A Preliminary Study on Kernel Ridge Regression for Parametric Behavioral Modeling of IC Buffers

Original
A Preliminary Study on Kernel Ridge Regression for Parametric Behavioral Modeling of IC Buffers / Yuldashev, D., Atlante, M., Stievano, I.S., Trinchero, R.. - ELETTRONICO. - (2026), pp. 1-4. (2026 IEEE 30th Workshop on Signal and Power Integrity (SPI) Torino (Ita) Jun 14-17, 2026) [10.1109/spi68887.2026.11594754].

Availability:
This version is available at: 11583/3013007 since: 2026-07-13T06:49:29Z

Publisher:
IEEE

Published
DOI:10.1109/spi68887.2026.11594754

A Preliminary Study on Kernel Ridge Regression for Parametric Behavioral Modeling of IC Buffers

Dilyorjon Yuldashev[1*], Marco Atlante[1,2], Igor S. Stievano[1], Riccardo Trinchero[1]
[1] Dept. Electronics and Telecommunications, Politecnico di Torino, Italy
[2] Univ Brest, Lab-STICC, CNRS, UMR 6285, F-29200 Brest, France
[*] corresponding: {dilyorjon.yuldashev@polito.it}

***Abstract*—This paper presents a preliminary parametric modeling scheme for IC buffers, able to reproduce the dynamic behavior of port variables as a function of external parameters. The approach combines a NARX representation with kernel regression and is tested on the output current of a commercial buffer, with temperature as the external parameter. Preliminary results indicate that the proposed method is a promising solution for parameter-dependent dynamic modeling.**

***Index Terms*—Digital integrated circuits, IC buffer modeling, kernel ridge regression, parametric identification, temperature-dependent modeling.**

I. Introduction

Modern information technology devices demand highly accurate design and signal integrity analysis due to increasing performance and technological constraints. Reliable numerical models of digital IC buffers are therefore essential, as they govern signal behavior across interconnections. Traditional transistor-level models of IC buffer, while accurate, are often impractical for system-level simulations due to their high computational cost, proprietary nature and limited availability of detailed models.

Behavioral modeling provides an efficient alternative, aiming to replicate device performance without exposing internal details [1]–[3]. For more than thirty years, the Input/Output Buffer Information Specification (IBIS) has been the industry standard for modeling IC buffers [4]. IBIS offers a structured modeling framework based on equivalent circuits and a predefined two-piece architecture. While this structure simplifies model construction, it can compromise accuracy under complex operating conditions. To address these limitations, machine learning techniques, particularly Artificial Neural Networks (ANNs) and Recurrent Neural Networks (RNNs), have emerged as attractive alternatives thanks to their ability to capture highly nonlinear behavior [5], [6]. However, these methods often benefit from larger datasets, involve a nontrivial training effort, and are not always straightforward to integrate into standard SPICE-based simulation flows. Kernel-based methods can be seen as a promising alternative for system identification and IC behavioral modeling, thanks to their convex and linear training formulation [7]–[10]. Kernel regression, such as the Kernel Ridge Regression (KRR), can be combined with a Nonlinear AutoRegressive with eXogenous inputs (NARX) representation to build compact and accurate data-driven behavioral models able to predict the dynamic behavior of the port variables of IC buffers [11]–[13]. This modeling framework simplifies training and relies on a linear model structure, which can be efficiently implemented within SPICE-like simulation environments [12]. As a result, this approach offers an effective compromise between the rigidity of equivalent-circuit-based methodologies and the complexity of ANN- and RNN-based solutions.

This paper presents a preliminary extension of the above framework to parameter-dependent dynamic modeling. System parameters are incorporated as explicit modeling variables and treated as additional inputs to the regression model. A dedicated training strategy is introduced to exploit transient waveforms collected at different values of the system parameter, enabling the construction of a single unified model capable of predicting the dynamic behavior of the variable of interest over the entire operating space. In this preliminary study, the effectiveness, accuracy, and limitations of the proposed framework are investigated for the dynamic modeling of the output current of a commercial driver as a function of temperature. Temperature is selected as the system parameter because of its significant impact on the dynamic behavior of the port variables of IC buffers [14], [15]. Future work will address more complex scenarios and improved training strategies.

