Journal of Electronic Testing: Theory and Applications

High Level Fault Modeling of Analog Circuits Through Automated Model Generation Using Chebyshev And Newton Interpolating Polynomials

--Manuscript Draft--

Manuscript Number:	
Full Title:	High Level Fault Modeling of Analog Circuits Through Automated Model Generation Using Chebyshev And Newton Interpolating Polynomials
Article Type:	Manuscript
Keywords:	Automated Model Generation; High Level Fault Modeling; VHDL-AMS; Chebyshev polynomials; Newton polynomials; state-space; nonlinear transmission line
Corresponding Author:	Likun Xia, PhD University Teknologi PETRONAS Tronoh, Perak MALAYSIA
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	University Teknologi PETRONAS
Corresponding Author's Secondary Institution:	
First Author:	Likun Xia, PhD
First Author Secondary Information:	
Order of Authors:	Likun Xia, PhD
	Muhammad Umer Farooq, MSc
	Ian M Bell, PhD
Order of Authors Secondary Information:	

Powered by Editorial Manager® and ProduXion Manager® from Aries Systems Corporation

Dear editor and reviewers

The authors would like to declare that the results in this work have not been published in any other journals. The method was previously published in the 2013 ASP-DAC as highlighted in the reference session. The similarity between them is the methodology, but the result parts have never been published. The acceptance in ASP-DAC is the encouragement for us to go further, we are more confident about our work. Therefore, we have produced more results and decide to submit our work to JETTA.

*Manuscript
Click here to download Manuscript: JETTA.docx
Click here to view linked References

High Level Fault Modeling of Analog Circuits Through Automated Model Generation Using Chebyshev And Newton Interpolating Polynomials

Likun Xia, *Member, IEEE*, Muhammad Umer Farooq, Ian M. Bell, *Member IEEE*, Aamir Saeed Malik, *Senior Member, IEEE*

***Abstract*—With the help of automated model generation (AMG), high level modeling (HLM) of analog circuits is able to provide useful speedup and acceptable accuracy compared with standard SPICE-level circuit simulation. Unfortunately, this is not the case for high level fault modeling (HLFM) and high level fault simulation (HLFS). This is still the most critical issue that industry is facing in reducing analog testing cost. We present a novel algorithm using a fusion of Chebyshev and Newton interpolating polynomials (CNIP) in nonlinear state-space (ss) termed AMG-CNIP for HLFM in analog circuits. It is structured in MATLAB and the hardware description language (HDL) VHDL-AMS, respectively. The properties of AMG-CNIP are investigated by modeling nonlinear transmission line circuits using transient analysis. Results show that the AMG-CNIP models can handle both linear and nonlinear fault simulations with reasonable accuracy, and simulation speedup is achieved compared to standard SPICE-level simulations.**

Index Terms— Automated Model Generation, High Level Fault Modeling, VHDL-AMS, Chebyshev polynomials, Newton polynomials, state-space, nonlinear transmission line

I. Introduction

PROGRESS in silicon fabrication processes has drastically reduced feature sizes, allowing more transistors on a die, which enables larger chips and more complicated functionality from a single chip. Increased system complexity typically results in more challenging test requirements, particularly for analog and mixed signal (AMS) circuits on chips. To prevent shipping defective devices it is essential to develop and verify effective testing before the production stage.

Analog testing strongly influences time-to-market and time-to-profit since its costs dominate approximately 90% of the whole testing cost in modern AMS ICs [1],[2]. Unfortunately, analog test suffers from a lack of automated test pattern generation (ATPG) algorithms, long testing time and unknown test quality [3-8]. This is not the case in digital circuits testing, because several mature Design for Testability (DFT) [1] and Built-In-Self-Test (BIST) [2] methods exist.

To test a modern AMS circuit comprising millions of transistors, transistor level fault simulation (TLFS) is no longer a practical solution [9]. During the last few years, high level modeling (HLM) and high level fault modeling (HLFM) techniques have been proposed for verification and testing of complex AMS circuits, promising high speed [10-14] with reasonable accuracy. Initially the models were generated manually, e.g., [10], [15], [16], but this approach was a tedious process and often failed to capture critical nonlinear effects stemming from unanticipated interactions between blocks, or from second order phenomena in device models. Thus, automated model generation (AMG) approaches were proposed. These take a detailed description, typically a SPICE-level netlist, and generate a much simpler macromodel via an automated computational procedure [17].

A number of AMG approaches have been proposed for macromodeling of both weakly and strongly nonlinear analog circuits and system, e.g., weakly nonlinear methods including polynomial-based [18], strongly nonlinear methods including trajectory piecewise linear (TPWL) [19], piecewise polynomial (PWP) [20], scalable trajectory piecewise linear (STPWL) [21] and transfer function trajectory (TFT) [22].

However, none of these approaches were applied to HLFM at a system level. Their effectiveness in this context may be uncertain because of the additional requirements placed on models in a fault simulation context. For example, faulty behavior may force non-faulty subsystems into highly nonlinear regions of operation, which may not be covered by standard models, thus undermining fault propagation accuracy.

Xia et al. proposed an AMG approach termed multiple model generation system using delta operator (MMGSD) to perform HLM and HLFM at a system level [23]. It was the first time that an AMG approach was employed for HLFM. Certain level of accuracy was expected, but simulation speedup was not achieved because of computational overhead of model switching for which high level simulator is not optimized.

In this paper we present a method termed Automated Model Generation using Chebyshev and Newton Interpolating Polynomials (AMG-CNIP). A fusion of Chebyshev and Newton (CN) interpolating polynomial is developed to capture nonlinear effects of analog circuit. *Single* CN polynomial model in ss is needed in order to achieve speedup. The AMG-CNIP model is evaluated by performing HLM and HLFM of nonlinear transmission line circuits in a system level

simulator called *SystemVision*.

II. Previous Work and Background

In this section some existing AMG approaches to capture nonlinearity are reviewed.

A. Multiple Model Generation System Using Delta Operator (MMGSD)

MMGSD is the black-box AMG method that estimates a macromodel based on system input/output data only. It consists of two modules: an Automatic Model Estimator (AME) and an Automatic Model Predictor (AMP) [24]. The former observes error variation in output voltage against input voltage amplitude. If output error exceeds a pre-defined threshold, a region is created, in which a new linear model is generated. Each linear model is structured as RARMAX model structure modified with delta operator (δ) shown in equation (1).

$$\delta^m y(t) = -(a_1\delta^{m-1} + \ldots + a_m)y(t) + (b_0\delta^n + \ldots + b_n)u(t) \quad (1)$$

Delta operator is used to avoid aliasing effects at high frequency and transform discrete time RARMAX model into continuous time by shortening the sampling interval (delta) [24].

The process of splitting regions continues until the desired accuracy is achieved at the output. In this way multiple models are generated to capture nonlinear behavior. The AMP then loads model parameters from the AME and predicts responses for different inputs. After that, the models are converted to VHDL-AMS using a routine called multiple model conversion system using delta operator (MMCSD) [23] and run in the *SystemVision* software. During evaluation at the system level, the MMGSD VHDL-AMS model has to switch discretely among different linear models using '*if-else*' conditional checks. Discrete switching is well-suited for digital circuits where digital circuit simulators (e.g., *ModelSim* from *MentorGraphics*) can synthesize these conditions into structures such as tri-state buffers or multiplexers, whereas analog circuit simulators working at behavioral level are not mature enough to instantiate analog hardware. They are required to go through a huge number of Newton-Raphson iterations to solve circuit equations. Therefore, the models fail to achieve simulation speedup.

B. Trajectory Piecewise Methods

Trajectory piecewise methods are categorized in the white-box AMG class. White-box AMG macromodeling methods present analog circuit using nonlinear ss model structure as shown in equation (2).

$$q(\dot{x}(t)) = f(x(t)) + Bu(t)$$
$$y(t) = C^T x(t) \quad (2)$$

where $x(t) \in R^P$ is the state vector at time *t* representing the unknown node voltages and branch currents; $f: R^P \to R^P$ and $q: R^P \to R^P$ are nonlinear vector valued functions representing charge/flux and resistive terms in the circuit respectively, it can behave as nonlinear diode current equation or nonlinear transistor drain current equation etc.; *B* is a SS-dependent $P \times M$ input matrix that can be considered as constant if its value does not change with time; *u:* $R \to R^M$ is the input signal; *C* is a $P \times K$ output matrix and $y: R \to R^K$ is the output signal.

If nonlinear charge is linear function of state variable *x*(*t*) then equation (2) is reduced to ss of form shown in equation (3), where *E* is a constant matrix.

$$E\frac{d}{dt}x(t) = f(x(t)) + Bu(t)$$
$$y(t) = C^T x(t) \quad (3)$$

Trajectory piecewise AMG methods capture nonlinearities using Taylor polynomials obtained at different operating points from application specific training inputs. The authors of [19] employ purely linear functions (first order Taylor polynomials) in each piecewise region to capture nonlinear behavior, whereas the authors of [25] improve local approximation properties by utilizing second order Taylor polynomials in each region. Local approximation properties can also be improved by employing a scalable architecture with Gaussian Mixture Models (GMM) [21]. The GMM increases the likelihood of generating unreachable states in the ss, which are not excited during model training. The authors of [22] further improve scalability property of TWPL macromodels by proposing a method called Transfer Function Trajectory (TFT). This transforms state linearization matrices (obtained through training trajectories) into the frequency domain and casts the problem of finding non-reachable states as a regression fitting problem. They use vector fitting (VF) algorithms to construct a model for non-reachable states in the frequency domain.

