Local Approximation Improvement of Trajectory Piecewise Linear Macromodels through Chebyshev Interpolating Polynomials

Muhammad Umer Farooq, Likun Xia
Centre of Intelligent Signal and Imaging Research (CISIR)
Universiti Teknologi PETRONAS (UTP)
Bandar Seri Iskandar, Tronoh, 31750, Perak, Malaysia
mohammod_omer_farooq@yahoo.com,likun_xia@petronas.com.my

Abstract—**We introduce the concept of two dimensional (2D) scalability of trajectory piecewise linear (TPWL) through the exploitation of Chebyshev interpolating polynomials in each piecewise region. The goal of 2D scalability is to improve the local approximation properties of TPWL macromodels. Horizontal scalability is achieved through the reduction of number of linearization points along the trajectory; vertical scalability is obtained by extending the scope of macromodel to predict the response of a nonlinear system for inputs far from training trajectory. In this way more efficient macromodels are obtained in terms of simulation speed up of complex nonlinear systems. The methodology developed is to predict the nonlinear responses generated by faults introduced in Micro Electro-Mechanical Systems (MEMS) accelerometer during fabrication, that are used to obtain the seismic images for oil and gas discovery. We provide the implementation details and illustrate the 2D scalability concept with an example using nonlinear transmission line.**

Keywords-Trajectory; Chebyshev polynomial; Taylor polynomial; State space (SS); Model order reduction (MOR)

I. Introduction

To model nonlinear circuits and systems, model order reduction (MOR) techniques have been developed over the time. These approaches input netlist level description of circuits and produce system matrices that generate same response as the original system. MOR techniques are well matured for linear systems [1–3], while this is still under development for nonlinear systems.

Majority of MOR techniques [4–8] rely on the piecewise linearization (PWL) of original system response. Linearization is performed by expanding nonlinear function into its Taylor polynomials. Trajectory methods like [4],[9] use linear and quadratic Taylor polynomials in their piecewise regions. However, due to poor convergence properties of Taylor polynomials, these approaches cannot guarantee the correct response for the inputs that are not included in training or have greater magnitudes than training inputs.

It is known that Chebyshev polynomial models significantly reduce the number of linearization points as compared to Taylor polynomial models. For relatively weak nonlinearity single polynomial model can be achieved for whole trajectory. It is also shown that Chebyshev polynomial models are able to predict accurate response for the inputs far from the training input and simulate faster than Taylor polynomial macromodels.

Numerical results confirm basis of 2Dimensional (2D) scalability approach. However, there is a tradeoff between scalability and complexity, i.e., increasing the Chebyshev polynomial order reduces the number of linearization points, but increases the complexity of the system, that results in heavy computation. Previous attempts (e.g., [10]) use Chebyshev polynomials standalone to approximate nonlinear functions in electronic circuits. To the best of our knowledge approximating state-space (SS) nonlinear function with Chebyshev polynomial is the first attempt to predict the response of nonlinear electronic circuits and system.

This paper targets shortcoming above and proposes 2D scalability of trajectory methods to improve its local approximation properties. In this approach, the Taylor polynomial models is replaced with Chebyshev interpolating polynomials in each piecewise region by taking advantages of excellent convergence properties of Chebyshev polynomials. The SS is employed and split into multiple linearization regions on the basis of convergence of Chebyshev polynomials.

The remainder of the paper is organized as follows. In section II a brief background of existing MOR techniques for nonlinear systems is given. 2D scalability approach is discussed in section III. The process of obtaining Chebyshev macromodels is discussed in section IV, and evaluation results of Chebyshev macromodels are given in section V. The conclusion is provided in section VI.

