Symbolic Macromodeling of Parameterized S-Parameter Frequency Responses

Dirk Deschrijver
Dept. of Information Technology
Ghent University - IBBT
9000 Ghent, Belgium
Email: dirk.deschrijver@intec.ugent.be

Tom Dhaene
Dept. of Information Technology
Ghent University - IBBT
9000 Ghent, Belgium
Email: tom.dhaene@intec.ugent.be

Abstract—This paper presents an evolutionary algorithm for symbolic macromodeling of parameterized frequency responses. The method does not require an a priori specification of the multivariate functional form or complexity of the model. Numerical results are shown to illustrate the performance of the technique.
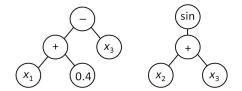
I. Introduction

Efficient design space exploration requires parametric macromodels that describe the dynamic behavior of scalable systems in terms of the frequency and several design variables, e.g. geometrical layout or substrate characteristics. The calculation of compact and accurate models is not a trivial task, and the development of novel macromodeling techniques has received a lot of attention over the past few years. An overview of recent work is found in [1]-[6] and the references therein.

This paper presents an alternative macromodeling procedure that is based on symbolic regression. The goal of the algorithm is to find a mathematical multivariate expression that provides an accurate approximation of the parameterized frequency response. The new approach is more generic than conventional macromodeling techniques, in a sense that it does not require an *a priori* specification of the functional form of the model (e.g. rational, polynomial, neural network, etc) or the model complexity (e.g. number of basis functions or the number of hidden layers). Note that this method provides a compact and scalable macromodel, but it does not take physical properties into account (such as causality, stability and passivity).

The usefulness of the symbolic macromodeling approach is illustrated by applying it to a scalable waveguide example.

II. Symbolic Macromodeling

Symbolic macromodeling is a biologically inspired method that mimics the process of Darwin's evolution theory and the mechanisms of genetic variation and natural selection [7]. It is based on the concept of genetic programming, and computes a *multi-gene* regression model that approximates a set of tabulated, parameterized frequency samples. These data samples can be S-parameters $S(\vec{x})$ that depend on several input parameters $\vec{x} = \{x_1, ..., x_n\}$ (e.g. frequency, length and widths of metallizations in an EM-circuit or substrate parameters). The multivariate model expression is represented as an α-weighted sum of parse trees $T_m(\vec{x})$ (also called *genes*) that

$$T(x_1,x_2,x_3)=\alpha_0+\alpha_1((x_1+0.4)-x_3)+\alpha_2(\sin(x_2+x_3))$$

Fig. 1.   Example : Multi-gene regression model

depend on some (or all) of the n input parameters $\vec{x}$.

$$T(\vec{x}) = \alpha_0 + \alpha_1 T_1(\vec{x}) + ... + \alpha_M T_M(\vec{x}) \qquad (1)$$

Each parse tree $T_m(\vec{x})$ is composed of primitive functions (e.g. $+, -, /, *, \log, \sin, \exp, ...$) and terminals (i.e. input variables $\vec{x}$ and constants). The optimal values of the scalar weights α_m are determined by a least-squares procedure that regresses the genes against the output data. The structure of the genes is evolved automatically as described in the following section. Fig. 1 shows an example of a multi-gene regression model.

III. Evolutionary Algorithm

The algorithm starts by generating an initial population of randomly generated individuals $T(\vec{x})$, which all represent a potential solution to the approximation problem. Each individual has a random number of genes $T_m(\vec{x})$ with $m \leq M$. In order to measure the quality of a particular solution, the fitness function F assigns a *fitness value* to each individual. This fitness function is the objective function that the algorithm aims to optimize. As some individuals will be more fit than others, the best performing individuals are selected and three genetic operators are applied to create new offspring [8] :

− Direct Reproduction, Crossovers and Mutations −

1) The direct reproduction operator simply copies the entire individual to the next generation without modification.
2) The crossover operator selects two parent individuals from the population and combines them to create new offspring by swapping genes or subtrees of the genes
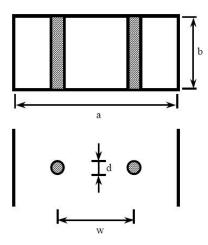3) The mutation operator selects a single gene from the individual and makes a random alteration to its subtree.

Fig. 2. Cross-sectional view and top view of inductive posts [9]

Fig. 3. Real part of transmission coefficient S_{21}

This evolutionary process is repeated over many generations, in order to produce individuals with increasing fitness. Once the algorithm is terminated, the "best" individual from the population is selected and designated as the final result. More details about genetic programming can be found in [7].