The trajectory piecewise methods discussed above can accurately capture weak and strong nonlinear behaviors for analog circuits. This is similar to MMGSD in terms of using multiple models in that it can result in low computational efficiency while performing HLM and HLFM. The root cause of using multiple polynomials in ss are the poor approximation and interpolation properties of the Taylor series. Intuitively an efficient solution to avoid model switching would be the use of single high order Taylor polynomial. However, Taylor polynomials have poor convergence properties due to non-uniform approximation error distribution [26]. Approximation errors from Taylor polynomials are small as long as the input stays close to the expansion point, but increase rapidly away from the expansion point. This means that for inputs far from expansion points, even higher order Taylor polynomials cannot accurately predict system responses [28].

The situation depicted in Fig. 1 explains why *multiple* models are required instead of *single* model to capture nonlinear circuit response (time vs. amplitude), where solid line (above) is circuit input and solid line (below) is the circuit response. To accurately approximate the complete circuit response, low order Taylor polynomials are generated at multiple linearization points (black dots), each covering a small region of the system response. A given polynomials will not be useful outside its own region, and new models are required for

other regions and linearization points. However, a *single* low order Taylor polynomial approximation is sufficient for relative small input/output signals (typical of linear small signal modeling), but such an approximation becomes unacceptably poor for larger input amplitudes.

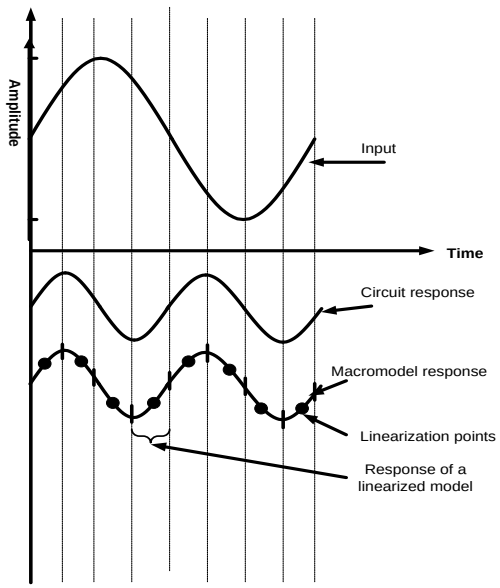


Fig. 1. Multiple models using Taylor polynomials

It is important to notice that the approximation and interpolation properties of a polynomial in ss are interrelated. To correctly predict the response for inputs that are different from the training inputs, a polynomial must have good interpolation properties. This enables it to successfully interpolate between excited states, which were covered during macromodel training, in order to deal with non-excited states that appear during macromodel evaluation. The quality of a macromodel's response for non-excited states mainly depends on the interpolation properties of the polynomial used for the linearization of nonlinear components in the circuit.

Our aim is to reduce model switching due to its impact on the efficiency of high level model execution. Ideally the model would use single polynomial for approximation of the complete trajectory. This implies use of higher order polynomials than those employed in the MMGSD and trajectory piecewise methods described above. Any such a polynomial must have good approximation and interpolation properties for inputs that are large and far from the training input.

There are a number of candidate interpolating polynomials such as Chebyshev polynomials, Newton polynomials, Lagrange polynomials, Hermite polynomials. Chebyshev polynomials have shown good interpolation and approximation properties due to their discrete orthogonality property and minimax approximation of a function [26][28]. To clarify the rationale here, a numerical example is provided using a nonlinear function $f(x) = \frac{x^2 \exp^{-\frac{x^2}{2}}}{\sqrt{2\pi}} + x\sin(2\pi x)$, where x is input and has value of x = -2.5:0.05:2.5. A plot of input x versus Taylor polynomial responses of different orders and Chebyshev polynomial response is shown in Fig. 2.

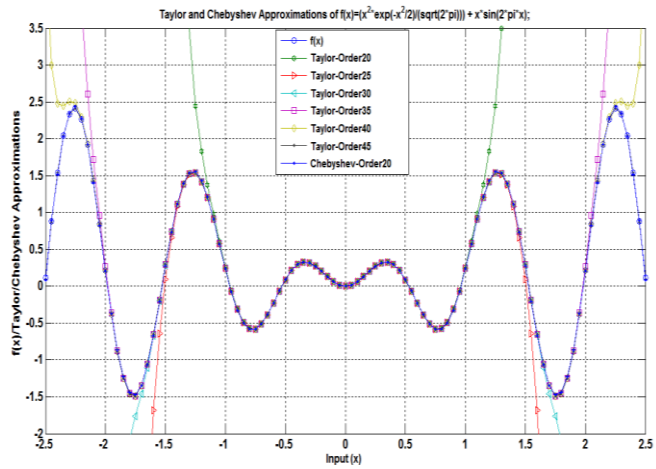


Fig. 2. Comparison of convergence properties of Taylor and Chebyshev polynomials [27]

One can observe from Fig. 2 that as long as input x is close to the expansion point at the origin (within the range [-1, 1]), the fit for all Taylor polynomials is good. As x becomes larger, the Taylor polynomials start diverging from the original function. This indicates that single Taylor polynomial is only effective as long as input magnitude is small and approximation is poor for large input magnitudes. Although a Taylor polynomial of order 45 fits the complete response it is impractical to use such a high order polynomial in ss to model an analog circuit. However a single Chebyshev polynomial of order 20 completely fits the curve. This demonstrates that Chebyshev polynomials have better approximation properties than Taylor polynomials. Therefore, Chebyshev polynomials in a nonlinear ss model structure are used in the paper.

III. Automated Model Generation Algorithm using Chebyshev and Newton Interpolating Polynomials (AMG-CNIP)

AMG-CNIP consists of two parts: model generation and model evaluation [27] as illustrated in Table 1.

TABLE 1 AMG-CNIP ALGORITHM FOR MODEL GENERATION AND EVALUATION

Algorithm 1 AMG-CNIP algorithm for model generation, HLM and HLFM

1. **Model Generation**
2. Formulate circuit KCL equations in ss form.
3. Simulate original circuit with training input and automatically measure range (to use for CN polynomial generation) using ARF.
4. **for all** 1D nonlinear component **do**
5. Apply 1D formulation to obtain *single* CN polynomial for all single ended nonlinear components.
6. **end for**

7. **for all** 2D nonlinear components **do**
8. Apply 2D formulation to obtain *single* CN polynomial for all nonlinear components with two node connection.
9. **end for**
10. Employ 1D and 2D CN polynomials in ss to obtain AMG-CNIP macromodel (7).
11. Given training input, simulate AMG-CNIP model (8) to tune CN polynomial order (N) using APT algorithm given in Table 1.

12. **Model Evaluation**
13. Given an evaluation input waveform and AMG-CNIP model parameters
14. **for** all time steps **do**
15. Interpolate CN polynomial in ss
16. Solve ss equations to obtain output *y*
17. **end for**

The key part of AMG-CNIP is to generate and tune CN polynomial for ss as shown in Fig. 3. The polynomial starts with the selection of an arbitrary value of order *N*, it is then tuned using the algorithm termed Automatic Polynomial Tuner (APT) shown in table 2 in order to obtain an optimal AMG-CNIP macromodel. Therefore, unnecessarily high orders are avoided that may affect the simulation speed. Where $y(t)$ is the original circuit output for a set of training inputs and γ a pre-defined error tolerance.

The one-dimension (1D) and two-dimension (2D) mathematical formulations are then required to obtain CN polynomials for the nonlinear function $f(x)$. The dimensions are dependent on the requirement of nodes, i.e., 2D is needed when a nonlinear component is connected to two different nodes, otherwise 1D is required.

1D CN polynomial for single ended components can be obtained through a three-step procedure described below.

1. To obtain a set of input data points x'_k using Chebyshev node formula given in equation (4) [29].

$$x'_k = \frac{b-a}{2}\cos\frac{2N+1-2k}{2(N+1)}\pi + \frac{a+b}{2} \tag{4}$$

   where *a* is the lower limit, *b* is the upper limit of the range [*a*,*b*] of nonlinear function *f*(.), *k=0,1,2,…,N*, *N* is the degree of polynomial.
2. To use Chebyshev nodes x'_k obtained in step 1 as inputs to original nonlinear function *f*(.) and calculate corresponding target points $f(x'_k)$. Thus to obtain a set of interpolation points (x'_k , $f(x'_k)$).
3. Use the points (x'_k , $f(x'_k)$) obtained in step 2, 1D NDD method [29] is applied to achieve the polynomial in the form shown in equation (5).

$$n_N(x) = a_0 + a_1(x-x'_0) + a_2(x-x'_0)(x-x'_1) + \ldots + a_n(x-x'_0)\ldots(x-x'_N) \tag{5}$$

   where $a_0, a_1, \ldots, a_N$ are coefficients of the polynomial calculated using NDD; x'_k are the Chebyshev nodes from step 1.