II. Background

A. Macromodeling of Nonlinear Systems

A nonlinear dynamical system can be described by a SS approach [4] shown in equation (1).

$$\begin{cases} \dfrac{dg\,(x(t))}{dt} = f\,(x(t)) + B(x(t))u(t) \\ y(t) = C^{T}x(t) \end{cases} \tag{1}$$

where $x(t) \in R^{N}$ is state vector at time t representing the unknown node voltages and branch currents;

978-1-4673-3030-5/13/$31.00 ©2013 IEEE

$f: R^N \to R^N$ and $g: R^N \to R^N$ are nonlinear vector valued functions representing charge/flux and resistive terms in the circuit respectively; B is a SS-dependent $N \times M$ 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 $N \times K$ output matrix and $y: R \to R^K$ is the output signal.

Several MOR techniques have been developed to model strong and weak nonlinear circuits and systems using the SS approach in equation (1) [4–8]. Another class of macromodeling approach use black-box modeling approaches, e.g., [11–13]. However, these approaches do not take into account system insight information and are less applicable for reduction methods. Our development of 2D scalability framework is based on white box state space macromodeling techniques.

Two of the most popular techniques for modeling strong nonlinear systems using SS are Trajectory Piecewise Linear (TPWL) [4] and Piecewise Polynomial (PWP) [9]. A brief description of these approaches is provided in the following subsection.

B. TPWL and PWP

TPWL was originally developed to model the behavior of strong nonlinear systems. In essence, it splits full SS into adjoining polytopes and linearizes the nonlinear response around an expansion point in each polytope. Macromodels are then generated along an application specific training trajectory. The method requires a pure linear model (Taylor polynomials of degree one) shown in equation (2) to linearize the nonlinear functions *f*(*x*) and *g*(*x*).

$$\frac{d}{dt}\left(g(x_o)+G_o(x-x_o)\right)=f(x_o)+A_o(x-x_o)+Bu \qquad (2)$$

Where G_o and A_o are the Jacobian matrices evaluated in each region around expansion point x_o; B is a constant matrix.

The final model is the weighted combination of *s* linearized models using the equation (3), where $\widetilde{w}_i(x)$ are the state-dependent weights.

.

$$\frac{d}{dt}\left[\sum_{i=0}^{s-1}\widetilde{w}_i(x)(g(x_i)+G_i(x-x_i))\right] = \sum_{i=0}^{s-1}\widetilde{w}_i(x)(f(x_i)+A_i(x-x_i)+Bu) \qquad (3)$$

TPWL has excellent global approximation properties due to piecewise nature; but it has limited capability to predict the response of the systems for inputs far from training trajectory. The limitation arises from the utilization of first order Taylor polynomials in each piecewise region.

To improve the local approximation properties of TPWL Dong et al. [9] proposed the PWP approach. PWP adopts the same methodology of TPWL of generating multiple adjoining polytopes using appropriately selected linearization points. The difference is that PWP uses higher order Taylor polynomials in each piecewise region. Therefore PWP approach increases the scope of the local model to predict the response of a system with inputs having higher magnitudes than from training input. The resulting model is shown in equation (4).

$$E\frac{d}{dt}x = f(x_i)+A_i^{(1)}x_i^{(1)}+A_i^{(2)}x_i^{(2)}+Bu(t),\, y = Cx \qquad (4)$$

where $x^{(1)} = x - x_i$, $x^{(2)} = (x - x_i)\otimes(x - x_i)$; $A_i^{(1)}$ and $A_i^{(2)}$ are the first and second order derivatives, respectively. The final model is a weighted combination of *s* linearized models.

In the early work, PWP only utilized up to quadratic term of Taylor polynomial in each local model. Intuitively for inputs far from training trajectory, higher order polynomials (greater than degree 2) could be employed in local models to achieve better accuracy. It is known that Taylor polynomials have bad convergence properties due to non-uniform approximation error distribution [14]. Approximation error is less as long as input stays close to expansion point, but increases rapidly as input moves away from expansion point. This means that for trajectories far from training trajectories, even higher order Taylor polynomials cannot predict the accurate response.

