Symbolic Modeling of Non-linear Devices for the Virtual Test Bed

Roger A. Dougal, Senior Member, IEEE and Eugene V. Solodovnik

Abstract—This contribution illustrates that a symbolically assisted technique can provide an efficient and rapid path for developing complex nonlinear device models for power system simulations. As an example, a phase-domain induction machine model is developed with the aid of a symbolic tool. The developed symbolic tool automatically constructs a time-domain power system component models in the Resistive Companion Form (RCF) that is widely used in time-domain simulators. The Automatic Differentiation Technique (ADT) is utilized within the context of a symbolic modeling language, and the tool has been implemented for the Virtual Test Bed (VTB) simulation environment.

Index Terms—Automatic differentiation, symbolically-assisted simulation, Resistive Companion Form, induction machines, power system modeling.

I. INTRODUCTION

Modern power systems contain many nonlinear devices, such as motors, generators, transformers, surge arresters, and power electronic components. Large-scale systems consisting of many nonlinear components are often characterized by complex nonlinear dynamics. The dynamics of such systems are usually studied using time domain simulation methods. There is a variety of numerical integration methods: backward Euler’s, trapezoidal, Simpson’s, Runge-Kutta’s, Gear’s methods, etc. Among these methods, trapezoidal integration is the most popular one in network transient analysis due to its merits of low distortion and absolute-stability (A-stable). For example, trapezoidal integration rule is used in EMTP [1], Spice [2], and Virtual Test Bed (VTB) [3]. Ordinarily, dynamic equation of each circuit element is integrated using trapezoidal integration and thus the element is represented by a parallel combination of an equivalent resistance and an equivalent current source. With this modeling technique, network nodal analysis can then be performed to obtain an overall circuit simulation. This simulation methodology is usually known as Resistive Companion Form (RCF) method [4].

Among many advantages of the RCF modeling method, there is one drawback: the manual creation of the models is a very tedious process. Derivation of the discretized model equations requires analytic evaluation of numerous partial derivatives. Consider for example that a four terminal device requires the computation of at least $4^2$ coefficients in an RCF. Each of these coefficients is expressed by means of partial derivatives of complex expressions.

This contribution illustrates that a symbolically assisted technique can provide an efficient and rapid path for developing RCF models of complex nonlinear devices for power system simulations. We describe how the Automatic Differentiation Technique (ADT) [5] is applied within the context of a symbolic modeling language to develop a model generation tool for the VTB environment. We illustrate the process with the example of a phase-domain model of an induction machine.

II. AUTOMATIC DIFFERENTIATION TECHNIQUE

The major effort in the development of the RCF model equations is the computation of the Jacobian matrices which requires computing many partial derivatives of the device equations. We apply the automatic differentiation technique to eliminate this effort. This technique employs the fact that exact derivatives of the function can be obtained by repeated application of the chain rule. The advantages of this approach are: the process can be readily automated using a high level computer language; the results are accurate, since they are identical to those obtained from analytic evaluation of the derivatives; the run-time computational effort required is relatively short, and equivalent to manually derived analytical models.

The automatic differentiation algorithm consists of two phases: forward and reverse.

A. Forward Phase of ADT

The forward phase involves the calculation of the derivative values of the elementary functions. For example, let us consider a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, which depends on $n$ variables $x_1, x_2, ..., x_n$ called independent variables. The value of the function is obtained through the use of the intermediate variables $x_{n+1}, x_{n+2}, ..., x_N$, so that $f(x) = x_N$. Each variable is calculated from an elementary function $g_j$. For example, for the simple instruction $x_i = x_{i-1} + x_{i-5}$, we associate the function $g_i(x_{i-1}, x_{i-5}) = x_{i-1} + x_{i-5}$. These functions depend only on independent and intermediate variables that have been already
processed. Then, the derivatives $g_{i,j} = \frac{\partial g_i}{\partial x_j}$ for each of the elementary functions $g_i$ are determined and stored.