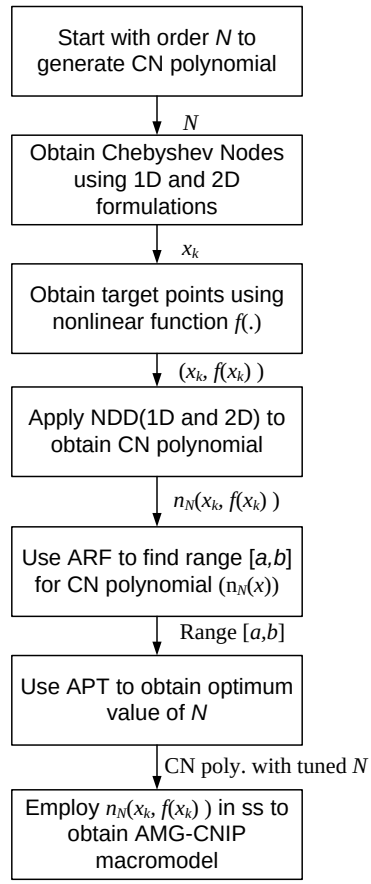


Fig. 3. AMG-CNIP model generation flow

TABLE 2

ALGORITHM FOR AUTOMATIC POLYNOMIAL TUNER (APT)

Algorithm 2: Algorithm for Automatic Polynomial Tuner (APT)

1. Start with arbitrary order *N*. The initial value is usually set in range of 1 to 3;
2. Select training input waveform;
3. **for** each transient time step **do**
4. Compute input *u(t)* for current time *t*;
5. Calculate macromodel output ($y'(t)$) using equation (7));
6. Measure absolute error $\varepsilon = |y'(t) - y(t)|$
7. **If** $\varepsilon > \gamma$
8. Increment *N* by 1;
9. Apply three-step procedure (1D and 2D) with new *N* to re-calculate CN polynomial.
10. **else**
11. **If** transient time finish
12. Exit
13. **else**
14. Increment transient time step *t* and go to 4.
15. **end if**
16. **end if**
17. **end for**

Similarly 1D polynomial for two-ended components can be obtained through three-step procedure described below.

1. Calculate Chebyshev nodes (x'_k, y'_k) for two voltages (x,y) using equation (6).

$$x'_k = \frac{b-a}{2}\cos\left(\frac{2N+1-2k}{2(N+1)}\pi\right)+\frac{a+b}{2}$$
$$y'_k = \frac{b-a}{2}\cos\left(\frac{2N+1-2k}{2(N+1)}\pi\right)+\frac{a+b}{2} \qquad (6)$$

2. Calculate target nodes using set of Chebyshev nodes (x'_k, y'_k) as input to nonlinear function f(.) and obtain a set of interpolation points ((x'_k, y'_k),f(x'_k, y'_k)) to obtain CN polynomial.
3. Using a set of interpolation points obtained in the last step; apply a 2D NDD [29] method to achieve the 2D CN polynomial of the form shown in equation (7).

$$n_N(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{n-i} a_{i,j} x'^i y'^j \qquad (7)$$

To automatically find the range [a,b] for nonlinear function $f(x)$, a simple procedure termed automatic range finder (ARF) is applied.

1. Simulate original circuit with training input and store state vector $x(t)$.
2. Simulate nonlinear current function $f(x(t))$ with state vector $x(t)$ as its input.
3. Find minimum and maximum values of current $f(x(t))$ and apply as [a,b] range for both 1D and 2D formulation given in subsections III.A and III.B.

The CN polynomial based on [a,b] will automatically cover lower value cases for all nonlinear components in the circuit. With this approach, one does not need to compute different CN polynomial for each nonlinear component and single CN polynomial can be used for linearization of all nonlinear components in ss.

1D and 2D polynomial with range [a,b] can be employed in nonlinear ss model structure shown in equation (8).

$$E\frac{d}{dt}x(t) = n_N(x(t)) + bu(t)$$
$$y(t) = c^T x(t) \qquad (8)$$

IV. Automatic Model Conversion (AMC)

To perform HLM and HLFM at the system level using AMG-CNIP macromodels, one needs to 'translate' AMG-CNIP mathematical macromodel in MALTAB environment into an electrical circuit using VHDL-AMS at the *Systemvision* software. AMC employ VHDL-AMS built-in attribute '*integ*' [31] (shown in Fig. 4) to solve ss differential equation in equation (8).

```
--CN polynomial parameters
constant a_1d_1: real := 5.571831423275393e-09;
constant a_1d_2: real := 3.353644694695958e-10;
constant a_1d_3: real := 7.273126596624195e-16;
begin
temp1=
(-((a_1d_1*v1**2)+(a_1d_2*v1)+a_1d_3)-(a_1d_1*(v1-v2)**2)+
((a_1d_2*(v1-v2))+a_1d_3) – (2.0*v1)+v2+i_in);
```

Fig. 4. Highlight of AMG-CNIP HLM

V. Experimentation Results

A. Evaluation of AMP-CNIP using Single Model Assumption

Interpolation and approximation properties of Taylor and CN polynomial macromodels for *single* model assumption are compared using a nonlinear transmission line circuit shown in Fig. 5. Single model is carried out for 1D and 2D polynomials, and results from AMG-CNIP are compared with TPWL with proper orthogonal decomposition (POD) projection (TPWL-POD).

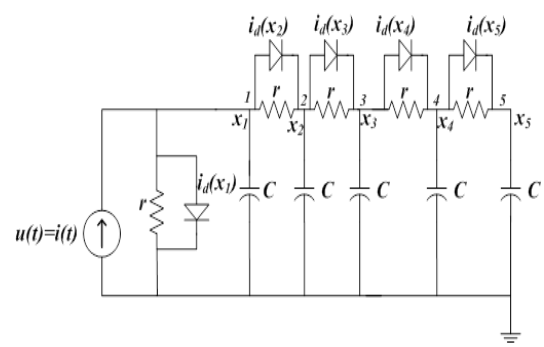


Fig. 5. Schematic of nonlinear transmission line circuit (NLTx) with $i_d(x) = \exp(40x) - 1$

It consists of resistors, capacitors and diodes with a simplified current model. All resistors and capacitors are set at unit value, i.e., R=C=1. The input is a current source entering node 1, $u(t) = i(t)$; the single output is the voltage at node N, $y(t) = x_N(t)$.

Initially a simple case of 1D polynomial is considered, i.e., one diode, resistor and capacitor in the circuit. As the diode is connected to ground, diode nonlinear current $f(x)$ is a function of only one node voltage across it. A training input shown in equation (9) is employed.

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) - \ldots a_{na} y(t-na) + b_1 u(t-1) + b_2 u(t \qquad (9)$$

To cover a wide range of signals, values of parameters in equation (9) are randomly selected, e.g., *Amp* = [0.1,0.4,0.5,0.7,1.0]; *freq* = [10,50,100,200,500,1000] and θ= [0,π/4,π/2,π,2π]. Training waveform obtaind using these parameters is shown in Fig. 6.

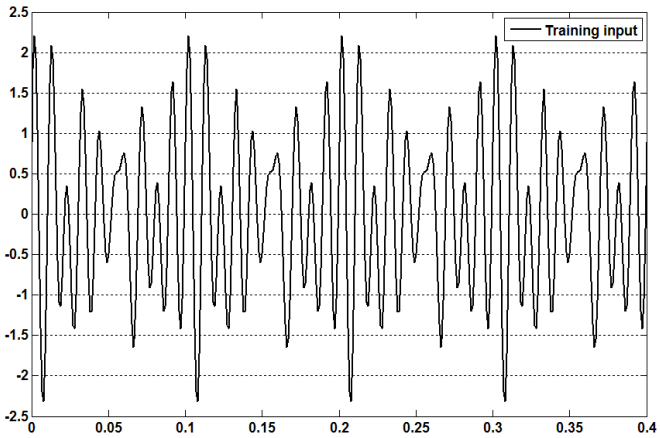


Fig. 6. Training input waveform

Based on the input, the models are generated. In AMP-CNIP macromodel ARF is [-0.8, 1.5797], APT tunes CN polynomial order to 6 with error tolerance γ =1.0e-3; for 1D Taylor polynomial macromodel (TAM), the order is 6; for TPWL-POD the order q is set to 1, since there is only one node in the circuit. Eight linearization points are generated during model training to approximate complete trajectory.

For evaluation, two test waveforms are used based on equation (9): one has *Amp*=±2.0A that is ±0.5A less than training input waveform; another has *Amp*=±5.0A that is ±2.5A larger than training input waveform. Values of other parameters are set to *freq*=250Hz and θ=3π/4.

Initially the first waveform is applied with a transient simulation of 400ms (with sampling rate of 1*u*s) is run and the outputs are illustrated in in Fig. 7 but only the last 100ms is displayed.

It is seen that results from AMG-CNIP and TPWL-POD macromodels are indistinguishable from original circuit response. TAM macromodel does not achieve the same. Regarding the evaluation time AMG-CNIP macromodel complete simulation in approximately 500ms, whereas TPWL-POD takes 850ms to complete simulation due to model selection process.

Then the second waveform is used. The simulation result is shown in Fig. 8. It is seen that AMG-CNIP response produces better accuracy than both TPWL-POD and TAM due to its better approximation and interpolation properties.