From discussion above it is clear that further improvement is required for TPWL models in terms of the *scalability*. Tiwary et al. [6] greatly improve the scalability of TPWL model by utilizing a scalable interpolating architecture. However, they only use pure linear model (first order Taylor polynomial) in each piecewise region, which has the inherited limitations of capturing only weak nonlinearities.

C. Chebyshev Polynomials

Major problem in classical interpolating polynomials (such as Newton, Largrange etc.) is that the interpolating nodes are not properly spaced due to which approximation of a function is not accurate [15]. Selection of target nodes for interpolating polynomials is critical for accuracy and Chebyshev nodes are considered as the suitable candidate due to their discrete orthogonality properties [15].

Chebyshev polynomials of first kind are given in equation (5) [16], where x' are the nodes of these polynomials; N is the order of polynomial.

$$T_{N+1}(x') = \cos((N+1)\cos^{-1}x) \qquad (5)$$

The nodes of Chebyshev polynomials in equation (5) are calculated using equation (6) as long as input lies in the interval [-1,1].

$$x'_k = \cos\frac{2N+1-2k}{2(N+1)}\pi \qquad k{=}0,1...,N \qquad (6)$$

For an arbitrary interval [a,b], input *x* needs to be normalized and nodes are calculated according to equation (7).

$$x'_k = \frac{b-a}{2}\cos\frac{2N+1-2k}{2(N+1)}\pi+\frac{a+b}{2} \qquad k{=}0,1...,N \qquad (7)$$

To approximate a function *f(x)*, discrete orthogonality of Chebyshev polynomials can be exploited to define the truncated Chebyshev series shown in equation (8) [16].

$$f(x) \cong c_N(x) = \sum_{m=0}^{N} d_m T_m(x')\Big|_{x'=\frac{2}{b-a}(x-\frac{a+b}{2})} \qquad (8)$$

where

$$d_m = \frac{2}{N+1}\sum_{m=0}^{N} f(x_k)\cos\frac{m(2N+1-2k)}{2(N+1)}\pi \quad \textit{for m=1,2...,N}$$

To generate interpolating polynomials that can be practically simulated with SS of form (1), a combination of Netwon polynomials with Chebyshev knots [15] can be used. This implies that the knots of interpolating polynomials are generated using Chebyshev knot formula given in equation (5) or (6) and then generate the polynomial using Newton divided difference method. Resulting polynomial for approximating the function *f(x)* can be found by equation (9), where $c_1, c_2, c_3..c_n$ are the polynomial coefficients.

$$f(x) \approx c_1 x^{n-1} + c_2 x^{n-2} + c_3 x^{n-3} + \ldots + c_n^1 \qquad (9)$$

III. Chebyshev Polynomials For 2D Scalability Of Local Models

A. 2D Scalability of the Trajectory Macromodels

It has been shown in section II that TPWL utilizes pure linear model in each linearized region, whereas PWP uses higher order Taylor polynomials to improve local approximation in each piecewise region. Due to limited region of convergence of Taylor polynomials, these polynomials are unable to interpolate the non-reachable states that are not excited during model training. Primarily, because of guaranteed convergence it allows to use higher order polynomials in each region. Secondly, it reduces the overall number of linearization points used for model generation. Chebyshev polynomials guarantee convergence inside an interval [a,b], as long as no singularities lies within the interval [16]. This means that higher order Chebyshev polynomials can be employed confidently in each region to interpolate non-reachable states during simulation. Advantages of using Chebyshev polynomials in SS are twofold.

Consider a function $f(x) = \dfrac{x^2 \exp^{-\frac{x^2}{2}}}{\sqrt{2\pi}} + x\sin(2\pi x)$, which is evaluated at 100 points of x = -2.5:0.05:2.5. All points are used for curve fitting. Taylor series expansions are performed around the origin. One can observe from fig.1 that as long as input x is small within the range of (-1,1), Taylor polynomial fit is accurate (at the expense of high polynomial orders).