II. Proposed Modeling Approach

Without loss of generality, the discussion focuses on the constitutive relationship governing the port variables of a generic IC driver, illustrated in Fig. 1. This relationship can be expressed by the following nonlinear dynamic map:

$$\begin{aligned} y(t;\boldsymbol{\theta}) = f\Bigg(&\boldsymbol{u}(t;\boldsymbol{\theta}), \frac{d\boldsymbol{u}(t;\boldsymbol{\theta})}{dt}, \ldots, \frac{d^r\boldsymbol{u}(t;\boldsymbol{\theta})}{dt^r}, \\ &\frac{dy(t;\boldsymbol{\theta})}{dt}, \ldots, \frac{d^r y(t;\boldsymbol{\theta})}{dt^r}; \boldsymbol{\theta}\Bigg), \end{aligned} \tag{1}$$

where $f(\cdot;\boldsymbol{\theta})$ is a nonlinear operator parameterized by the system parameters collected in the vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_p]^\mathsf{T}$. The scalar output $y(t;\boldsymbol{\theta}) = i_o(t;\boldsymbol{\theta})$ denotes the current at the output port, while the input (row) vector $\boldsymbol{u}(t;\boldsymbol{\theta}) = [\,v_i(t;\boldsymbol{\theta}), v_o(t;\boldsymbol{\theta})\,]$ collects the input excitation v_i and the output voltage v_o at the load parameterized by $\boldsymbol{\theta}$. The integer r defines the differential order of the system, corresponding to the maximum order of the time derivatives of both output and input variables.

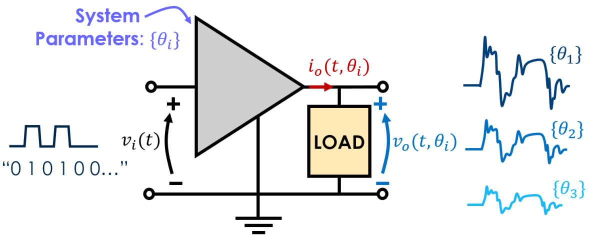

Fig. 1: Generic I/O buffer driven by the input bitstream $v_i(t;\boldsymbol{\theta}_i)$ and operating under different values of the system parameter $\boldsymbol{\theta}_i$. The output responses $v_o(t;\boldsymbol{\theta}_i)$ and $i_o(t;\boldsymbol{\theta}_i)$ depend on the parameter value.

To enable data-driven modeling, the continuous-time relationship (1) is discretized using a uniform sampling period T_s. By approximating the derivative operators with finite backward differences, the continuous map is transformed into a discrete-time NARX formulation [11]–[13]. Defining the sampled output and input at the k-th time step as $y_k(\boldsymbol{\theta}) = i_o(kT_s;\boldsymbol{\theta})$ and $\boldsymbol{u}_k(\boldsymbol{\theta}) = \left[v_i(kT_s;\boldsymbol{\theta}),\, v_o(kT_s;\boldsymbol{\theta})\right]$, the parametric model writes:

$$\hat{y}_k(\boldsymbol{\theta}) = \tilde{f}\big(\boldsymbol{u}_k(\boldsymbol{\theta}),\ldots,\boldsymbol{u}_{k-r}(\boldsymbol{\theta}), y_{k-1}(\boldsymbol{\theta}),\ldots,y_{k-r}(\boldsymbol{\theta});\,\boldsymbol{\theta}\big). \tag{2}$$

Based on (2), the primary modeling goal is to approximate the nonlinear parametric map $\tilde{f}(\cdot;\boldsymbol{\theta})$ from data. To ensure that the model remains valid over the entire parameter space, the training dataset must cover a representative range of dynamic behaviors and parameter configurations. This is achieved through a systematic characterization campaign where the device responses are collected across the admissible variations of the parameter $\boldsymbol{\theta}$.

To this aim the device is simulated under M different load configurations and Q distinct parameter values. For each load and parameter configuration (m, q), a transient simulation is performed, and the resulting waveforms are uniformly resampled with period T_s, yielding a sequence of L discrete-time samples. The complete training dataset $\mathcal{D}$ is constructed by aggregating the data from all configurations:

$$\mathcal{D} = \bigcup_{q=1}^{Q}\bigcup_{m=1}^{M}\mathcal{D}_{m,q}, \tag{3}$$

where each subset $\mathcal{D}_{m,q}$ collects the input–output samples associated with the m-th load at the specific parameter $\boldsymbol{\theta}_q$:

$$\mathcal{D}_{m,q} = \left\{\left(\boldsymbol{x}_l^{(m)}(\boldsymbol{\theta}_q),\, y_l^{(m)}(\boldsymbol{\theta}_q)\right)\right\}_{l=1+r}^{L}. \tag{4}$$