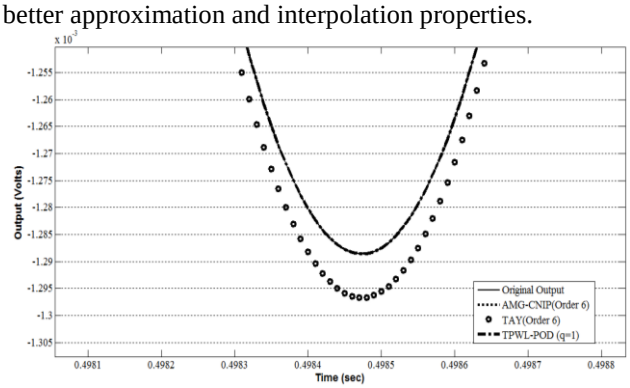


Fig. 7. Transient response of 1D polynomial macromodels for input waveform with *Amp*= ±2.A (last 100ms)

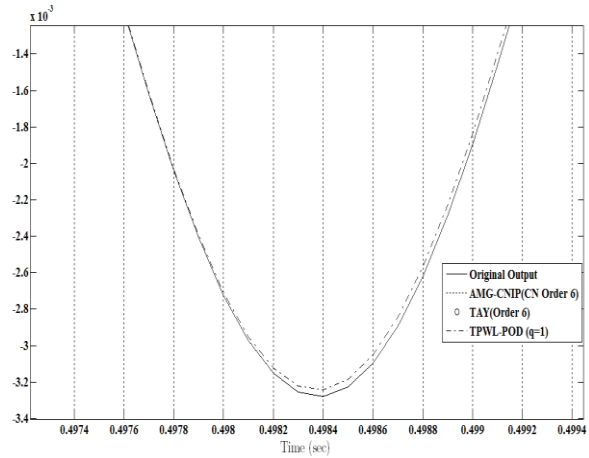


Fig. 8. Transient response of 1D polynomial macromodels for input waveform with *Amp*=±5.0A (last 100ms)

To perform single model assumption for 2D case, five nodes are used in nonlinear transmission line circuit. The same training input shown in Fig. 6 is used. For AMG-CNIP APT tunes the CN polynomial order to 10 with error tolerance γ =1.0e-3. CN polynomial order is high in this case as there are more diodes in the circuit and they significantly contribute to nonlinear behavior of circuit. For 2D TAM, the order of 10 is required; in TPWL-POD the order q is 3. Twenty linearization points are obtained along the training trajectory. The output voltage is taken at node 1.

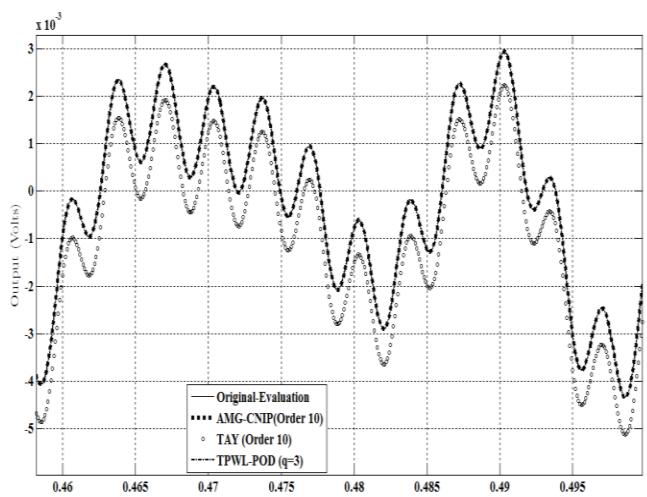


Fig. 9. Transient response of 2D polynomial macromodels for input with *Amp*= ±2A (last 100ms)

The same approach and waveforms as for 1D is performed for 2D. A transient simulation of 500ms is run for first waveform and simulation result is highlighted in Fig. 9. AMG-CNIP and TPWL-POD macromodels shows the better accuracy than TAM. AMG-CNIP macromodel performs faster (2.5s) than TPWL-POD (3.01s).

While the second stimulus is used only AMG-CNIP can achieve the accurate results as illustrated in Fig. 10.

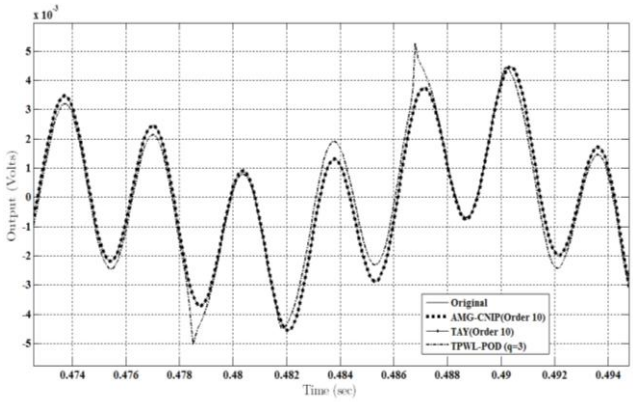


Fig. 10. Transient response of 2D polynomial macromodels for input waveform with *Amp*=±5A (last 100ms)

B. HLM using AMG-CNIP Macromodels

In this part, AMG-CNIP algorithm is compared with the MMGSD [23] in order to approve the advantages as compared to state-of-art. Both AMGs (AMG-CNIP and MMGSD) are initially evaluated in MATLAB for accuracy and speed up comparison using NLTx line circuit and then HLM is performed.

To perform HLM, two benchmark circuits are used: the nonlinear transmission line circuit (NLTx) shown in Fig. 5 with nonlinear diode current function *f(x)* obtained from SPICE2G6 [30]; a 3-stage cascaded NLTx circuit seen in Fig. 11, where the operational amplifiers (opamps) are functioned as voltage to current conversion and resistors R_g are used to set the current gain at opamp current output.

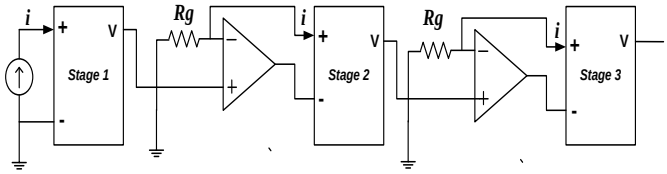
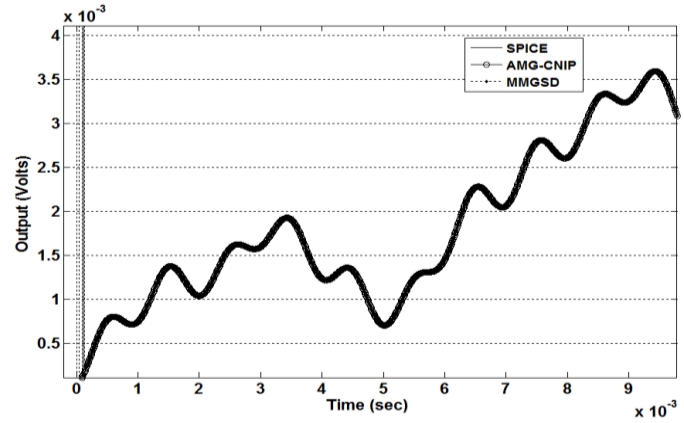


Fig. 11. 3-stage cascaded nonlinear transmission line circuit

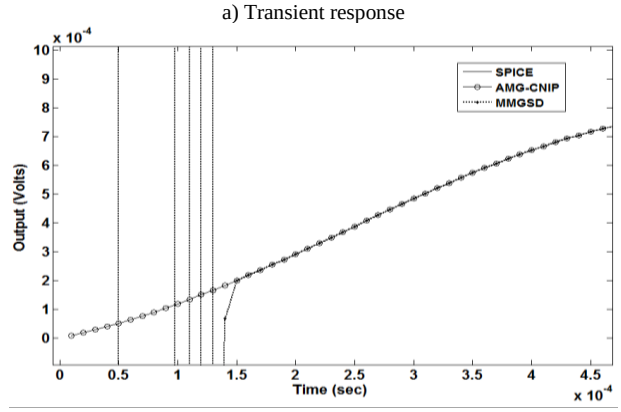
Macromodels for MMGSD and AMG-CNIP are generated using the training waveform shown in Fig. 6. MMGSD require 3 models to achieve reasonable accuracy at output. For the AMP-CNIP, APT module tunes polynomial order *N* to 3. ARF find range for nonlinear function *f(x)* to be [-0.003, 0.008] with γ =1.0e-3.

The stimulus for evaluation is the sine waveform with the amplitude of *Amp*=±2.0A, *freq*=250Hz and θ=3π/4s. The transient analysis is performed for 10ms (with sampling rate of 1*us*). Simulation result is shown in Fig 12.

It illustrates that AMG-CNIP response is settled from the start of simulation and can predict correct response, whereas MMGSD macromodel starts predicting the correct response at approximately 150us. During this time, response tends towards infinity (output voltage is ≈ $7\text{x}10^{24}$ volts). Hence computational overhead involved here for MMGSD is to attain stable state due to model switching until the best model is selected. MMGSD takes 150us to stabilize and predict correct response, whereas AMG-CNIP is stable from beginning. This indicates that AMG-CNIP is faster to predict correct response as compared to MMGSD.



a) Transient response



b) Zoomed waveform (Initial 156*us*)

Fig. 12. MMGSD and AMG-CNIP responses for NLTx line circuit

To perform HLM, AMG-CNIP and MMGSD macromodels are initially converted into VHDL-AMS behavioral models using the AMC routine in section IV. HLMs are then tested in different configurations (Table 3) using circuits shown in Fig. 5 and Fig. 11.

