in \mathbb{R}^{N \times N}$ are formed within the modified implicit Euler class as:

$$J_L \leftarrow \begin{bmatrix} l_{11} & \ldots & l_{N_1 1} & 0 & \ldots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ l_{1N} & \ldots & l_{N_1 N} & 0 & \ldots & 0 \end{bmatrix}^T \quad J_R \leftarrow \begin{bmatrix} r_{11} & \ldots & r_{1N} \\ \vdots & \vdots & \vdots \\ r_{N1} & \ldots & r_{NN} \end{bmatrix}$$

Then, the matrices A, B, C, D, E, F are formed as:

$$\begin{aligned} A &\leftarrow (J_L)_{i,j} \;\; i \in \{1, \ldots, N_1\}, j \in \{1, \ldots, N\} \\ B &\leftarrow (J_L)_{i,j} \;\; i \in \{N_1 + 1, \ldots, N\}, j \in \{1, \ldots, N\} \\ C &\leftarrow (J_R)_{i,j} \;\; i \in \{1, \ldots, N_1\}, j \in \{1, \ldots, N\} \\ D &\leftarrow (J_R)_{i,j} \;\; i \in \{N_1 + 1, \ldots, N\}, j \in \{1, \ldots, N\} \\ E &\leftarrow A - hC \quad \text{and} \quad F \leftarrow B - D \end{aligned}$$

where $h > 0$ is the stepsize used in the simulation. The matrix $A_{DAE} \in \mathbb{R}^{N \times N}$ and the vector $b_{DAE} \in \mathbb{R}^N$ are formed as:

$$\begin{aligned} A_{DAE} &\leftarrow \begin{bmatrix} E^T & F^T \end{bmatrix}^T \\ b_{DAE} &\leftarrow J_L x(t_n) + \begin{bmatrix} hc_1 & \ldots & hc_{N_1} & c_{N_1+1} & \ldots & c_N \end{bmatrix}^T \end{aligned}$$

where $x(t_n)$ is the solution at $t = t_n$. The solution $x(t_n + h)$ at $t = t_n + h$ is currently obtained by solving $A_{DAE}x = b_{DAE}$ using the LU solver available within ODEINT.

3) Delinking Linear solvers from DAE solvers: In the existing ODEINT codebase, the LU solver is invoked from within the implicit Euler class. Consequently, the matrix A_{DAE} and the vector b_{DAE} (generated within the modified implicit Euler class at each time step) cannot be accessed from outside the implicit Euler class. This in turn restricts the choice of the linear solver to the ones available within ODEINT. Hence, RE-INTEGRATE EMT currently delinks the linear solver routines from the DAE solver routines. As a consequence, a linear solver cannot be invoked from within the DAE solver class. Instead, A_{DAE} and b_{DAE} can now be accessed outside the DAE solver class which creates room for incorporating novel linear solvers within RE-INTEGRATE EMT in the future.

B. SUNDIALS: Implicit-Explicit Solvers

SUNDIALS has a package of ODE and DAE solvers. For this work, SUNDIALS ImEx solvers are utilized to exploit the stiffness property present within DAEs representing the dynamics of a power grid equipped with power electronics. However, SUNDIALS ImEx solvers do not incorporate first-order solvers that are available within ODEINT.

1) ImEx Solvers: Implicit-Explicit Solvers: SUNDIALS consists of a DAE solver package called ARKODE that can solve initial value problems (IVPs) given by (3).

$$x' = f_I(x, t) + f_E(x, t); \qquad x(t_0) = x_0 \tag{3}$$

where, x, t represents state of the system and time respectively; $f(x, t)$ denotes the equations that represent the dynnamics of the system. The dynamics of the system are segregated into two parts: "stiff" and "non-stiff" components. $f_I(x, t)$ contains the "stiff" components of the system and $f_E(x, t)$ contains the "non-stiff" components of the system. The ARKODE package in SUNDIALS supports one-step time implicit-explicit (ImEx) additive Runge–Kutta methods. The "stiff" components-based states in $f_I(x, t)$ are integrated using diagonally-implicit Runge-Kutta methods (DIRK), and the "non-stiff" components-based states are integrated using explicit Runge-Kutta methods (ERK) [9]. The user should provide both functions $f_I(x, t)$ and $f_E(x, t)$ for the mixed stiff/non-stiff problems that define the dynamics of the system.