Note that the index l ranges from $1+r$ to L for each individual simulation trace, ensuring that all required past samples are available for the regressor construction. The regressor vector $\boldsymbol{x}_l^{(m)}(\boldsymbol{\theta}_q)$ collects the current input/output samples at time step l and the corresponding past samples for the m-th load configuration and parameter setting $\boldsymbol{\theta}_q$, and is defined as

$$\begin{aligned}\boldsymbol{x}_l^{(m)}(\boldsymbol{\theta}_q) = \big[&\boldsymbol{u}_l^{(m)}(\boldsymbol{\theta}_q),\ldots,\boldsymbol{u}_{l-r}^{(m)}(\boldsymbol{\theta}_q),\\ &y_{l-1}^{(m)}(\boldsymbol{\theta}_q),\ldots,y_{l-r}^{(m)}(\boldsymbol{\theta}_q)\big]^{\mathsf{T}}.\end{aligned} \tag{5}$$

The resulting training dataset is then used together with KRR to learn the parametric map $\tilde{f}$. For a generic load configuration, denoted by the symbol $(*)$, and a parameter setting $\boldsymbol{\theta}_*$, the predicted output at k-th time step $\hat{y}_k^{(*)}(\boldsymbol{\theta}_*)$ is obtained by evaluating the kernel expansion over the entire training set [16]–[19]:

$$\hat{y}_k^{(*)}(\boldsymbol{\theta}_*) = \sum_{\substack{m,q=1\\ l=r+1}}^{M,Q,L}\alpha_{l,m,q}\,K\Big(\big(\boldsymbol{x}_l^{(m)}(\boldsymbol{\theta}_q),\boldsymbol{\theta}_q\big),\,\big(\boldsymbol{x}_k^{(*)}(\boldsymbol{\theta}_*),\boldsymbol{\theta}_*\big)\Big), \tag{6}$$

where $\alpha_{l,m,q}$ are the regression coefficients estimated during training for the training input with parameter value $\boldsymbol{\theta}_q$, load m and time instant l, while $\boldsymbol{x}_l^{(m)}(\boldsymbol{\theta}_q)$ and $\boldsymbol{x}_k^{(*)}(\boldsymbol{\theta}_*)$ are the input and test regressor defined in (5).

The presence of past predictions $\hat{y}_{k-1}^{(*)}(\boldsymbol{\theta}_*),\ldots,\hat{y}_{k-r}^{(*)}(\boldsymbol{\theta}_*)$ in the regressor $\boldsymbol{x}_k^{(*)}(\boldsymbol{\theta}_*)$ makes the model recursive, which is essential for multi–step–ahead dynamic prediction. During prediction, the parameter value $\boldsymbol{\theta}_*$ is assumed to be known and constant for each time step k.

The kernel function $K(\cdot,\cdot)$ in (6) is defined as a separable kernel, decomposing the overall similarity into independent similarity measures in the dynamic ($\boldsymbol{x}$) and parametric ($\boldsymbol{\theta}$) domains [7], [10]. Considering two generic training samples indexed by (l, m, q) and (l', m', q'), the kernel evaluation writes:

$$K\big((\boldsymbol{x}_l^{(m)}(\boldsymbol{\theta}_q),\boldsymbol{\theta}_q),\,(\boldsymbol{x}_{l'}^{(m')}(\boldsymbol{\theta}_{q'}),\boldsymbol{\theta}_{q'})\big) = \tag{7}$$

$$k_{\boldsymbol{x}}(\boldsymbol{x}_l^{(m)}(\boldsymbol{\theta}_q),\boldsymbol{x}_{l'}^{(m')}(\boldsymbol{\theta}_q))\cdot k_{\boldsymbol{\theta}}(\boldsymbol{\theta}_q,\boldsymbol{\theta}_{q'}). \tag{8}$$

The regression model is then trained by solving the regularized linear system [8], [17]:

$$\boldsymbol{\alpha} = \left(\mathbf{K} + \lambda\mathbf{I}_{N_{\mathrm{tr}}}\right)^{-1}\boldsymbol{y}, \tag{9}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{N_{\mathrm{tr}}}$ collects the regression coefficients, $\mathbf{I}_{N_{\mathrm{tr}}}$ denotes the identity matrix of size $N_{\mathrm{tr}} \times N_{\mathrm{tr}}$, and λ is the Tikhonov regularization parameter used to ensure numerical stability and to balance data fidelity and model smoothness. The vector $\boldsymbol{y} \in \mathbb{R}^{N_{\mathrm{tr}}}$ stacks the training outputs $y_{l,m}(\boldsymbol{\theta}_q)$ for all parameter values $q = 1,\ldots,Q$, load configurations $m = 1,\ldots,M$, and time indices $l = r+1,\ldots,L$. The total number of training samples is $N_{\mathrm{tr}} = Q\,M\,(L-r)$.