Outside the region, Taylor polynomials start diverging from solution and even increasing order does not show significant difference.

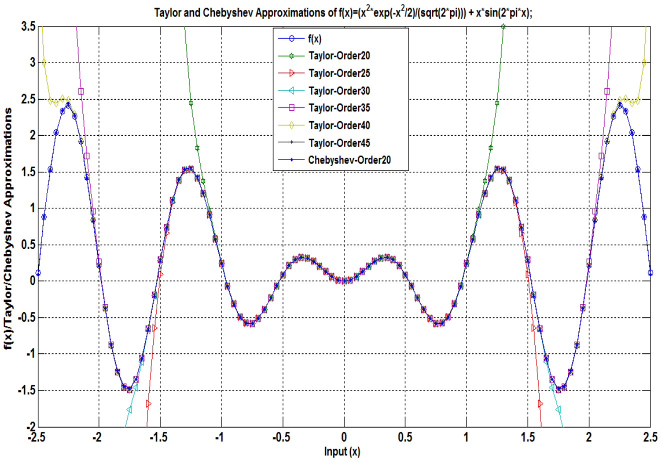

Fig. 1. Higher degree Chebyshev and Taylor polynomial approximations

On the other hand, single Chebyshev polynomial (of relatively high order of 20) is able to fit the complete curve. This implies that for large input swing, Chebyshev models can cover the large SS regions as compared to Taylor polynomials, where multiple linearization points with many polynomials are used to cover large inputs. This also provides the intuition that total number of linearization points can be *reduced* using Chebyshev polynomials, thus enabling fast simulation of macromodels.

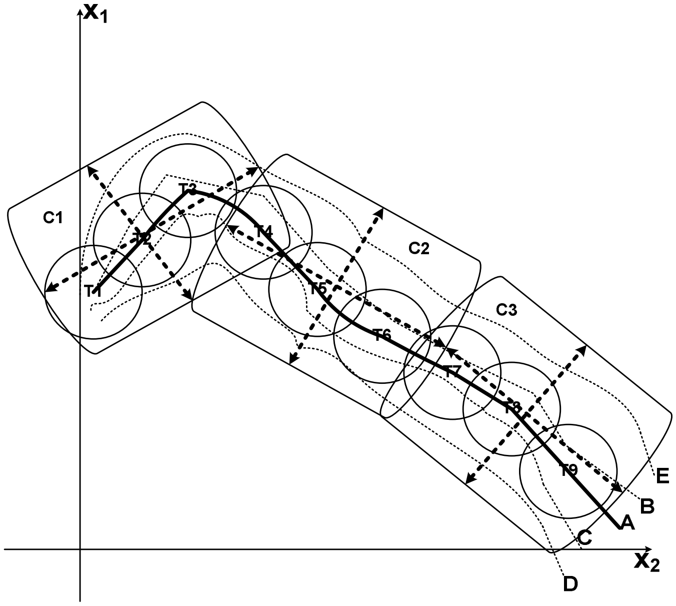

Fig. 2. Generation of linearized models using Taylor and Chebyshev polynomials along a trajectory of nonlinear system in a 2D SS

Consider an example illustrated in fig. 2, 9 linearization points are generated along a training trajectory *A* using Taylor polynomials. It can be seen that as long as input is close to linearization points, trajectories (*B* and *C*) fall inside the regions of linearization points. Once input is far from training input, other trajectories (D and E) are distant from *A*. However, large rectangles representing Chebyshev linearization regions are able to cover all trajectories far from training trajectory *A*. It implies that Chebyshev polynomials can improve the scalability of TPWL models and increase the scope of a macromodel in 2D: horizontally the number of linearization points is reduced; vertically the span is increased to cover more trajectories far from training trajectory (2D extensions can be observed from dashed lines in fig. 2).