TABLE 1 CONFIGURATIONS FOR AMG-CNIP HLM TESTING

Configuration No	Settings
Config 1	NLTx line
Config 2 (3-stage Cascaded)	SPICE – HLM – SPICE
Config 3 (3-stage Cascaded)	HLM – SPICE – HLM
Config 4 (3-stage Cascaded)	SPICE – SPICE – HLM
Config 5 (3-stage Cascaded)	HLM – HLM – HLM
Config 6 (3-stage Cascaded)	HLM – SPICE – SPICE

In Config1, NLTx line interfaces with input current source only. There are total of 15 nodes in three stage cascaded nonlinear transmission line circuit and output voltage can be taken at any node while performing system level simulations. A

transient simulation of 200s is run in SystemVision. The stimulus is a current sine wave at 500Hz frequency and ±2.0A amplitude. AMG-CNIP HLM, MMGSD HLM and SPICE circuit are run simultaneously and the simulation results are illustrated in Fig. 13, but signals at only last 0.1s are displayed.

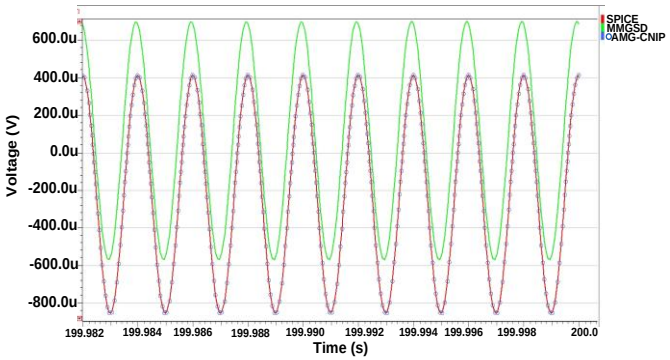


Fig. 13. Transient response of MMGSD and AMG-CNIP for Config1 circuit

It is seen that MMGSD HLM (mmgsd_hlm) fails to perform HLM correctly, whereas AMG-CNIP HLM response (v1_hlm) matches the original SPICE response (v1_spice). To measure relative error in percentage equation (10) is used, where *i* is the number of samples obtained from SystemVision simulation. Relative error percentage for AMG-CNIP for Config1 circuit is 5.49e-8%, which is almost negligible.

$$\mathrm{Re}l.error = \frac{\sum_i \left| \frac{y_SPICE_i - y_AMG_i}{y_SPICE_i} \right|}{length(y_SPICE)} \times 100$$

(10)

Simulation speed up is crucial and is calculated using equation (11). In this case AMG-NIP achieves a simulation speedup of 1.96x compared to standard SPICE circuit simulation.

$$Simulation_Speedup = \frac{Mean(SPICE_CPU_time)}{Mean(AMG-CNIP_CPU_time)}$$

(11)

For the system level testing, we start with Config2, where second stage of SPICE circuit is replaced with AMG-CNIP HLM. A transient simulation of 15s is performed. Simulation results can be found in Fig. 14. Similar input signal is given to the circuit as for Config1 and output voltage is taken at node 6 of 3-stage cascaded circuit (which is first output of HLM). v6_spice is full SPICE circuit response and v6_hlm is Config2 circuit response. It is seen that AMG-CNIP achieves good accuracy with relative error of 0.72%. Mean CPU time of full SPICE circuit for 30 runs is 28.3s and mean CPU time of Config2 circuit for 30 runs is 25.26s. Hence a simulation speedup of 1.12x is achieved compared to full SPICE circuit.

In Config3 circuit the first and last stages are replaced with AMG-CNIP HLM. The same input and analysis as Config2 is performed. Outputs taken at node 5 and node 11 are shown in Fig. 15.

It is seen that Config3 circuit with two HLMs in circuit is working with reasonable accuracy. The relative error (measure at node 11) is 0.71%. Average CPU time for Config3 is 12.75s and SPICE full circuit average simulation time is 28.3s. Hence, a simulation speedup of 2.2x is achieved.

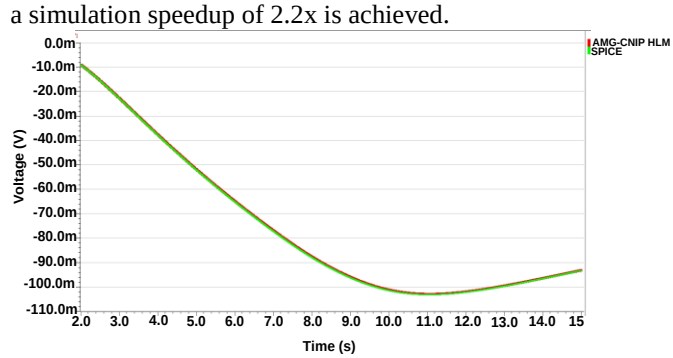


Fig. 14. Transient response of AMG-CNIP HLM for Config2

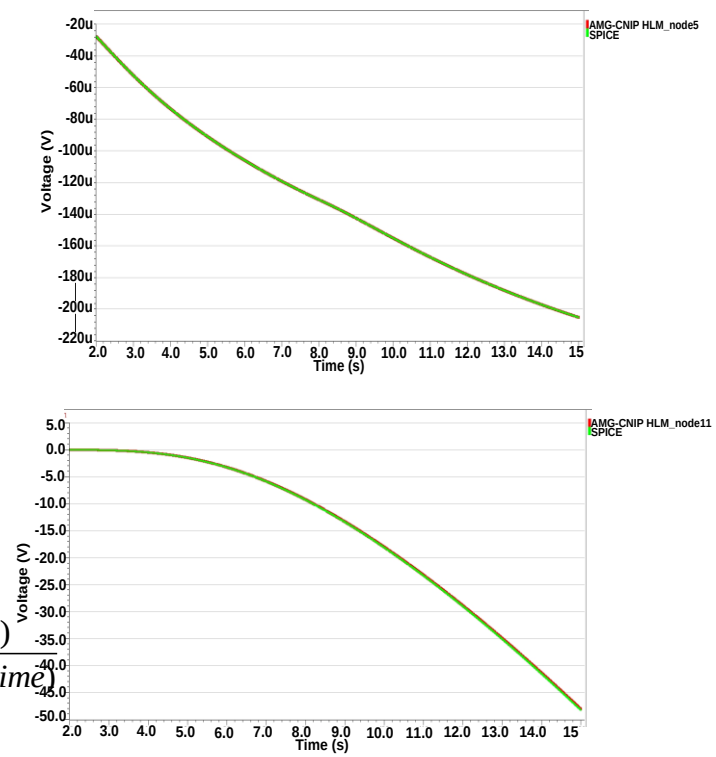


Fig. 15. Transient response of AMG-CNIP HLM for Config3 (node 5 and 11)

In Config4 we replace the last stage of cascaded nonlinear transmission line circuit with the AMG-CNIP HLM. Output voltage is taken at nodes 11, 12 and 13, respectively. Simulation results are shown in Fig. 16. It is noted that output voltage at node 11 reaches to -45 volts. Similarly voltages at node 12 and 13 can reach to -30 volts and -21 volts, respectively and AMG-CNIP HLM is able to predict responses accurately. Above responses shows excellent interpolation and approximation properties of CN polynomials because during training, macromodel has been trained to predict output

responses that swing within the range of millivolts. Whereas, in this configuration output voltages reach to unexpected voltage (-45 volts) and AMG-CNIP HLM can still achieve the acceptable results. Relative errors measured at nodes 11, 12 and 13 are 0.0085%, 0.863%, and 1.12% accordingly. The average CPU time for Config4 circuit is 25.31s. The simulation speed up achieved is 1.11x.

Moreover, the similar approach is used for Config5, where all SPICE modules are replaced with AMG-CNIP HLM. The outputs at node 5, 10 and 15 show the correct response can be achieved even with little output error in preceding stage as seen in Fig. 17. The relative error measured using AMG-CNIP HLM at node 5 is 8.4e-7%, at node 10 is 0.68% and at node 15 is 1.54%. Average CPU time for Config5 circuit is 17s and speedup achieved is 2.62x. It shows that by using more HLMs in the circuit, simulation speedup can be achieved.

