POLITECNICO DI TORINO
Repository ISTITUZIONALE

Compressed SPICE-Compliant IC Models via Machine Learning Kernel Regression

Original
Compressed SPICE-Compliant IC Models via Machine Learning Kernel Regression / Atlante, M., Trinchero, R., Bradde, T., Manfredi, P., Stievano, I.S.. - ELETTRONICO. - (2024), pp. 1-3. (IEEE Electrical Design of Advanced Packaging and Systems (EDAPS) Bangalore (Ind) 17-19 December 2024) [10.1109/edaps64431.2024.10988464].

Availability:
This version is available at: 11583/3000088 since: 2025-05-13T07:35:41Z

Publisher:
IEEE

Published
DOI:10.1109/edaps64431.2024.10988464

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

30 October 2026

Compressed SPICE-compliant IC Models via Machine Learning Kernel Regression

Marco Atlante[1]
Dept. Electronics and Telecomm.
Politecnico di Torino
Torino, Italy
marco.atlante@polito.it

Riccardo Trinchero[2]
Dept. Electronics and Telecomm.
Politecnico di Torino
Torino, Italy
riccardo.trinchero@polito.it

Tommaso Bradde[3]
Dept. Electronics and Telecomm.
Politecnico di Torino
Torino, Italy
tommaso.bradde@polito.it

Paolo Manfredi[4]
Dept. Electronics and Telecomm.
Politecnico di Torino
Torino, Italy
paolo.manfredi@polito.it

Igor S. Stievano[5]
Dept. Electronics and Telecomm.
Politecnico di Torino
Torino, Italy
igor.stievano@polito.it

***Abstract*—This paper introduces a fully behavioral machine learning methodology for generating compact and accurate models of IC buffers. The proposed approach leverages a vector-valued implementation of the kernel Ridge regression to construct models based on observations of device transient responses recorded during normal operation. A key focus is placed on developing an efficient compression scheme to minimize model complexity (i.e., the number of regression coefficients), resulting in a compact mathematical representation that can be efficiently integrated into any SPICE-based solver.**

***Index Terms*—digital integrated circuits, buffer modeling, signal integrity, high-speed interconnects, kernel regression.**

I. Introduction

For decades, the signal and power integrity community has focused on the development of accurate and efficient simulation models of digital IC buffers to be used for the quality and reliability assessment of high-speed digital channels.

This trend has been largely driven by the development of the Input/Output Buffer Information Specification (IBIS), supported by electronic design automation (EDA) tools and silicon vendors [1]. IBIS promotes a modeling approach based on fundamental building blocks representing the functionalities of key elements within IC buffers. Despite the widespread adoption of IBIS, the research community has continued to explore alternative solutions aimed at enhancing accuracy and/or simplifying the model generation process [2]–[4].

With the rapid advancement of machine learning (ML) techniques, behavioral modeling approaches for IC buffers based on recurrent neural networks (RNNs) have gained significant attention due to their structural flexibility and accuracy [5], [6]. On the other hand, the training of such kind of model can be computationally expensive, since it requires the solution of a non-convex optimization problem and the tuning of several hyperparameters (e.g., number of layer, number of neuron, etc...). Moreover, due to their inherently nonlinear formulation and complexity of RNN structures, their implementation in SPICE-like circuital solver is rather complex and usually it is done in advance hardware description languages, such as Verilog-A [6].

In the above modeling framework, kernel regressions can be seen as a promising alternative for the development of accurate and compact behavioral models of IC buffers [7]. The underlying idea is to use traditional feedforward kernel regressions together with a nonlinear autoregressive exogenous (NARX) model of the system. This approach results in a linear model with respect to its parameters with a small set of hyperparameters to tune, thus potentially simplifying its direct implementation into SPICE-like solvers.

In this work the vector-valued kernel Ridge regression (VV-KRR) [8] is adopted to capture the dynamical nonlinear behavior of a single-ended IC buffer incorporating the effects of power supply fluctuations and the modeling of both the output and power supply currents. The model is built directly from a set of observations of the IC port transient responses during normal device operation. Moreover, a preliminary compression scheme is proposed to reduce the overall complexity of the obtain model, with the aim of providing a compact, accurate and efficient implementation in any SPICE compliant solver.