B. *Representation of nonlinear function with Chebyshev polynomials*

Without lost of generality, state equation (1) can be expressed as equation (10)

$$E\frac{d}{dt}x = f(x) + Bu(t) \tag{10}$$

where function *g(x)* is considered to be linear and is replaced with a constant *E*. Similarly, *B* is considered to be time invariant.

The only nonlinear function in equation (10) is *f(x)* that can be approximated using Chebyshev interpolating polynomial. Replacing *f(x)* in equation (10) with n^{th} order Chebyshev interpolating polynomial, resulting state equation has the form shown in equation (11).

$$E\frac{d}{dt}x = \left\{c_1x^{n-1} + c_2x^{n-2} + c_3x^{n-3} + \ldots + c_n^1\right\} + Bu(t) \tag{11}$$

where $c_1, c_2, \ldots, c_n$ are the polynomial coefficients computed using Newton divided difference method [15].

IV. Macromodel Generation Using Chebyshev Polynomials

In this section we provide steps to obtain macromodels using Chebyshev interpolating polynomials.

A. *Data Extraction from SPICE*

To generate macromodels of original system, SPICE data (node voltages and branch currents) is utilized in the nonlinear SS form of equation (1). Most SPICE engines, e.g., SPICE2 and SPICE3 do not solve the circuit equations in the SS form. Instead, they use Modified Nodal Analysis (MNA) method [17]. It means SPICE output in SS form cannot be directly utilized.

Typically SPICE engines use Newton-Raphson (NR) method [17] to solve nonlinear circuits. NR method works by linearizing the nonlinear elements to form equivalent companion models. Once companion models are obtained, their values are stamped to build the linearized circuit matrices. These matrices are then solved typically using LU factorization method.

During implementation, transient simulations is run using system level software called SystemVision (SV) [18], which requires SPICE3 for analog circuit simulations. To obtain training trajectories from SV, we capture data after each successful convergence of NR step and before stamping it into MNA matrix. In a sense, raw data is required to manipulate and thus can be directly used in the SS formulation. Data transfer between SV and MATLAB/Simulink is carried out through special virtual bus interface [18]. After the completion of transient simulation, data is transferred from Simulink model into the MATLAB environment.

B. *Macromodel Training Input*

In order to generate accurate macromodels, the macromodel needs to be trained with the input that can excite all possible states of SS. A good training input should be strong enough that it forces the SS to its upper bounds [9].

Typically for training, trajectories are generated by combining signals of different amplitudes, frequencies and phases. In this study, we use single training data by superimposing a 500mV, 10*us* PRBS signal on 120Hz, ±2 V sinusoidal shown in fig. 3. High frequency PRBS signal is expected to excite all possible states of SS and high input values force SS to its maximum limits.

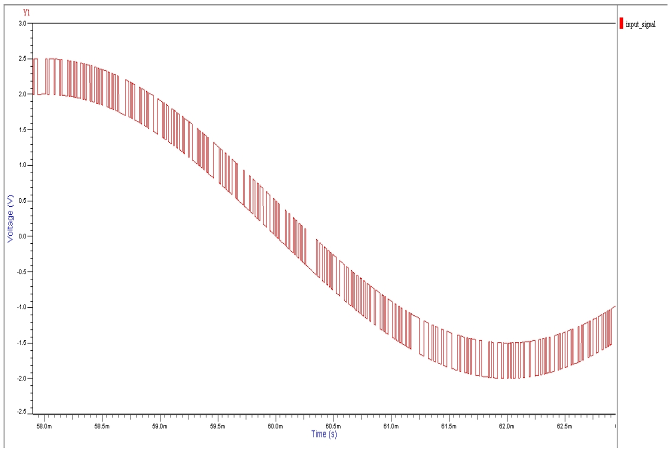

Fig. 3. Highlight of a strong training input

C. *Macromodel Generation*

The model generation steps are summarized as follows.