IV. EXAMPLE : RECTANGULAR WAVEGUIDE

A. Calculation of Frequency Response

In this example, the method is applied to build a multivariate macromodel of a parameterized frequency response. A moment method is used to compute the transmission coefficient S_{21} of two perfectly conducting round posts, centered in the E-plane of a rectangular waveguide [10]. The system is parameterized by 2 variables : frequency f and post-spacing w. A symbolic macromodel is calculated for a standard WR90 rectangular waveguide, where the frequency f varies between [7-10] GHz and the spacing w between [15-17] mm. The layout of the inductive posts is shown in Fig. 2 and the frequency response is shown in Fig. 3 and Fig. 4.

B. Symbolic Macromodeling

Both the real part $\Re e(S_{21})$ and the imaginary part $\Im m(S_{21})$ of the calculated frequency response are modeled independently by two *multi-gene* regression models, see (1). [11]

The maximum number of genes (M) is set to 12, and the maximum tree depth of each gene is set to 3. The primitive operations are chosen to be $*, -, +, /, \sqrt{|...|}, (...)^2$. Note that other primitive operations can be included, although this was not necessary to obtain a sufficiently accurate macromodel.

First an initial population of 300 randomly generated individuals is generated, and the fitness of each individual is computed as the root-mean-squared prediction error over the data set. Based on the fitness values, tournament selection chooses the individuals that are used to create offspring. If one or more individuals have the same fitness value, then the one with the fewest nodes is selected. Elitism ensures that individuals with the best fitness values (2% of the population) are copied directly to the next generation without applying genetic operators. This process is repeated for 200 generations.
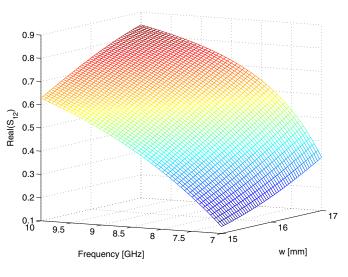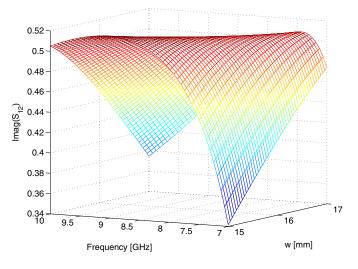

Fig. 4. Imaginary part of transmission coefficient S_{21}

As illustrated by Fig. 5, the accuracy of the fittest individual in each population improves over the course of the run.

C. Selection of the Best (Pareto-Optimal) Model

Once the evolutionary algorithm has converged, one of the individuals in the final population is selected and designated as the "best" solution. This individual is preferably the one with the highest accuracy (i.e. the model with the smallest fitness value) and the smallest model complexity (i.e. fewest number of nodes in the parse tree). Since these two design objectives are usually conflicting, a pareto front is formed that presents a compromise among these two criteria. The fitness and model complexity of each individual in the final population is shown by a dark (blue) circle in Fig. 6 and Fig. 7. The light (green) circles correspond to models that are not
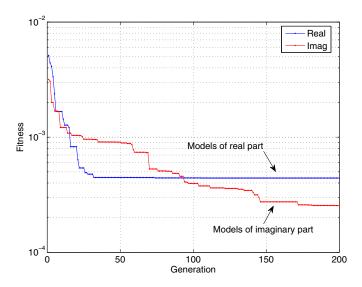
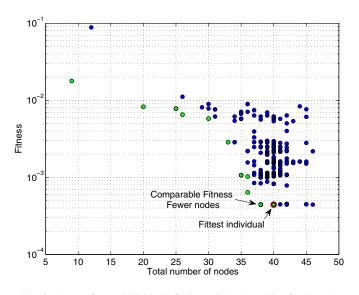Fig. 5. Fitness of most accurate individual over course of the run

Fig. 7. Pareto front : Individuals final population (models of imaginary part)
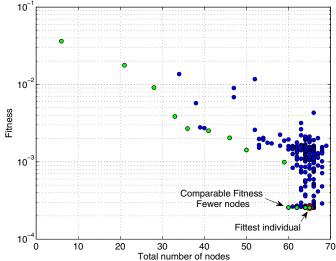

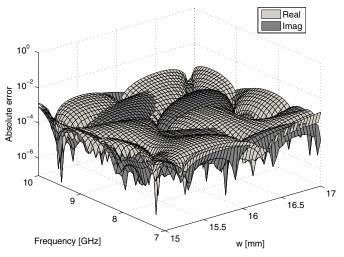Fig. 6. Pareto front : Individuals final population (models of real part)

Fig. 8. Absolute fitting error of the best models

outperformed by any other model in terms of *both* fitness and model complexity. These are so-called "pareto-optimal" models. The most accurate individual, i.e. the one with the smallest fitness value, is marked by a bold (red) circle. It is noted that some of the pareto-optimal models have a fitness value that is comparable to the optimal solution, but a lower model complexity. This information can be exploited by the designer to select a compact model that also meets the desired accuracy specifications.