II. IC modeling via Kernel Ridge Regression

Let us consider the problem of building a dynamical model for the electric port variables of the IC driver shown in Figure 1 in terms of the following NARX model [7], [9], [10]:

$$\hat{\boldsymbol{y}}_k = f(\boldsymbol{y}_{k-1}, \ldots, \boldsymbol{y}_{k-p}, \boldsymbol{u}_k, \ldots, \boldsymbol{u}_{k-p}), \qquad (1)$$

where f is a generic nonlinear mapping, $\boldsymbol{y}_k = [y_k^{(1)}, y_k^{(2)}] = [i_{2,k}, i_{3,k}]$ and $\boldsymbol{u}_k = [v_{1,k}, v_{2,k}, v_{3,k}]$ represent the true output vector and corresponding input vector collecting the voltages and currents at the device ports at discrete time index k (e.g., $i_{2,k} = i_2(kT_s)$, being T_s the assumed sampling period), and $\hat{\boldsymbol{y}}_k$ is the estimated output at the discrete time k. The value p denotes the model order.

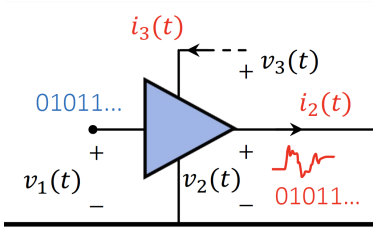


Fig. 1. Illustration of a typical IC driver with its relevant input and output electrical variables.

The VV-KRR in [8] can be suitably adopted to build the vector-valued NARX model in (1) using as training data $\mathcal{D} = \{(\tilde{\boldsymbol{x}}_l, \boldsymbol{y}_l)\}_{l=1+p}^{L}$, where $\tilde{\boldsymbol{x}}_l = [\boldsymbol{x}_l, \boldsymbol{y}_{l-1}, \ldots, \boldsymbol{y}_{l-p}]^T$ in which the vector $\boldsymbol{x}_l = [\boldsymbol{u}_l, \ldots, \boldsymbol{u}_{l-p}]$ collects the current and past discrete samples of the input variables [9], [10]. Applying the block-diagonal formulation of the VV-KRR (see [8] for additional details), the model for the generic i-th output at the k-th time instant writes:

$$\hat{y}_k^{(i)} = \sum_{l=1+p}^{L} \alpha_l^{(i)} K_{\boldsymbol{x}}(\tilde{\boldsymbol{x}}_l, [\boldsymbol{x}_k, \hat{\boldsymbol{y}}_{k-1}, \ldots, \hat{\boldsymbol{y}}_{k-p}]^T), \quad (2)$$

where $\hat{y}_k^{(i)}$, with $i = 1$ and 2, is the actual prediction of the model for the i-th output, $\{\alpha_l^{(i)}\}_{l=1+p}^{L}$ are the corresponding model coefficients to be estimated during the training and $K_{\boldsymbol{x}}(\cdot,\cdot)$ is a conventional scalar kernel function acting on the input space.

For each output, the coefficients $\boldsymbol{\alpha}^{(i)} = [\alpha_{1+p}^{(i)}, \ldots, \alpha_L^{(i)}]^T$ are estimated using the training set $\mathcal{D}$ by solving the following linear system [8]:

$$\boldsymbol{\alpha}^{(i)} = (\mathbf{K}_{\boldsymbol{x}} + \lambda \mathbf{I})^{-1} \boldsymbol{y}^{(i)}, \quad (3)$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{K}_{\boldsymbol{x}}$ is the Gramian matrix of the kernel acting on the input space, with elements $[K_{\boldsymbol{x}}]_{kj} = K_{\boldsymbol{x}}(\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}_j)$. The vector $\boldsymbol{y}^{(i)} = [y_{(1+p)}^{(i)}, \ldots, y_L^{(i)}]^T$ collects the training samples for the i-th output, and λ serves as a Tikhonov regularization parameter. In this work, a Gaussian Radial Basis Function (RBF) kernel is used. All the model hyperparameters are tuned using the validation set.

It is important to note that, thanks to the block-diagonal kernel structure used in the considered implementation of the VV-KRR, the number of hyperparameters to be tuned during the model training is independent of the number of the output variables. This is because the Gramian kernel matrix $\mathbf{K}_{\boldsymbol{x}}$, the regularization parameter λ and the kernel function $K_{\boldsymbol{x}}$ in (3) are shared across all output dimensions. This characteristic simplifies the SPICE implementation in the next section.

III. Model Compression & SPICE Implementation

The modeling of dynamical systems from transient data has the inherent weakness of dealing with a possibly long sequence of samples arising from the system responses. This leads to a large number of training samples which can negatively impact the efficiency of a possible implementation of the model in a SPICE-like solver. In fact, the number of coefficients and of evaluations of the kernel function $K_{\boldsymbol{x}}$ in (3) is equal to the number of training samples (i.e., $N_{train} = (L - 1 - p)$).