1. Simulate the full system with training input proposed in section IV.B and transfer data to MATLAB environment as explain in section IV.A.
2. Initialize order *n* for Chebyshev polynomials.
3. Generate Chebyshev interpolating knots using equations (6) or (7) according to input range.
4. Employ Newton divided difference method to obtain approximating polynomial in terms of state variable *x*, such that $f_c(x)$ becomes:

$$f_c(x) \approx c_1x^{n-1} + c_2x^{n-2} + c_3x^{n-3} + \ldots + c_n^1 \tag{12}$$

5. Traverse the full training trajectory while ensuring that relative error $relerror = \left| \frac{f(x) - f_c(x)}{f_c(x)} \right| < \delta$. δ is the relative error tolerance.
6. If *relerror* > δ , the order of polynomial increases by some predefined increment steps γ. Value of γ is usually set in the range of (1,5). The process repeats from step 2 to 6 until transient simulation data ends.

V. Evaluation of Chebyshev Polynomial Macromodels

A simplified nonlinear transmission line shown in fig. 4 [9] is employed to evaluate the algorithm and then compare it with Taylor polynomial based PWP implementations. The circuit consists of resistors, capacitors and diodes with constitutive equation $i_d = \exp(40v) - 1$. All resistors and capacitors have unit value, i.e., *R*=*C*=1. The single input is the current source entering node 1, *u*(*t*) = *i*(*t*); the single output is the voltage at node 1, *y*(*t*) = v_1(*t*).

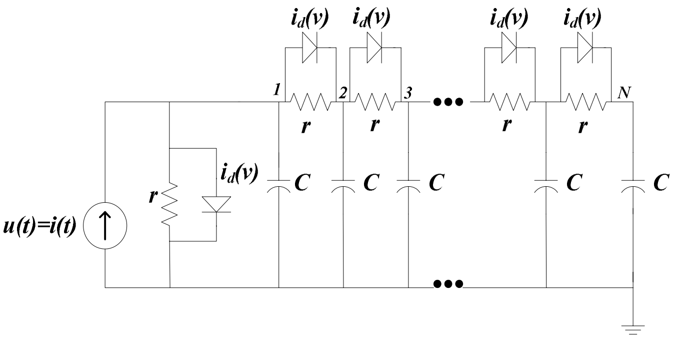

Fig. 4. Nonlinear Transmission Line [9]

For training of macromodel, input signal shown in fig. 3 is used. To evaluate the macromodel generated using Taylor and Chebyshev polynomials, a transient simulation of 200ms is run with input signal of *u*(*t*)=*i*(*t*)=2*sin*(*2π10t*), input current amplitude of ±2A and 10Hz frequency. In this implementation, we generate single Taylor and Chebyshev polynomial models representing nonlinear function *f(x)* for the whole trajectory.

Fig. 5. show the simulation results of Chebyshev and Taylor polynomial macromodels. Order of both polynomials is set to *10.*It is clearly seen that output generated by Chebyshev macromodel matches exactly with original systems output. Whereas, there is a lose in accuracy for Taylor polynomial macromodel.

To evaluate these macromodels for inputs far from training input, we run a transient simulation in which input is larger than the training input, i.e., *u*(*t*)=*2.6sin*(*2pi10t*) with input current amplitude of ±*2.6A*. Order of both polynomials is 10. One can clearly observe from the fig. 6 that accuracy of output voltage for Taylor macromodel is poor. It is mentioned in section III.A that as input starts moving far from training trajectory the output of the Taylor polynomial macromodel starts diverging from the original solution. In this case, input is only *0.1A* far from training trajectory and there is a significant lost in accuracy for Taylor macromodel. On the other hand, the Chebyshev macromodel output matches with the original.