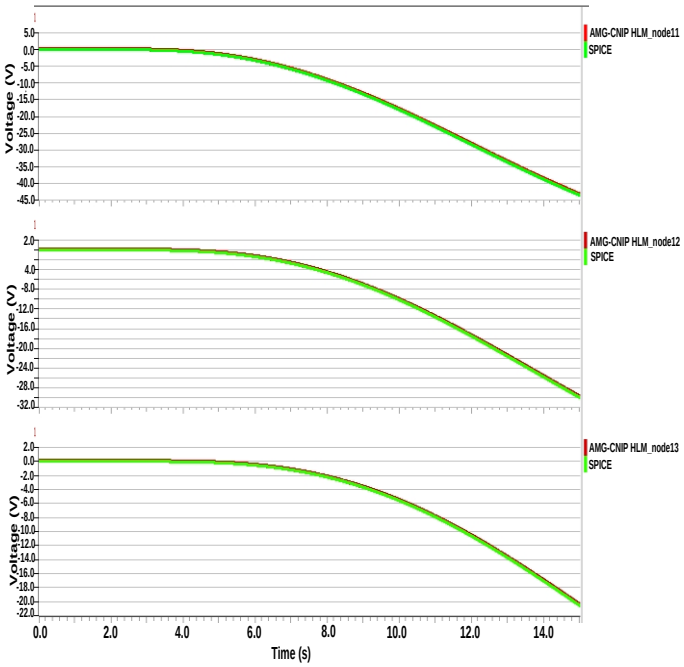


Fig. 16. Transient response of AMG-CNIP HLM for Config4 (nodes 11,12 and 13)

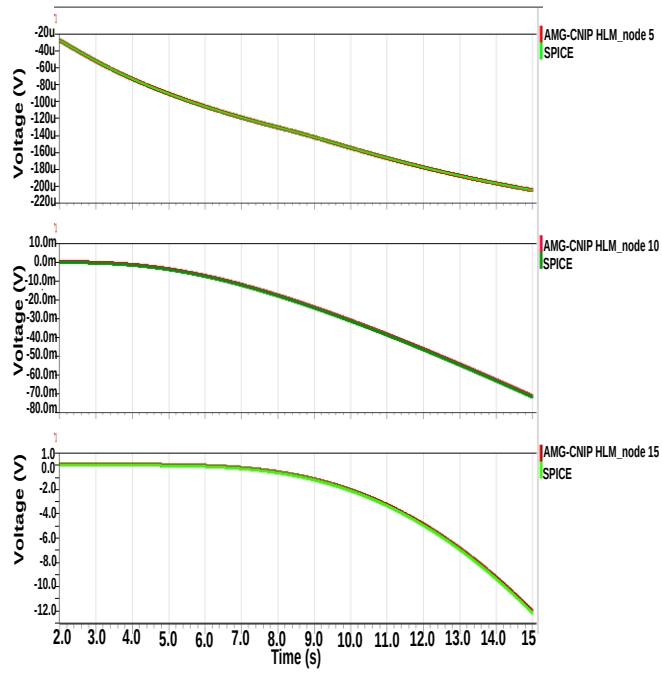


Fig. 17. Transient response of AMG-CNIP HLM for Config5 (nodes 5, 10 and 15)

Complete summary of simulation speed up and accuracy for six different configurations of AMG-CNIP HLM is provided in Table 4.

C. HLFM for Short Fault at Diode D1

In this section HLFM is performed by manually adding a short resistor (r_s) of 1$u\Omega$ across a diode in NLTx circuit. One fault is introduced at a time in the circuit. The diodes used are D1 and D2 in six configurations as shown in Table 5.

TABLE 2 CONFIGURATIONS FOR AMG-CNIP HLFM TESTING (SHORT FAULT AT D1)

Configuration No.	Settings
Config 1	NLTX line
Config 2 (3-stage Cascaded)	HLFM – SPICE – SPICE
Config 3 (3-stage Cascaded)	SPICE – HLFM – SPICE
Config 4 (3-stage Cascaded)	SPICE – SPICE – HLFM
Config 5 (3-stage Cascaded)	HLM – HLFM – SPICE
Config 6 (3-stage Cascaded)	HLM – HLM – HLFM

Initially a short fault is introduced at diode D1. A 3rd order CN polynomial is generated for AMG-CNIP macromodel and the range for nonlinear function *f(x)* is [-0.0001, 0.0001]. The MMGSD models provide reasonable accuracy and require about 158us settling down, whereas AMG-CNIP response is settled from the start of simulation (Fig. 18).

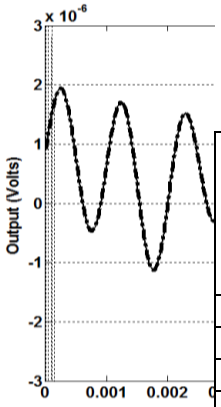
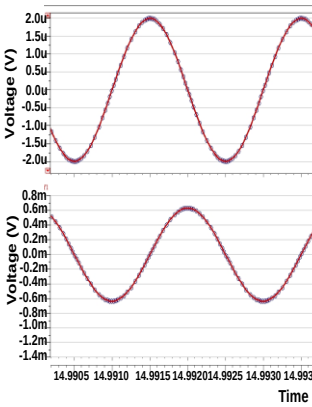


Fig. 18. MMGSD and AMG-CNIP response for NLTx line faulty circuit

Configuration Fault Free circuit	SPICE (s)		AMG-CNIP HLM (s)		Speed up $\frac{SPICE}{AMG_CNIP}$	Accuracy Rel. error (%)
	Mean	Max	Mean	Max		
NLTX line	15.65	30.2	8.0	30.3	1.96x	5.49e-8
SPICE_HLM_SPICE	28.3	52.2	25.26	45.8	1.12x	0.72
HLM_SPICE_HLM	28.3	52.2	12.75	20.6	2.2x	0.71
SPICE_SPICE_HLM	28.3	52.2	25.32	47.8	1.11x	1.12
HLM_HLM_HLM	28.3	52.2	10.8	17	2.62x	1.54
HLM_SPICE_SPICE	28.3	52.2	12.68	30	2.23x	3.7e-7

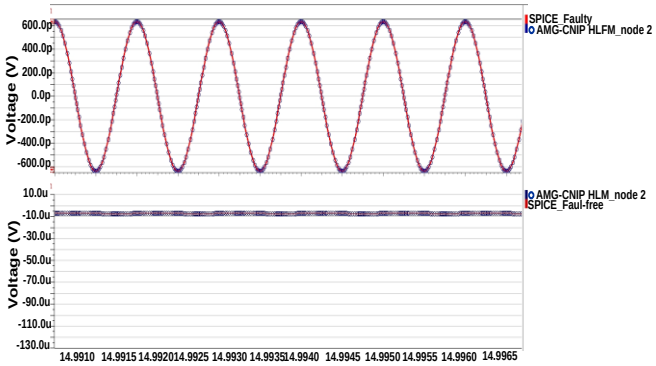
TABLE 3 SUMMARY OF SIMULATION SPEED UP AND ACCURACY FOR AMG-CNIP HLM



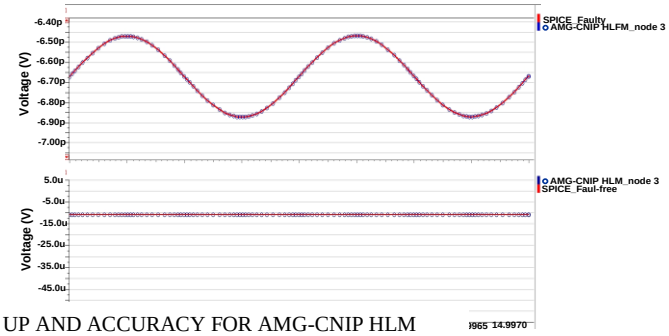
a Transient response of AMG-CNIP HLFM for Config1 at node 1

The HLFM is then performed using Config1 circuit in SystemVision with the transient analysis of 15s. An input sine waveform is applied with 500 Hz frequency and ±2.0A current amplitude.

The MMGSD model fails to perform HLFM. The simulation result of AMG-CNIP for Config1 circuit (last 100ms) is shown in Fig. 19. The outputs at nodes 1, 2 and 3 are displayed. The fault-free responses at nodes 1, 2 and 3 are also shown. It is noted that AMG-CNIP HLM cannot be employed to predict correct response for faulty circuit as difference between faulty and fault-free responses is significantly different as shown in Fig. 19. It is seen that voltage at node 1 (Fig. 19(a)) swings between -1.2mV and 0.6mV for fault-free circuit (lower waveform), whereas voltage at node 1 swings between -2uV and 2uV for faulty circuit (upper waveform). Similarly, the difference in amplitudes is obvious at node 2 and 3 (Fig. 19(a), 19(b)) for fault-free and faulty circuit responses. Hence, cannot simply use HLM for faulty circuit, one needs to generate faulty model to perform HLFM at system level.



b Transient response of AMG-CNIP HLFM for Config1 at node 2



c Transient response of AMG-CNIP HLFM for Config1 at node 3

Fig. 19. Transient response of AMG-CNIP HLM and HLFM for Config1 (nodes 1, 2 and 3) simulation result

Configuration D1 Faulty	SPICE (s)		AMG-CNIP HLFM (s)		Speed up $\frac{SPICE}{AMG_CNIP}$	Accuracy Rel. error (%)
	Mean	Max	Mean	Max		
NLTX line	16.1	20.5	7.3	11.8	2.2x	3.4e-12
HLFM_SPICE_SPICE	32.7	36.3	12.1	25.6	2.7x	3.8e-7
SPICE_HLFM_SPICE	29.96	33.5	27.3	31.7	1.09x	0.001
SPICE_SPICE_HLFM	30.2	38	27.5	36.6	1.08x	0.01
HLM_HLFM_SPICE	29.96	33.5	12.1	16.8	1.78x	0.09
HLM_HLM_HLFM	30.2	38	10.59	16.1	2.85x	0.01

TABLE 4 SUMMARY OF SIMULATION SPEED UP AND ACCURACY FOR AMG-CNIP HLFM (FAULT AT DIODE D1)

Relative error for Config1 measured at node 1 is 3.4e-12, at node 2 is 1.28e-6 % and at node 3 is 4.21e-10 that is almost negligible. Average CPU time (for 30 runs) of AMG-CNIP HLFM is 7.3s and average CPU time of SPICE circuit is 16.1 s. Simulation speedup achieved is 2.2x.