Taking inspiration from advanced compression methods for kernel regressions [8], [11], this work presents a simple model compression scheme. Specifically, the model is trained by using a subset of $N_{train} = (\bar{L} - 1 - p)$ samples $\bar{\mathcal{D}} = \{(\bar{\boldsymbol{x}}_l, \bar{\boldsymbol{y}}_l)\}_{l=1+p}^{\bar{L}}$, with $\bar{\mathcal{D}} \subseteq \mathcal{D}$, randomly chosen without replacement from the original training set $\mathcal{D}$, leading to a compressed and compact model with beneficial effects on the training and prediction time, without compromising the model accuracy. This can be justified by the usual high correlation among samples of the transient responses.

The equations of the compressed model are then implemented in LTSpice using behavioral sources for the currents i_2 and i_3 (i.e., the model outputs) as a function of the port voltages observed at the present and past time samples, taking advantage of the fact that the same kernel function $K_{\boldsymbol{x}}$ in (3) is used by all the outputs. It is important to notice that, with SPICE commands, the observation of a variable at a given past instant is conveniently carried out via a matched transmission line with a fixed delay (e.g., $(k-1)T_s$) driven by a controlled voltage source fed by the variable of interest (e.g., v_1).

IV. Results

The results collected in this section are based on the commercial Texas Instruments transceiver SN74ALVCH16973 with nominal power supply voltage VDD = 1.8 V. The HSPICE transistor-level description provided by the vendor is used as the reference to compute the model responses.

Four alternate set of responses are used for the model training: #1) an open-ended transmission line with 75 Ω characteristic impedance and 1.5 ns delay; #2) a lumped 50 Ω resistor in parallel to a 10 pF capacitor; #3) a series connection of a 50 Ω resistor and the nominal power supply; #4) a lumped 150 Ω resistor. The power supply voltage (i.e., $v_3(t)$) is forced by a piece-wise linear voltage source connected to the power supply pin, defining a stepwise behavior of the voltage during normal buffer operation. The waveforms obtained from these configurations were sampled with a sampling period $T_s = 500$ ps, leading to a full training dataset with $N_{train} = 958$ samples for a dynamic order $p = 3$. The above dataset is used to train a full VV-KRR model.

Moreover, a set of compressed models with a smaller number of training samples corresponding to $N_{train} = 600$ and $N_{train} = 300$ are generated, according to the procedure in Section III. The above models are then implemented in LTSpice. The accuracy of the obtained models is then tested on a new simulation setup involving an open-ended transmission line with 50 Ω characteristic impedance and 2 ns delay.

Figure 2 compares the reference output and power supply current waveforms with the ones predicted by the LTSpice implementation of the considered VV-KRR models. The first and third panels of the figure show, with black lines, the reference waveforms of $i_2(t)$ and $i_3(t)$ obtained from a HSPICE simulation of the test configuration. The shaded

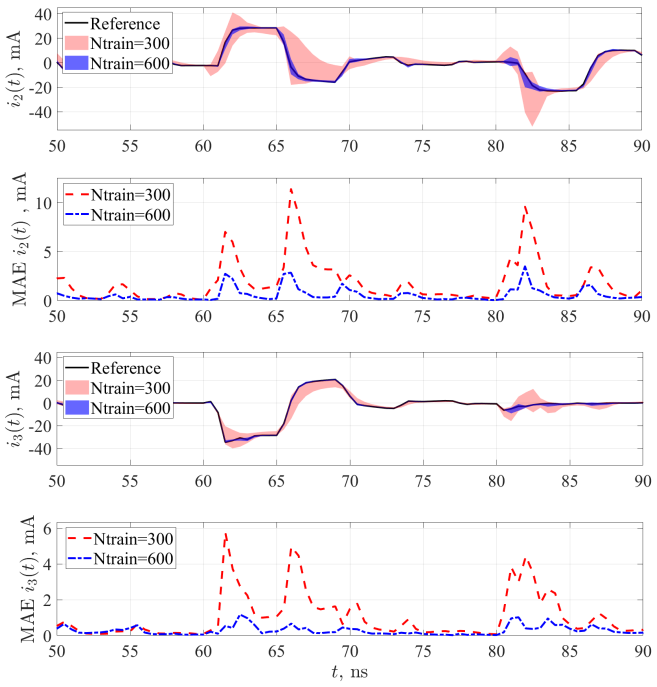


Fig. 2. Model validation: comparison between the reference output $i_2(t)$ and power supply $i_3(t)$ currents in the test set and the model prediction obtained by the proposed compressed models, along with the corresponding MAE.

areas indicate the spread of the predictions obtained from 10 instances of the compressed models trained with different random subsets of $N_{\text{train}} = 600$ (blue) or $N_{\text{train}} = 300$ (red) data samples. The second and fourth panels shows the mean absolute error (MAE) between the model predictions and the reference signals. As expected, the results show that higher compression leads to greater variability in the model responses, highlighting the impact of sample selection on the model variance. The MAE plots illustrate the robustness of compressed models, despite the slight accuracy loss occurring with higher compression.