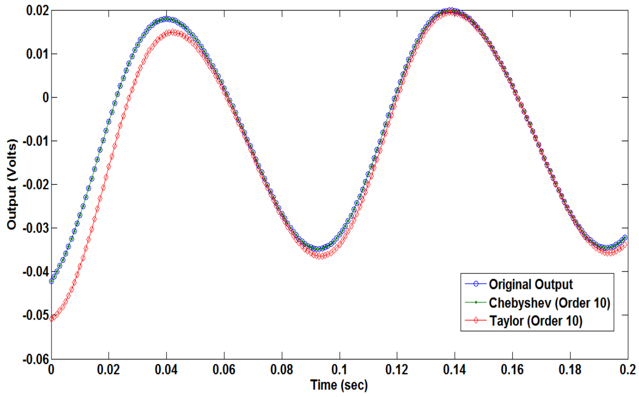

Fig. 5. Chabyshev and Taylor polynomial macromodels outputs for input of *i*(*t*)=2*sin*(*2π10t*)

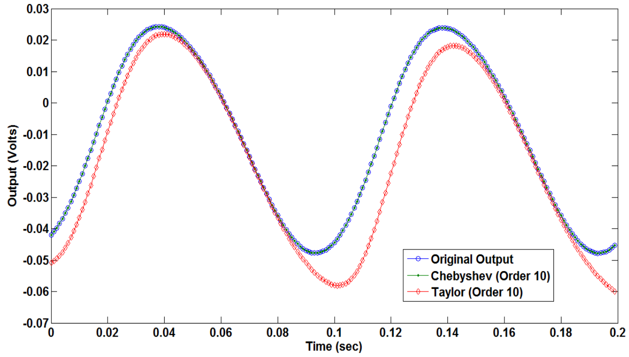

Fig. 6. Chebyshev and Taylor polynomial macromodels outputs for input of *i*(*t*)=2.6*sin*(*2π10t*)

We also test capability of Chebyshev polynomial macromodel for the case when the system input is far from training input. A transient simulation is run with a input signal *u*(*t*)=*5.4sin*(2π10t), i.e., input current swings between ±*5.4A,* which means that input is 2.9A away from training input.

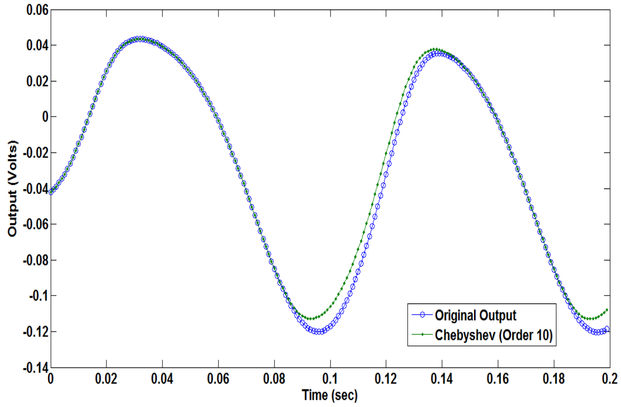

Fig. 7. Chebyshev polynomial macromodel output for input of *i(t)*=5.4*sin(2π10t)*

It is clear from fig. 7 that even for the input far from training input, Chebyshev macromodel still can provide more accurate results. This proves that Chebyshev macromodels are efficiently able to interpolate the states that are not excited by training input.

Moreover, Chebyshev macromodels performs about ten times faster than the Taylor macromodels.

VI. Conclusion

This paper presents a 2D scalability approach to improve the local approximation properties of TPWL macromodels through the utilization of Chebyshev interpolating polynomials. To scale the SS horizontally, we use single polynomial of order 10 for both Chebyshev and Taylor macromodels to cover the whole trajectory. Both achieve good accuracy under the same conditions but Chebyshev macromodels run faster than Taylor macromodels. We also compare their capabilities to predict responses for the inputs far from training trajectory. Compared with Chebyshev, Taylor macromodels lose their accuracy significantly even only ±0.1A far from training input current. On the other hand Chebyshev macromodel accuracy is good for input ±0.1A far from training. The upper bound of Chebyshev macromodel is also tested with the input current of ±*5.4A* that is ±2.9A far from training input swing ±2.5A. Chebyshev macromodel shows reasonable accuracy for this input as well. Therefore, Chebyshev polynomials are able to scale trajectory models in 2D and achieve simulation speedup as compared to conventional Taylor polynomial macromodels.