Moreover, we replace middle and last stage with AMG-CNIP HLFMs respectively in Config3 and Config4. A transient simulation of 15s is performed and output is taken at nodes 10 and 11 for Config3 and at nodes 11 and 12 for Config4. Simulation results are shown in Fig. 20 and Fig. 21 respectively. Speedup achieved for Config3 is 1.09x and speed up achieved for Config4 is 1.08x as compared to full SPICE faulty circuits.

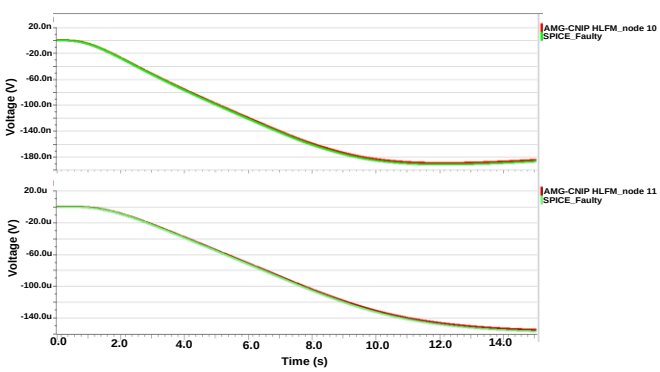


Fig. 20. Transient response of AMG-CNIP HLFM for Config3 (nodes 10 and 11)

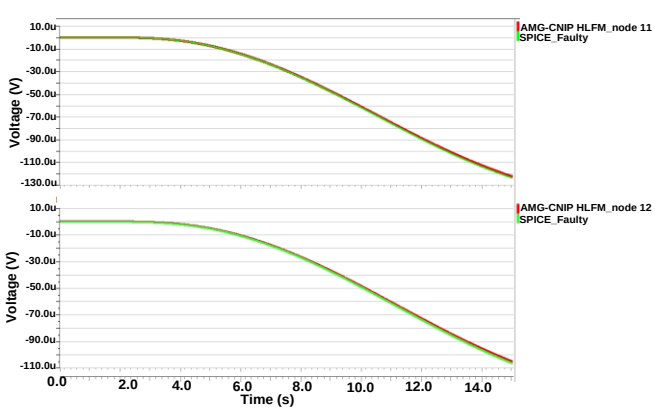


Fig. 21. Transient response for AMG-CNIP HLFM for Config4 (nodes 11 and 12)

It is noted that simulation speedup achieved in Config2 (2.7x shown in Table 6) is larger than in latter two cases (Config3 and Config4), i.e., 1.09x and 1.08x, although single stage is replaced in all cases (Config2, Config3 and Config4). Config2 and Config1 show almost similar speedups. In Config2 HLFM performs with similar operating conditions to Config1 (i.e., both directly interface with input current source and configuration is similar to training configuration), whereas in Config3 and Config4, operating conditions are different for HLFM as they are interfaced with SPICE circuit instead of input current source. Therefore, HLFMs are evaluated more rigorously in these configurations and thus show less speedup with showing reasonable accuracy. Complete summary of all configurations for HLFM of fault at D1 is given in Table 6.

D. HLFM for Short Fault at Diode D2

In second experimentation we introduce a fault at diode D2 and test HLFM in six configurations shown in Table 5. AMG-CNIP requires 3rd order CN polynomial with ARF range for nonlinear function *f(x)* to be [-1e-6, 1e-6]. MMGSD needs approximately 154us of settling time and AMG-CNIP macromodel response is settled from beginning. Ignoring initial 155us, the response of AMG-CNIP and MMGSD for evaluation input is shown in Fig. 22.

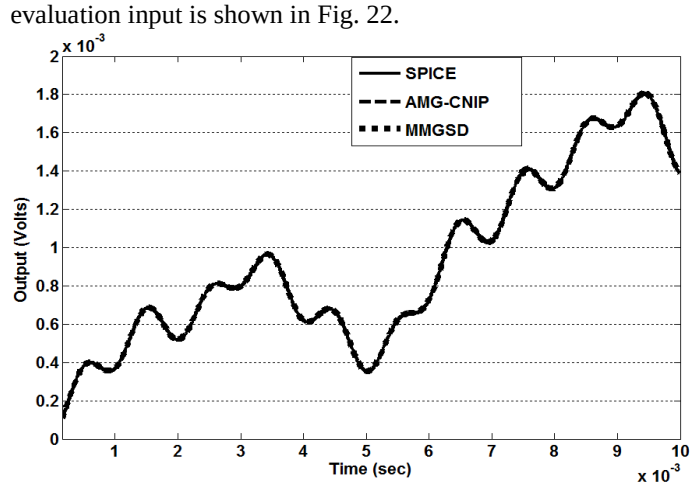


Fig. 22. MMGSD and AMG-CNIP response for Config1 fault at D2

Table 5 SUMMARY OF SIMULATION SPEED UP AND ACCURACY FOR AMG-CNIP HLFM (SHORT FAULT AT DIODE D2)

Configuration D2 Faulty	SPICE (s)		AMG-CNIP HLFM (s)		Speed up $\frac{SPICE}{AMG_CNIP}$	Accuracy Rel. error (%)
	Mean	Max	Mean	Max		
NLTX line	17.2	21	12.4	20.6	1.38x	3.75e-10
HLFM_SPICE_SPICE	34.5	38.9	20.5	25.6	1.68x	2.656e-7
SPICE_HLFM_SPICE	50.9	57.3	50.1	56	0.9x	0.005
SPICE_SPICE_HLFM	40.1	47.2	37.5	46.1	1.06x	0.013
HLM_HLFM_SPICE	48.9	54	22.8	27.6	2.14x	0.005
HLM_HLM_HLFM	40.1	47.2	17.6	30.1	2.26x	0.007

HLFS results for Config1 and Config2 with fault at diode D2 are shown in Fig. 23 and Fig. 24. Summary of experimentation for six configurations for fault at D2 is shown in Table 7.

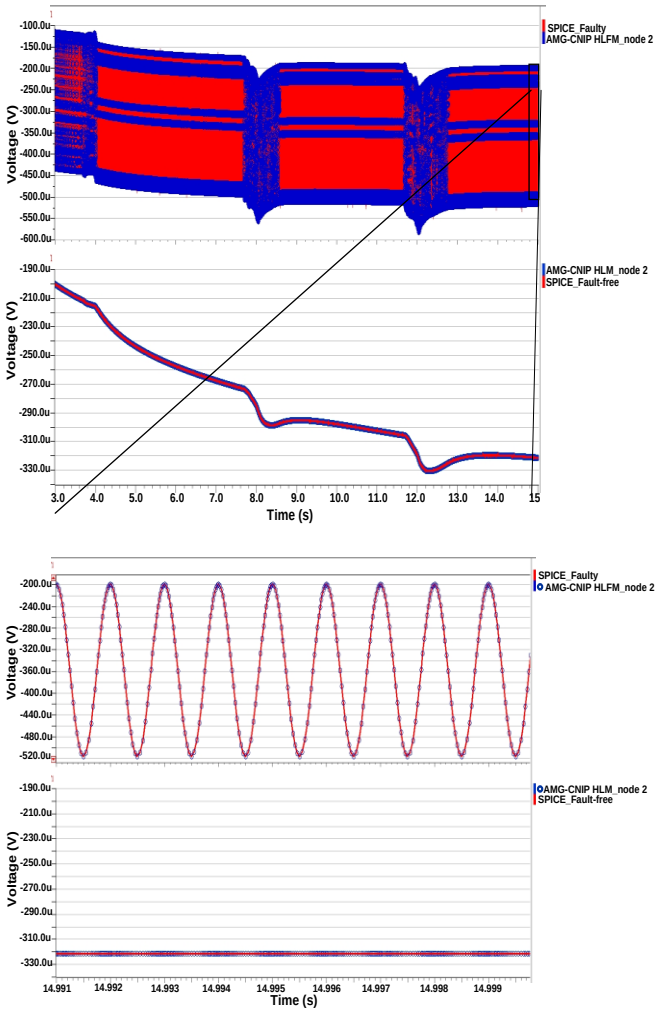


Fig. 23. Transient response of AMG-CNIP HLM and HLFM for Config1 (node 2) with fault at D2

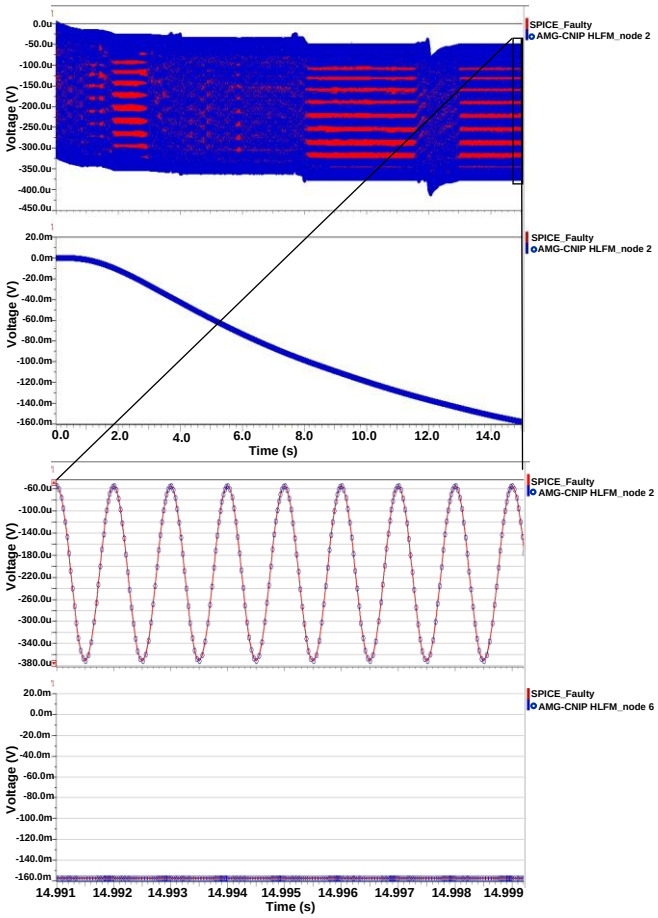


Fig. 24. Transient response of AMG-CNIP HLFM for Config2 (nodes 2 and 6) with fault at D2