2) Sources of Stiff System DAEs in EMT Simulation: The key factors that can play a key role in introducing stiffness into the DAEs are ratios of inductances, capacitances, high-frequency switching actions of power-electronics systems, rapid dynamics of various filter components, fast dynamics of the input and output voltages, and interactions between the filter components and loads. Furthermore, non-linearities in the systems also play a key role in determining the stiffness in the systems. Some examples of "stiff" system of DAEs include: arm current dynamics during sub module (SM) blocked condition in Modular multilevel converter (MMC) - High Voltage

direct current (HVdc) [10]; inductor current dynamics in direct current (dc)-dc converters models, dc-alternating current (ac) inverters models, and filter currents and voltages dynamics in dc-ac inverters of the photovoltaic (PV) inverter system models, among others [11]. Some examples of the "non-stiff" system of DAEs include: SM capacitor voltage dynamics in MMC-HVdc system models [10], input and output voltage dynamics of the dc-dc converter models [11], among others.

3) ImEx Solvers: Implementation and Automation: Once the "stiff" and "non-stiff" DAEs of the system are identified, the system module is developed in C programming language using ImEx solver module ARKODE and included into $f_I(x,t)$ and $f_E(x,t)$. The "stiff" and "non-stiff" functions are then processed (discretization required for EMT simulation) using $ARKStepEvolve$ function (in the ARKODE integrator), thereby automating the entire discretization process for EMT simulation of the PV inverter module with SUNDIALS. Subsequently, the system can employ either direct or iterative solvers to solve these equations and update the state variables.

4) Linear Solvers: SUNDIALS: In this research, the matrix-based solver SUNLinSol Dense from SUNDIALS is chosen for example purposes; however, users have the flexibility to select alternative solvers from SUNDIALS or external sources.

IV. Case Studies and Simulation Results

In this section, two case studies are shown: (a) PV inverter simulation using ImEx solver in SUNDIALS; (b) Circuit simulation using the modified ODEINT based DAE solver.

A. PV inverter

The PV inverter module considered in this use-case is shown in Fig. 2. The PV inverter module consists of PV array, dc-dc boost converter, dc-ac inverter, and LCL filter. In the PV inverter EMT simulation model, the voltage and current dynamics of capacitors and inductors in dc-dc converter, dc-ac inverter, and LCL filter are represented by DAEs [11]. In [11], the EMT simulation model of the PV inverter module is simulated by state-space method. In the case of state-space approach, the PV inverter module's DAEs are assembled based on Kirchoff's current and voltage laws to form the system DAEs. The formed system DAEs are then discretized manually using hybrid discretization methods resulting in linear system of equations. The linear system of equations is then solved using generic matrix inversion methods such as LU decomposition or Gauss elimination methods.

In the proposed method using ImEx solvers, DAEs are directly included into the $f_I(x,t)$ and $f_E(x,t)$ based on the stiffness category. In the case of PV inverter module, DAEs for the input and the output voltage dynamics of the dc-dc converter are classified as "non-stiff" components and DAEs for the inductor current dynamics of the dc-dc converter, the inverter currents of the dc-ac inverter are classified as "stiff" components. These DAEs are implemented using the automation process explained in Section III-B3. The linear system of equations generated from this automated discretization process is solved at each time step using the matrix-based solver SUNLinSol Dense from SUNDIALS.

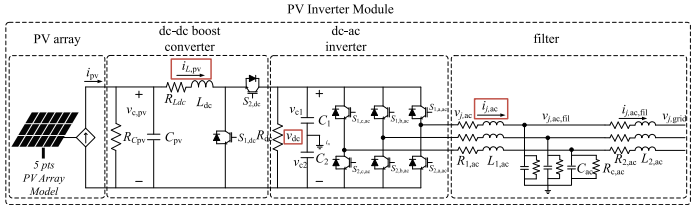


Fig. 2. PV inverter module architecture.

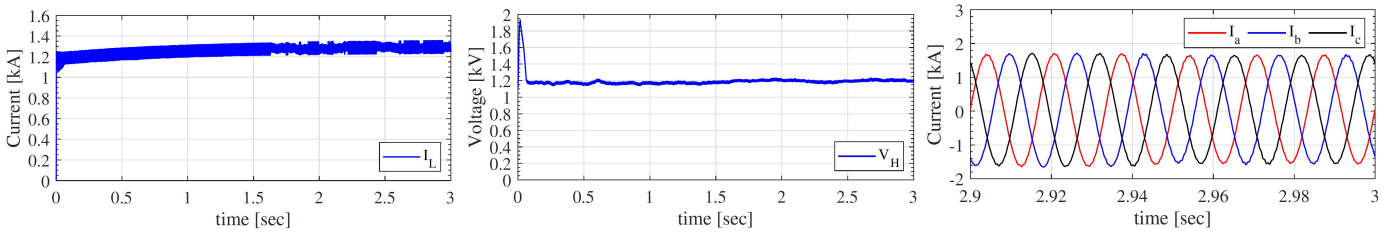
The simulation results for the PV inverter module simulation using SUNDIALS ImEx DAE solvers are shown in Figs. 3a-3c. The inductor current dynamics of the dc-dc converter, the dc-link voltage dynamics of the dc-ac inverter, and the dc-ac inverter current dynamics of the PV inverter module are shown in Figs. 3a-3c respectively (corresponding circuit components highlighted in Fig. 2). From the results, it is observed that the PV inverter module is stable, accurate and compares well with the baseline results from the existing simulators [11].