When the Gram matrix $\mathbf{K}$ is rank-deficient or exhibits rapidly decaying eigenvalues, the learning problem becomes ill-posed and the condition number $\mathrm{cond}(\mathbf{K})$ becomes very large. The Tikhonov regularization term $\lambda\mathbf{I}_{N_{\mathrm{tr}}}$ shifts the eigenvalues of $\mathbf{K}$ away from zero, improves the conditioning, and stabilizes the matrix inversion. As a result, the regularization

TABLE I: Training and Test Setup: Overview of the load configurations and temperature values used for constructing the training and validation datasets. The table summarizes the distinct load types, their corresponding electrical characteristics, and the operating temperatures.

Load ID	**Load Configuration Description**	**Simulation Temperatures (T [°C])**
#L1 (Training)	Open-ended transmission line with a 60 Ω characteristic impedance and a 10 ns delay.	$0, 35, 70$
#L2 (Training)	Open-ended transmission line with a 75 Ω characteristic impedance and a 4 ns delay.	$0, 35, 70$
#L3 (Training)	Lumped 50 Ω resistor connected in parallel with a 10 pF capacitor.	$0, 35, 70$
#L4 (Training)	Lumped 150 Ω resistor.	$0, 35, 70$
#L5 (**Test**)	Open-ended transmission line with a 50 Ω characteristic impedance and a 3 ns delay.	$20, 60$

parameter λ plays a crucial role in ensuring numerical stability of the estimator.

In this preliminary study, a radial basis function (RBF) kernel [18] is adopted for both $k_{\boldsymbol{x}}$ and $k_{\boldsymbol{\theta}}$:

$$k_{\boldsymbol{x}/\boldsymbol{\theta}}(\mathbf{r}, \mathbf{r}'; \sigma_{\boldsymbol{x}/\boldsymbol{\theta}}) = \exp\left(-\frac{\|\mathbf{r} - \mathbf{r}'\|^2}{2\,\sigma^2_{\boldsymbol{x}/\boldsymbol{\theta}}}\right), \tag{10}$$

where $\mathbf{r}$ denotes either the input regressor $\boldsymbol{x}$ (for $k_{\boldsymbol{x}}$) or the parameter vector $\boldsymbol{\theta}$ (for $k_{\boldsymbol{\theta}}$), and $\sigma_{\boldsymbol{x}/\boldsymbol{\theta}}$ are the corresponding kernel hyperparameters. Other kernel choices could be adopted as well.

The modeling approach proposed in this section constitutes a preliminary training strategy for the parametric model, explicitly addressing the potential ill-posedness of the resulting linear system. Future work will investigate more advanced training schemes to further improve conditioning and computational efficiency.

III. Application Example

The performance of the parametric modeling scheme presented in Sec. II is evaluated by predicting the transient behavior of the output current of a commercial buffer under different temperature conditions. The analysis relies on a transistor-level model of the Texas Instruments transceiver SN74ALVCH16973, powered with a nominal supply voltage of $V_{DD} = 1.8\,\text{V}$. The HSPICE model provided by the manufacturer is used as the reference.

Training and test datasets are obtained from HSPICE transient simulations performed over multiple load configurations and temperature values (i.e., $p = 1$), as summarized in Table I. The training dataset includes four distinct load configurations #L1–#L4 (i.e., $M = 4$), each simulated at three temperature values (i.e., $Q = 3$) such that: $\theta = 0$°C, 35°C, and 70°C. These temperatures span a representative portion of the device ambient operating range. For all training simulations, the device is driven by the same input bitstream “0101100000”, ensuring consistent excitation across different operating conditions. The test dataset is designed to assess model performance under unseen conditions. It includes simulation results for load configuration #L5, specified in Table I, at two intermediate temperatures not used during training, namely $\theta_* = 20$°C and 60°C. A different input bitstream, “0110100000”, is applied to further evaluate the generalization capability of the model. The training and test waveforms in the dataset are sampled with a sampling period of $T_s = 285\,\text{ps}$ up to $T_{\max} = 300\,\text{ns}$, resulting in $L = 670$ discrete-time samples for each waveform.