VI. Conclusion

This paper presents a novel automated model generation approach using Chebyshev and Newton interpolating polynomial (AMG-CNIP). It linearizes nonlinear components

in analog circuits using *single* CN polynomial. The AMG-CNIP macromodel with interpolation and approximation properties generated is able to provide accurate responses as compared with MMGSD. The macromodel structured in VHDL-AMS shows that the AMG-CNIP can achieve less than 2% of relative percentage error during HLM and HLFM at system level as compared with SPICE simulation. More importantly a simulation speedup of 2.8x than SPICE simulation is obtained. In future, the AMG-CNIP will be extended to perform HLM and HLFM of transistor level circuits, e.g., opamps. In addition a multi-variate multi-function interpolation scheme will be considered in order to generate single CN polynomial for semiconductor devices that change operating regions during circuit operation.

References

[1] C. Wang, F, *Digital Circuit Testing: A Guide to DFT and Other Techniques*. Academic Press, 1991.

[2] S. D. Dasnurkar and J. a. Abraham, "Real-time dynamic hybrid BiST solution for Very-Low-Cost ATE production testing of A/D converters with controlled DPPM," in *2010 11th International Symposium on Quality Electronic Design (ISQED)*, 2010, pp. 562–569.

[3] S. J. Spinks, *Fault simulation for structural testing of analogue integrated circuits*. PhD Thesis, Hull Univ. (United Kingdom), 1998.

[4] S. SUNTER and M. GRAPHICS, "Essential principles for practical analog BIST," *Electronic Design News*, 2010.

[5] J. A. Abraham and M. Soma, "Mixed-Signal Test-Tutorial C," in *The European Design and Test Conference (ED&TC)*, 1995.

[6] K. Arabi, "Special session 6C: New topic mixed-signal test impact to SoC commercialization," in *VLSI Test Symposium (VTS), 2010 28th*, 2010, pp. 212–212.

[7] E. K. Bartsch, "Improved Analogue Fault Simulation Through High Level Modeling," University of Hull, 1999.

[8] P. Kalpana and K. Gunavathi, "Test-generation-based fault detection in analog VLSI circuits using neural networks," *ETRI journal*, vol. 31, no. 2, pp. 209–214, Apr. 2009.

[9] B.Antao and F. M. El-Turky, "Automatic Analog Model Generation For Behavioral Simulation," *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 12.2.1–12.2.4, 1992.

[10] P. R. Wilson, Y. Kilic, J. N. Ross, M. Zwolinski, and A. D. Brown, "Behavioural modelling of operational amplifier faults using analogue hardware description languages," in *Proceedings of the Fifth IEEE International Workshop on Behavioral Modeling and Simulation. BMAS 2001 (Cat No.01TH8601)*, 2001, pp. 106–112.

[11] Y. Kilic and M. Zwolinski, "Behavioral Fault Modeling and Simulation Using VHDL-AMS to Speed-Up Analog Fault Simulation," *Analog Integrated circuits and signal processing*, vol. 39, no. 2, pp. 177–190, 2004.

[12] Y. Joannon, V. Beroulle, C. Robach, S. Tedjini, and J.-L. Carbonero, "Choice of a High-Level Fault Model for the Optimization of Validation Test Set Reused for Manufacturing Test," *VLSI Design*, vol. 2008, pp. 1–10, 2008.

[13] G. Gronthoud and H. G. Kerkhoff, "Reducing analogue fault-simulation time by using high -level modelling in dotss for an industrial design," *IEEE European Test Workshop, 2001*, pp. 61–67, 2001.

[14] E.K.Bartsch and I. Bell, "High Level Analogue fault simulation using linear and non linear models," *Radioengineering Journal*, vol. 8, pp. 32–34, 1999.

[15] A. Meixner and W. Maly, "Fault Modeling for the Testing of Mixed Integrated Circuits," in *International Test Conference (ITS)*, p. 564-572., 26-30 Oct, 1991

[16] N. Nagi and J. Abraham, "Hierarchical fault modeling for analog and mixed-signal circuits," in *10th IEEE VLSI Test Symposium*, 1992, pp. 96–101.

[17] J. Roychowdhury, "Algorithmic macromodelling methods for mixed-signal systems," *17th International Conference on VLSI Design. Proceedings.*, pp. 141–147, 2004.

[18] P. Li and L. T. Pileggi, "NORM: Compact Model Order Reduction of Weakly Nonlinear Systems," in *Order A Journal On The Theory Of Ordered Sets And Its Applications*, 2003, pp. 1–6.

[19] M. Rewienski and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 155–170, Feb. 2003.

[20] J. Roychowdhury, "Piecewise polynomial nonlinear model reduction," in *Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451)*, 2003, pp. 484–489.

[21] S. K.Tiwary and R. A. Rutenbar, "Scalable trajectory methods for on-demand analog macromodel extraction," in *Proceedings of the 42nd annual Design and Automation*, 2005, pp. 403–408.

[22] D. De Jonghe and G. Gielen, "Characterization of Analog Circuits Using Transfer Function Trajectories," *IEEE Transcation on Circuits and Systems I*, vol. 59, no. 8, pp. 1796–1804, Aug. 2012.

[23] L. Xia, I. M. Bell, and A. J. Wilkinson, "Automated Model Generation Algorithm for High-Level Fault Modeling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 7, pp. 1140–1145, Jul. 2010.

[24] L. Xia, I. M. Bell, and A. J. Wilkinson, "Approach for Automated Model Generation," in *4th International Conference on Design and Techonology of Integrated Systems on Nanascale Era (DTIS09)*, 2009, pp. 281–286.

[25] N. Dong and J. Roychowdhury, "General-Purpose Nonlinear Model-Order Reduction Using Piecewise-Polynomial Representations," *IEEE Transactions on Computer-Aided Design*, vol. 27, no. 2, pp. 249–264, 2008.

[26] J. P. Boyd, *Chebyshev and Fourier Spectrum Methods*. Section 2.7, Dover Publication, 2001.

[27] M. U. Farooq and L. Xia, "Local Approximation Improvement of Trajectory Piecewise Linear Macromodels

through Chebyshev Interpolating Polynomials," in *IEEE/ACM 18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2013, pp. 767–772.

[28] W. Y. Yang, W. Cao, T. S. Chung, and J. Morris, *Applied Numerical Methods Using MATLAB*. Wiley-Interscience, 2005, pp. 528.

[29] D. N. Varsamis and N. P. Karampetakis, "On the Newton Multivariate Polynomial Interpolation with Applications," in *7th International Workshop on Multidimensional (nD) Systems (nDs)*, 2011, vol. 0, pp. 1–8.

[30] "SystemVision SPICE reference," Mentor Graphics Corporation, 2010.

[31] P. J. Ashenden, G. D. Peterson, and D. A. Teegarden, *The System Designer's Guide to VHDL-AMS*. Morgan Kaufmann Publisher, 2003

Dr. Likun Xia

Dr. Likun Xia started his postgraduate study at the University of Newcastle, UK in September 2001, and obtained the Masters degree of in Electronic Engineering with a project thesis entitled "A system for accurate measurement of converter output voltage in motor drives" in early 2003.

Starting September 2003, Dr. Likun Xia pursued his research as a PhD student by joining VLSI Design and Test Group at the University of Hull, UK. His PhD thesis is in the field of High Level Fault Modelling based on Automated Model Generation under the supervision of Dr. Ian M Bell and Dr. Antony J. Wilkinson.

Dr. Likun Xia is currently a senior Lecturer/Researcher at Electrical & Electronics Engineering Department, Universiti Teknologi PETRONAS (UTP), Malaysia. He is a member of IEEE CAS and EAGE.

He is interested in mathematical behavior modeling for various applications including high level fault modeling for analogue and mix-signal circuit and system; Neuroscience on stress and depression issues; and seismic imaging on fractured reservoir. His research interests are especially in the following areas:

1. Neural signal/image analysis on alcohol, sexual addiction, depression, and stress observation by using EEG, fMRI;
2. Built-in Self Test (BIST) and r self-correction mechanism development for safety-critical, mission-critical, and remote-controlled systems in PETRONAS;
3. A novel algorithm development for alerting motorcycles approaching to avoid traffic accidents in malaysia;
4. Development of high level fault modeling using automated model generation approaches for analogue and mixed-mode circuits and systems;
5. Others areas including signal/image processing are also interested.

He has published several research papers including IEEE TCAD, European Journal of Neurology, and Advanced Science Letters as a result of his research. He has also attended international conferences such as ISCAS and EMBC.