B. Circuit with resistors, inductors, capacitors and battery

RE-INTEGRATE EMT was used to simulate the circuit shown in Fig. 4a for 0.5 s. At first, the circuit was represented using the netlist shown in Fig. 4b. Then this netlist was fed to RE-INTEGRATE EMT that generated 10 DAEs (8 ODEs and 2 AEs) representing the circuit's dynamics. Within RE-INTEGRATE EMT, the implicit Euler method was set as the DAE solver and the circuit was simulated using three stepsizes: 0.1 ms, 1 ms and 10 ms. For each simulation, the corresponding stepsize was provided as an input to the DAE solver for discretization. Consequently, at each time step, the tool produced an $A_{DAE} \in \mathbb{R}^{10\times10}$ and a $b_{DAE} \in \mathbb{R}^{10}$ which were fed to the LU solver (available within ODEINT) to solve the circuit at that time step. Note that, all six state variables (i.e., v_{0-1}, v_{5-4}, i_{2-1}, i^a_{4-3}, i^b_{4-3}, and i^c_{4-3}) were initialised to zero in the simulations. For each stepsize, the outputs obtained from RE-INTEGRATE EMT were compared with the outputs obtained by simulating the circuit of Fig. 4a in MATLAB Simulink using implicit Euler method and the same stepsize. The objective of this comparison was to verify the accuracy of RE-INTEGRATE EMT across different stepsizes. The obtained results are shown in Fig. 5. Further, the relative error ($r.e.$) between the outputs obtained from Simulink and RE-INTEGRATE EMT was evaluated as:

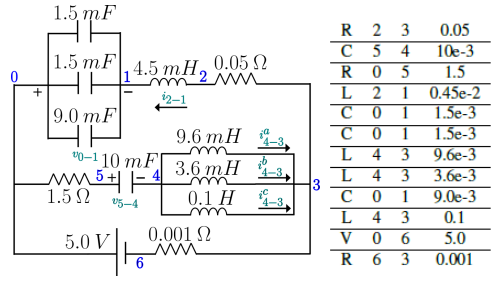
$$r.e.(\%) = \frac{|\text{Simulink output} - \text{RE-INTEGRATE output}|}{|\text{Simulink output}|} \times 100$$

The variation of the relative error against time is also shown in Fig. 5 for the two state variables v_{0-1} and v_{5-4}. It is seen from Fig. 5 that the outputs obtained from RE-INTEGRATE EMT are indistinguishable from the corresponding outputs obtained from Simulink. Further, the relative error is negligible in each of the six dynamics shown in Fig. 5. This demonstrates the generality of RE-INTEGRATE EMT across different stepsizes. The generality of RE-INTEGRATE EMT across different stiffness ratios was also investigated by varying the inductance (between nodes 1 & 2) and the capacitance (between nodes 4 & 5). The results obtained were found to be satisfactory. However, these results are not provided in this paper due to space limitations. Additionally, it is seen from Fig. 5 that unlike v_{5-4} which exhibits a non-oscillatory and over-damped dynamics, the state v_{0-1} exhibits a decaying oscillatory dynamics. This is because, unlike the 1.5 ohm resistance that provides significant damping, the 0.05 ohm resistance is too small to provide



(a) (b) (c)

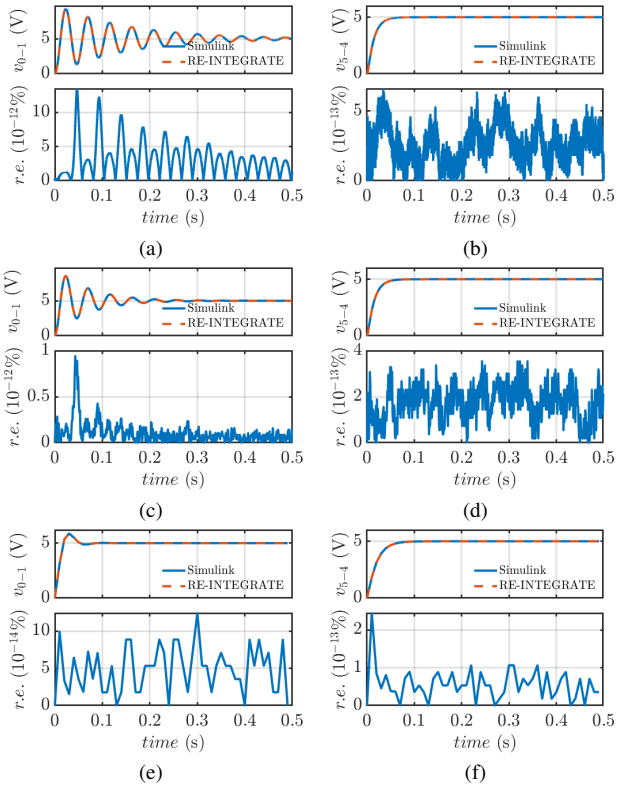
Fig. 3. PV inverter module simulation results: (a) dc-dc converter's inductor current (b) dc-ac inverter's dc-link voltage (c) dc-ac inverter's output current.