The training set is used together with the training scheme described in Sec. II, adopting a dynamic order $r = 4$, selected based on empirical observations. The resulting training dataset consists of a total of $QM(L - r) = 7992$ samples. Model training using the three-fold cross validation requires approximately 5 minutes on a Dell Precision 3490 equipped with an Intel Core Ultra 7 processor and 64 GB of RAM. MATLAB R2024b.

Due to the poor conditioning of the Gram matrices $\mathbf{K}_x$ and $\mathbf{K}_{\boldsymbol{\theta}}$, and thus of their product $\mathbf{K}$, caused respectively by strong correlations among the training waveforms and by repeated values of $\boldsymbol{\theta}$ within each load configuration, an unregularized model tends to produce undesirable numerical artifacts, such as spurious high-frequency oscillations in the predicted responses. For this reason, the regularization parameter λ plays a key role in the proposed scheme and is optimized within the interval $[10^{-2}, 1]$. This range includes relatively large values of λ, which promote model smoothness and prevent unwanted fast fluctuations in the predictions.

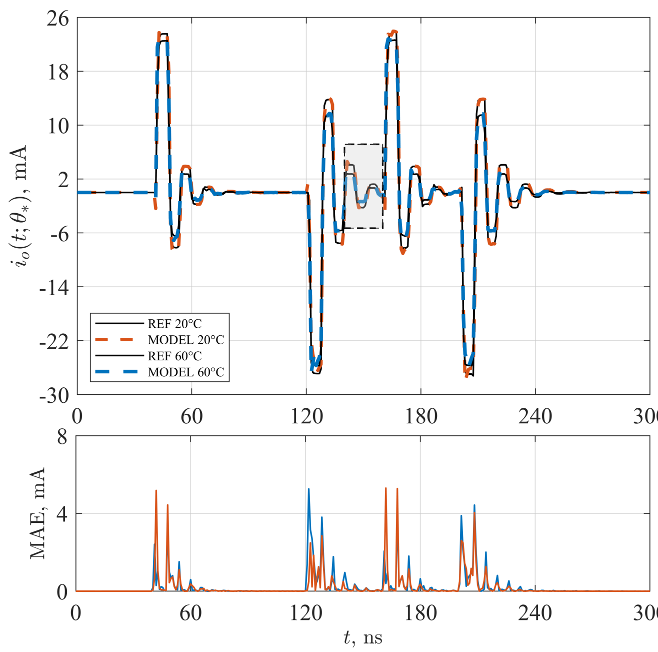

Fig. 2: Validation results for the test load configuration #L5. Top: comparison between the reference output current $i_o(t;\theta_*)$ (solid black) and the model predictions (dashed) at $\theta_* = 20$°C and 60°C. Bottom: MAE for the two test temperatures.

Figure 2 shows the validation results on the unseen load

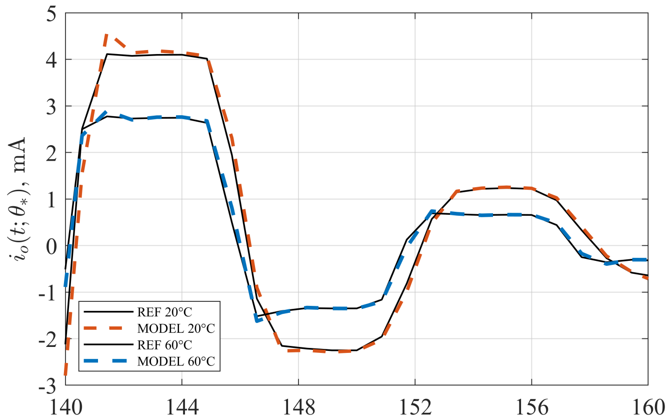

Fig. 3: Magnified view of the test waveform for load configuration #L5. The zoomed time window compares the reference output current $i_o(t;\theta_*)$ (solid black) with the model predictions (dashed) at $\theta_* = 20°$C and 60°C.

configuration #L5. The top plot compares the reference output current $i_o(t;\theta_*)$ (solid black) with the model predictions (dashed) at the test temperatures $\theta_* = 20°$C and $\theta_* = 60°$C. The bottom plot reports the mean absolute error (MAE) between the model predictions and the reference waveform for both test temperatures.

Figure 3 provides a zoomed view of a representative time window of the results in Fig. 2 (see the light gray rectangle). The zoom highlights the significant effect of temperature on the dynamic behavior of the output current waveforms and shows the close agreement between the reference response and the model predictions at the considering test temperatures.