Additionally, Fig. 3 compares the prediction of the best compressed models among the generated ones (dashed lines) to the reference responses (solid lines). This alternate validation proves the benefits of the proposed solution and highlights a general good matching of the reference responses even for the strongest compression with $N_{train} = 300$.

Summarizing, data compression demonstrates that a smaller number of bases still guarantees a good accuracy, with favourable effects on SPICE simulation time which changes from 56 s without compression with $N_{train} = 958$, to 36 s and 18 s for compressed models with $N_{train} = 600$ and $N_{train} = 300$, respectively. This linear scaling of the simulation time with respect to the actual number of expansion terms in (3) highlights the potential of the proposed SPICE-compliant macromodeling approach for applications in signal integrity and power integrity design flows, demonstrating its efficiency, accuracy, and robustness.

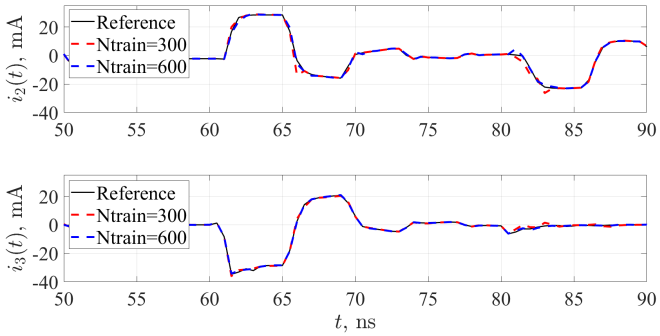


Fig. 3. Model validation: comparison between the reference output $i_2(t)$ and power supply $i_3(t)$ currents in the test set and the predictions obtained by the best compressed models.

V. Conclusions

This paper presented a preliminary behavioral modeling framework for IC buffers using the VV-KRR. By focusing on an efficient compression scheme, the method reduces model complexity while maintaining accuracy. This compact representation is easily integrated into SPICE-based solvers, offering a practical solution for improving signal and power integrity simulations.

References

[1] *I/O Buffer Information Specification, Ver. 7.2. Accessed: Dec. 12, 2023 [online]. Available: https://ibis.org/*

[2] G. Signorini, C. Siviero, M. Telescu, I.S. Stievano "Present and future of I/O-buffer behavioral macromodels," IEEE Electromagnetic Compatibility Magazine, vol. 5, no. 3, pp. 79–85, 2016.

[3] B. Mutnury, M. Swaminathan and J.P. Libous, "Macromodeling of nonlinear digital I/O drivers," IEEE Transactions on Advanced Packaging, vol. 29, no. 1, pp. 102–113, Feb. 2006.

[4] M. Souilem, N. Zgolli, T.R. Cunha, W. Dghais and H. Belgacem, "Signal and Power Integrity IO Buffer Modeling Under Separate Power and Ground Supply Voltage Variation of the Input and Output Stages," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 6, pp. 874–886, June 2023.

[5] Y. Cao and Q. -J. Zhang, "A New Training Approach for Robust Recurrent Neural-Network Modeling of Nonlinear Circuits," IEEE Transactions on Microwave Theory and Techniques, vol. 57, no. 6, pp. 1539-1553, June 2009.

[6] H. Yu, T. Michalka, M. Larbi and M. Swaminathan, "Behavioral Modeling of Tunable I/O Drivers With Preemphasis Including Power Supply Noise," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 1, pp. 233–242, Jan. 2020.

[7] R. Trinchero, T. Bradde, M. Telescu, I.S. Stievano "Modeling of IC Buffers from Channel Responses Via Machine Learning Kernel Regression", Proc. *28th IEEE Workshop on Signal and Power Integrity (SPI)*, Lisbon (Pt), May 12-15, 2024.

[8] N. Soleimani, R. Trinchero and F. G. Canavero, "Bridging the Gap Between Artificial Neural Networks and Kernel Regressions for Vector-Valued Problems in Microwave Applications," IEEE Trans. Microw. Theory Tech., vol. 71, no. 6, pp. 2319-2332, 2023.

[9] J.A.K. Suykens, et al., *Least Squares Support Vector Machines*, World Scientific Pub Co Inc, 2002.

[10] J. A. K. Suykens and J. Vandewalle, "Recurrent least squares support vector machines,"in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 47, no. 7, pp. 1109–1114, July 2000.

[11] A. J. Smola and B. Scholkop."Sparse Greedy Matrix Approximation for Machine Learning", in *Proc. of International Conference on Machine Learning (ICML)*, pag. 911–918, 2000.