R	2	3	0.05
C	5	4	10e-3
R	0	5	1.5
L	2	1	0.45e-2
C	0	1	1.5e-3
C	0	1	1.5e-3
L	4	3	9.6e-3
L	4	3	3.6e-3
C	0	1	9.0e-3
L	4	3	0.1
V	0	6	5.0
R	6	3	0.001

(a) (b)

Fig. 4. (a) Circuit diagram and (b) netlist of the circuit under study.



(a) (b) (c) (d) (e) (f)

Fig. 5. Results obtained from Simulink & RE-INTEGRATE EMT using stepsize 0.1 ms in (a) and (b), 1 ms in (c) and (d), & 10 ms in (e) and (f).

sufficient damping, leading to the under-damped dynamics of v_{0-1}. Further, as seen from Fig. 5, the oscillations decay faster as the stepsize is increased. This is due to the stiff decay property which enables the implicit Euler method to attain the steady state faster by ignoring the transients, when a higher stepsize is used.

V. Conclusion

In this paper, the DAE solvers and the automation processes currently employed for the newer generation EMT simulation tool called RE-INTEGRATE EMT are documented. A brief overview about the different components of RE-INTEGRATE EMT is provided. Two different use-cases are also reported that were implemented using the DAE solvers outlined in this study. From the simulation results, it is observed that the DAE solvers used are accurate for EMT simulations across different stepsizes as well as different stiffness ratios. Further, the automation processes employed within RE-INTEGRATE EMT are found to significantly reduce the manual intervention that was previously required to discretize the DAEs defining the power system components in EMT simulation models, lending itself to simulation of large-scale power networks.

Acknowledgment

The authors would like to thank Alireza Ghassemian from Department of Energy (DOE) Advanced Grid Modeling (AGM) program for his guidance and support in this work.

References

[1] S. Debnath, P. Marthi, J. Choi, and *et al.*, "EMT simulation of large PV plant & power grid for disturbance analysis," in *2023 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT-LA)*, 2023.

[2] H. W. Dommel, "Digital computer solution of electromagnetic transients in single-and multiphase networks," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-88, no. 4, pp. 388–399, 1969.

[3] [Online]. Available: https://www.headmyshoulder.github.io/odeint-v2/

[4] A. C. Hindmarsh, P. N. Brown, K. E. Grant, , and *et al.*, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[5] D. J. Gardner, D. R. Reynolds, C. S. Woodward, and C. J. Balos, "Enabling new flexibility in the sundials suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 48, no. 3, pp. 1–24, 2022.

[6] Q. Xia, S. Debnath, and J. Choi, "Assessing the feasibility of bordered block diagonal reordering in power system matrices using fully convolutional network," in *2024 IEEE Power Energy Society General Meeting (PESGM)*, 2024.

[7] J. Choi and S. Debnath, "Linear solvers for collector systems of generalized large-scale inverter-based resources," in *2024 IEEE Power Energy Society General Meeting (PESGM)*, 2024.

[8] Q. Xia, K. Kurte, and S. Debnath, "Reinforcement learning-based approach for emt automation of large-scale pv plants," in *2024 IEEE Power Energy Society General Meeting (PESGM)*, 2024.

[9] D. R. Reynolds, D. J. Gardner, C. S. Woodward, and R. Chinomona, "ARKODE: A flexible IVP solver infrastructure for one-step methods," *arXiv preprint arXiv:2205.14077*, 2022.

[10] S. Debnath and M. Chinthavali, "Numerical-stiffness-based simulation of mixed transmission systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9215–9224, 2018.

[11] J. Choi and S. Debnath, "Electromagnetic transient (emt) simulation algorithm for evaluation of photovoltaic (pv) generation systems," in *2021 IEEE Kansas Power and Energy Conference (KPEC)*, 2021.