B. Reverse Phase of ADT

The reverse phase concludes the computation of the derivatives of the given function using the data obtained at the forward phase. The derivative of the function $f(x)$ with respect to the variable $x_k$ can be obtained through adding all of the products of $g_{i,j}$ between nodes $x_k$ and $x_N$, i.e.

$$\frac{\partial f(x)}{\partial x_k} = \sum_{\text{path} C \text{ from } x_k \text{ to } x_N} \prod_{\text{arc}(i,j) \in C} g_{i,j} .$$

To save time on repetitive calculations, the additional variables

$$p_k = \sum_{\text{path} C \text{ from } x_k \text{ to } x_N} \prod_{\text{arc}(i,j) \in C} g_{i,j}, \quad 1 \leq k \leq N-1, \quad p_N = 1$$

are introduced and then the derivatives are computed using following equations:

$$p_N = 1$$

$$p_k = \sum_{j=k+1}^{N} g_{j,k} p_j, \quad n+1 \leq k \leq N-1 . \quad (1)$$

$$\frac{\partial f}{\partial x_k} = \sum_{j=k+1}^{N} g_{j,k} p_j, \quad 1 \leq k \leq n$$

III. AUTOMATIC GENERATION OF THE RCF MODELS IN VTB

The RCF model generator must create code that implements two basic functions: the initialization function, and the time stepping function. The initialization function is called once at the beginning of the simulation. The time stepping function is called repeatedly during the simulation. The computationally intensive parts of the ADT algorithm are completed during the initialization stage. The time stepping part includes only the evaluation of the obtained derivatives (1), ensuring optimal computational efficiency. The detailed descriptions of the algorithm can be found in the reference [6].

IV. APPLICATION EXAMPLE: DEVELOPMENT OF THE RCF MODEL OF THE INDUCTION MACHINE

To facilitate the development of the phase-domain RCF model of the induction machine [7], the model equations of the machine are manipulated and rearranged according to the format of equation (1). The model is highly nonlinear and has 23 equations corresponding to 23 nodes. Thus, the Jacobian matrix for this model has $23^2 = 529$ elements. Apparently, manual calculation of the Jacobian matrix would result in long development time of the model and even longer debugging time, since errors inevitably would be present if the RCF model were to be manually developed by a human. Such complexity of the phase-domain model of the induction machine is the main obstacle to manual development of its RCF representation. However, the symbolically assisted model generator allows for rapid development of the phase-domain RCF model of the induction machine. The automatic model generator ran in just fractions of a second and the resulting model was free of errors. Furthermore, computational efficiency of the automatically developed RCF model is comparable to that of the manually developed model.

To verify the new phase-domain RCF model of the induction machine and demonstrate the power of the developed symbolic model development tool, the starting transients of an induction machine are simulated and compared to the simulation results obtained by using a d-q reference model, which was developed in Matlab. The simulation results obtained from the new phase-domain RCF model in VTB and results from the standard d-q model in Matlab were in good agreement.

V. CONCLUSIONS

The symbolically-assisted tool provides a fast and efficient way of developing complex nonlinear device models. The tool greatly reduces the effort required to develop a complex simulation model.

The time-domain model of a general induction machine was rapidly developed with the aid of a symbolic model development tool. The resulting model was validated by comparison to a standard model and was proven suitable for power system dynamic simulation.

VI. REFERENCES


VII. BIOGRAPHIES

Roger A. Dougal earned the Ph.D. degree in electrical engineering at Texas Tech University in 1983 and immediately joined the faculty at the
University of South Carolina. Dr. Dougal has received the *Samuel Litman Distinguished Professor of Engineering* award, and has been honored as a *Carolina Research Professor*. He is a Senior Member of the IEEE. Prof. Dougal is the director of the Virtual Test Bed project, which is developing an advanced simulation and virtual prototyping environment for multidisciplinary dynamic systems.

**Eugene V. Solodovnik** obtained the E.E. diploma from the Taganrog State University of Radio-Engineering, Taganrog, Russia in 1991; the Ph.D. degree from the University of South Carolina in 2001. Dr. Solodovnik’s interests include control, modeling, analysis and simulation of power systems, and CAD software development.