The model accurately reproduces the buffer response at both unseen temperature values and on an unseen load configuration, as the MAE remains relatively low throughout the transient, confirming the practical applicability of the approach. Nevertheless, further improvements to the training scheme are needed to reduce training time and to improve the conditioning of the linear system solved during model identification.

IV. Conclusions & Future Work

This paper presented a preliminary modeling scheme aimed at providing a unified parametric model able to reproduce the dynamic behavior of the port variables of IC buffers as a function of external parameters. The proposed approach combines a NARX formulation of the device constitutive relations with kernel regression.

The performance of the method was assessed on the modeling of the output port current waveforms of a commercial IC buffer for different load configurations, using temperature as the external parameter. The results demonstrate the capability of the proposed framework to accurately predict the parametric dynamic behavior of the output current for unseen load and temperature values. However, this preliminary study also highlighted issues related to the poor conditioning of the training problem, which leads to increased training time and requires careful tuning of the regularization parameter. These aspects, together with the possible circuit implementation of the resulting model, will be investigated in future work.

References

[1] 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.

[2] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, “Nonlinear black-box modeling in system identification: a unified overview,” *Automatica*, vol. 31, no. 12, pp. 1691–1724, Dec. 1995.

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

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

[5] W. Liu, L. Zhu, F. Feng, W. Zhang, Q.-J. Zhang, Q. Lin, and G. Liu, “A Time Delay Neural Network Based Technique for Nonlinear Microwave Device Modeling,” *Micromachines*, vol. 11, no. 9, art. 831, 2020.

[6] Yi Cao, Runtao Ding and Qi-Jun Zhang, “State-space dynamic neural network technique for high-speed IC applications: modeling and stability analysis,” in IEEE Trans. Microw. Theory Tech., vol. 54, no. 6, pp. 2398–2409, June 2006.

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

[8] J. Shawe-Taylor and N. Cristianini, *Kernel methods: an overview.* Kernel Methods for Pattern Analysis, Cambridge: Cambridge University Press, 2004, pp. 25–46.

[9] B. Scholkopf, R. Herbrich, and A. J. Smola, “A generalized representer theorem”,in *Proc. 14th Annu. Conf. Comput. Learn. Theory*, 2001, pp. 416–426.

[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] M. Atlante, R. Trinchero, I.S. Stievano, M. Telescu, N. Tanguy, “SPICE-Based Behavioral Models of IC Buffers via Compact Kernel Regressions,” 2025 IEEE 29th Workshop on Signal and Power Integrity (SPI), May 2025.

[12] M. Atlante, R. Trinchero, I.S. Stievano, M. Telescu, N. Tanguy, “IC Modeling via Machine Learning Regressions: A Data-Driven Approach to SPICE Integration,” IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 15, no. 9, pp. 1814–1822, Sept. 2025.

[13] A. Kadochnikova, Y. Zhu, Z.-Q. Lang, V. Kadirkamanathan, “Integrated Identification of the Nonlinear Autoregressive Models With Exogenous Inputs (NARX) for Engineering Systems Design,” in IEEE Transactions on Control Systems Technology, vol. 31, no. 1, pp. 394–401, Jan. 2023

[14] W. Dghais, M. Souilem, F. Zayer, and A. Chaari, “Power supply-and temperature-aware I/O buffer model for signal-power integrity simulation,” Mathematical Problems in Engineering, vol. 2018, Article ID 1356538, 9 pages, 2018.

[15] I.S. Stievano, F.G. Canavero, I.A. Maio, “Behavioral Modeling of IC Ports Including Temperature Effects,” IEEE Transactions on Advanced Packaging, vol. 27, no. 1, pp. 200–207, Feb. 2004.

[16] A. Rudi, L. Carratino, and L. Rosasco, “Falkon: An optimal large scale kernel method,” Advances in neural information processing systems, vol. 30, 2017.

[17] 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.

[18] N. Soleimani, P. Manfredi and R. Trinchero, “Efficient Implementation of the Vector-Valued Kernel Ridge Regression for the Uncertainty Quantification of the Scattering Parameters of a 2-GHz Low-Noise Amplifier,” *2023 IEEE MTT-S Int. Conf. Numerical Electromagnetic and Multiphysics Modeling and Optim. (NEMO)*, Winnipeg, MB, Canada, 2023.

[19] F. Alavi, S. Hashemi, “An element-wise kernel learning framework,” Applied Intelligence, vol. 53, no. 8, pp. 9531–9547, Jun. 2023.