In future, this macromodeling technique will be applied to more complex systems and to perform High Level Modeling (HLM) to achieve speedup at system level. Also the macromodeling approach will be used for fault modeling in MEMS accelerometers.

Acknowledgment

This work is supported by the Fundamental Research Grant Scheme (Ref: FRGS 2/2010/TK/UTP/03/8), Ministry of High Education (MOHE), Malaysia and YUTP (Ref: YUTP-FRG/1/2012/EOR/039) by PETRONAS Sdn Bhd.

References

[1] P. Feldmann and R. W. Freund, “Efficient linear circuit analysis by Pade approximation via the Lanczos process,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639-649, May 1995.

[2] A. Odabasioglu, M. Celik, and L. T. Pileggi, “PRIMA: passive reduced-order interconnect macromodeling algorithm,” *Proceedings of IEEE International Conference on Computer Aided Design (ICCAD) ICCAD-97*, pp. 58-65, 1997.

[3] L. T. Pillage and R. A. Rohrer, “Asymptotic Waveform Evaluation for timing analysis,” *IEEE Trans. on Computer -Aided Design*, vol. 9, no. 4, pp. 352-366, 1990.

[4] 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.

[5] N. Dong and J. Roychowdhury, “General-Purpose Nonlinear Model-Order Reduction Using Piecewise-Polynomial Representations,” *Computer-Aided Design*, vol. 27, no. 2, pp. 249-264, 2008.

[6] S. Tiwary, “Scalable trajectory methods for on-demand analog macromodel extraction,” in *Proceedings of the 42nd annual Design and Automation*, 2005, pp. 403-408.

[7] B. Bond and L. Daniel, “Parameterized model order reduction of nonlinear dynamical systems,” *ICCAD-2005. IEEE/ACM International Conference on Computer-Aided Design, 2005.*, no. 4, pp. 487-494, 2005.

[8] D. Vasilyev, M. Rewienski, and J. White, “A TBR-based trajectory piecewise-linear algorithm for generating accurate low-order models for nonlinear analog circuits and MEMS,” in *Proceedings of the 40th conference on Design automation - DAC ’03*, 2003, p. 490.

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

[10] I. Sarkas, D. Mavridis, M. Papamichail, and G. Papadopoulos, “Large and small signal distortion analysis using modified Volterra series,” *2007 IEEE International Symposium on Circuits and Systems*, vol. 54, no. 2, pp. 133-142, Nov. 2007.

[11] 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.

[12] L. Xia, I. M. Bell, and A. J. Wilkinson, “A Robust approach for automated model generation,” in *4th IEEE International Conference on Design and Technology of Integrated systems of Nanoscale era*, 2009, pp. 281-286.

[13] L. Ljung, *System Identification Theory for User*. Upper Saddle River: Prentice Hall PTR, 1999.

[14] J. P. Boyd, *Chebyshev and Fourier Spectrum Methods*. Dover, 2001.

[15] W. Y. Yang, W. Cao, T.-S. Chung, and J. Morris, *Applied Numerical Methods Using MATLAB®*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005.

[16] D. C. Handscomb, *Chebyshev Polynomials*, 1st ed. Chapman & Hall/CRC, 2003.

[17] L. Pillage and R. Rohrer, *Electronic Circuit and System Simulation Methods*, 1st ed. McGraw-Hill, 1995.

[18] “SystemVision - Virtual Lab for Mechatronic System Design and Analysis - Mentor Graphics.” .