D. Accuracy of Macromodel

As mentioned above, the best models for the real and imaginary part of S_{21} are selected from the final populations by investigating the Pareto front. Fig. 8 shows the absolute fitting error over the design space for both models, and it is found that the maximum absolute error is approximately 10^{-3}

in both cases, which is sufficiently accurate. The parse trees of these models are concatenated as shown in Fig. 9. This yields a larger, complex-valued parse tree that characterizes the overall frequency response. Based on this compound expression, the magnitude and phase of the model can also be calculated and compared to the reference data, as shown in Fig. 10 and Fig. 11. It is clear that a good agreement is observed in both cases.

As a final test, a dense set of 2500 validation samples are computed and compared to the response of the model. It is found that the maximum absolute error over all data samples is -55.68 dB. The distribution of the absolute error over the reference samples is visualized by a histogram in Fig. 12.

V. CONCLUSION

This paper proposes an evolutionary algorithm for the symbolic macromodeling of parameterized frequency responses.
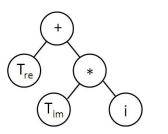
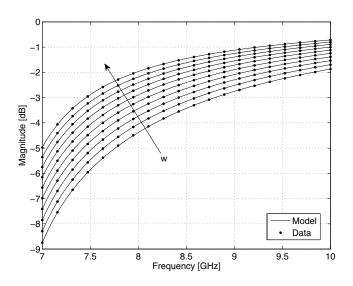Fig. 9.   Concatenation of 2 parse trees (models of real and imaginary part)
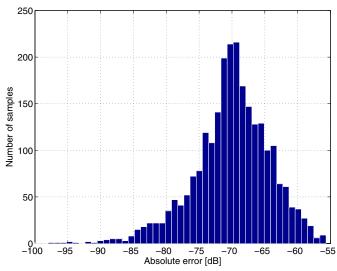

Fig. 12.   Histogram : absolute error distribution over 2500 validation samples

The method is applied to a scalable waveguide example, and its performance is analysed by numerical results. It is found that the method yields compact models with a fair accuracy.

ACKNOWLEDGMENT

Fig. 10.   Magnitude of complex model and reference data

Fig. 11.   Phase of complex model and reference data

REFERENCES

[1] S. Grivet-Talocia, S. Acquadro, S. Bandinu et al., "A Parameterization Scheme for Lossy Transmission Line Macromodels with Application to High Speed Interconnects in Mobile Devices", *IEEE Trans. on Electromagnetic Compatibility,* pp. 18-24, vol. 49, no. 1, Feb 2007.

[2] D. Deschrijver, T. Dhaene and D. De Zutter, "Robust Parametric Macromodeling Using Multivariate Orthonormal Vector Fitting", *IEEE Trans. on Microw. Theory and Techn.*, vol. 56, no. 7, pp. 1661–1667, Jul 2008.

[3] D. Deschrijver and T. Dhaene, "Stability and Passivity Enforcement of Parametric Macromodels in Time and Frequency Domain", *IEEE Trans. on Microw. Theory and Techn.*, vol. 56, no. 11, pp. 2435-2441, Nov 2008.

[4] T. Dhaene, D. Deschrijver, "Stable Parametric Macromodeling using a Recursive Implementation of the Vector Fitting Algorithm", *IEEE Microw. and Wireless Comp. Lett.*, vol. 19, no. 2, pp. 59-61, Feb 2009.

[5] P. Triverio, S. Grivet-Talocia and M. S. Nakhla, "A Parameterized Macromodeling Strategy with Uniform Stability Test", *IEEE Trans. on Adv. Packaging*, vol. 32, no. 1, pp. 205–215, Feb 2009.

[6] F. Ferranti, L. Knockaert, T. Dhaene, "Passivity-preserving Interpolation-based Parameterized Macromodeling of Scattered S-data", *IEEE Microw. and Wireless Comp. Lett.*, vol. 20, no. 3, pp. 133-135, Mar 2010.

[7] J. R. Koza, "Genetic Programming: on the Programming of Computers by Means of Natural Selection", MIT Press, MA (USA), 1992.

[8] M. Willis, H. Hiden, M. Hinchliffe, B. McKay and G. W. Barton, "Systems Modelling using Genetic Programming", Computers in Chemical Engineering, vol. 21, pp. S1161-1166, 1997.

[9] R. Lehmensiek, "Efficient Adaptive Sampling applied to Multivariate Multiple Output Rational Interpolation Models, with Applications in Electromagnetics-Based Device Modelling", PhD Thesis, Stellenbosch University, South Africa, 2001.

[10] Y. Leviatan, P. G. Li, A. T. Adams and J. Perini, "Single-post inductive obstacle in rectangular waveguide", IEEE Trans. Microwave Theory Techn., vol. 31, no. 10, pp. 806-811, Oct. 1983.

[11] D. Searson, "GPTIPS : Genetic Programming & Symbolic Regression for MATLAB", http://gptips.sourceforge.net, 2009.