POLITECNICO DI TORINO
Repository ISTITUZIONALE

Learning Compressed Parametric Dynamic Models for High-Speed Electronic Systems via Nyström Kernel Ridge Regression

Original
Learning Compressed Parametric Dynamic Models for High-Speed Electronic Systems via Nyström Kernel Ridge Regression / Yuldashev, D., Stievano, I.S., Trinchero, R.. - In: IEEE TRANSACTIONS ON COMPONENTS, PACKAGING, AND MANUFACTURING TECHNOLOGY. - ISSN 2156-3950. - ELETTRONICO. - (In corso di stampa), pp. 1-13. [10.1109/TCPMT.2026.3752459]

Availability:
This version is available at: 11583/3017828 since: 2026-11-17T04:17:13Z

Publisher:
IEEE

Published
DOI:10.1109/TCPMT.2026.3752459

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

©9999 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)

29 November 2026

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

Learning Compressed Parametric Dynamic Models for High-Speed Electronic Systems via Nyström Kernel Ridge Regression

Dilyorjon Yuldashev, *Student Member, IEEE,* Igor S. Stievano, *Senior Member, IEEE,* Riccardo Trinchero, *Member, IEEE*

***Abstract*—This paper proposes a compressed parametric dynamic modeling framework for high-speed electronic systems based on Kernel Ridge Regression. The approach combines a NARX representation with a separable kernel that accounts for both dynamic input-output histories and external parameter variations. A Nyström-compressed formulation is introduced to handle the large training datasets generated by transient simulations while limiting the number of model coefficients. The method is assessed on three signal-integrity examples, ranging from IC-buffer modeling to high-speed interconnects with linear and non-linear terminations with multiple geometrical parameters. The results demonstrate accurate transient and eye-diagram prediction with reduced training time and model complexity.**

***Index Terms*—parametric system identification, behavioral modeling, Kernel Ridge Regression, Nyström approximation, signal integrity, high-speed electronic systems.**

I. Introduction

Modern electronic systems require predictive models able to accurately reproduce dynamic responses over different operating conditions, parameter configurations, and excitation scenarios. This requirement is especially important in signal- and power-integrity analyses, where the observed transient behavior may be affected by device nonlinearities, interconnect propagation effects, loading conditions, and technological variability. In this context, compact data-driven behavioral models provide an attractive alternative to repeated circuit-level transient simulations, enabling efficient design exploration and system-level verification.

The construction of such models can be regarded as a parametric dynamic system identification problem. This differs from standard surrogate modeling in two main aspects. First, surrogate models usually learn a static map between a set of parameters and a quantity of interest [1], whereas dynamic system identification aims at learning an input-output dynamic map from transient data [2]. Second, in dynamic identification each transient simulation generates a complete time sequence, so that the number of regression samples depends not only on the number of simulated parameter configurations and excitation conditions, but also on the number of time samples in each waveform. Therefore, even a moderate number of transient simulations may lead to large training datasets.

Relevant application examples arise in high-speed electronic design, including IC buffers, interconnects, and complete links with linear or nonlinear terminations. In these cases, accurate and compact behavioral models are needed to predict transient waveforms and eye-diagram metrics under unseen operating conditions [3]–[5]. Standard modeling solutions, such as IBIS-based representations for I/O buffers [6], are widely used in industrial simulation flows. However, fixed-structure behavioral descriptions may become restrictive when additional parameter dependencies, general input-output relationships, or complex dynamic effects must be learned from data.

To address these challenges, data-driven behavioral modeling techniques have received increasing attention. Artificial Neural Networks (ANNs) and Recurrent Neural Networks (RNNs) have been investigated because of their ability to approximate nonlinear dynamic systems [7]–[9]. However, these approaches may involve nonconvex training procedures and their integration into conventional SPICE-like simulation environments can be less straightforward.

Kernel-based learning methods represent a promising alternative for nonlinear system identification and behavioral modeling [10], [11]. Kernel Ridge Regression (KRR), in particular, combines nonlinear approximation capabilities with a convex training formulation, leading to a linear system of equations for the determination of the model coefficients. When combined with a Nonlinear AutoRegressive with eXogenous inputs (NARX) representation, KRR can be used to construct dynamic models from transient input-output data [12]–[14]. In this framework, the relevant dynamic information is encoded in a finite history of past input and output samples, without explicitly modeling the internal state variables.

A preliminary version of this approach was presented in [15], where KRR-based NARX modeling was extended to account for parameter-dependent dynamic behavior and was applied to the port-current modeling of a commercial IC buffer as a function of temperature [16]. Although the results demonstrated the potential of the method, the preliminary formulation had two main limitations. First, the standard KRR training procedure required the construction and solution of a linear system whose size scales with the total number of training samples. This limited the amount of transient data that could be practically exploited. Second, a single autoregressive order was used for both input and output histories, which

Dilyorjon Yuldashev is with Turin Polytechnic University in Tashkent, 100095 Tashkent, Uzbekistan, and also with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy.
Igor Simone Stievano and Riccardo Trinchero are with the Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy (corresponding author: Riccardo Trinchero; e-mail: riccardo.trinchero@polito.it).

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

may be restrictive for systems characterized by propagation delays or by different effective memory lengths in the input and output variables.

This paper presents an extended formulation of the above KRR-NARX framework for parametric dynamic system identification. The proposed approach introduces two independent memory orders, r_u and r_y, for the input and output histories, respectively, and explicitly includes the system parameters in the regression variables. Moreover, a Nyström-compressed training formulation is introduced to improve scalability by reducing the number of model coefficients while still exploiting the information contained in the complete training dataset.

The present work extends the preliminary conference contribution in [15] by addressing both device-level and system-level parametric dynamic modeling problems [17], [19]–[24], [39]. The first application revisits the IC-buffer output-current modeling problem and extends it through the proposed compressed formulation. The second application considers the input-output modeling of a high-speed link with linear terminations whose response depends on two geometrical parameters. The third application further extends the validation to a high-speed link with non-linear IBIS-based driver and receiver terminations and five varying geometrical parameters. These examples demonstrate the applicability of the proposed method to problems of increasing complexity, ranging from device-level modeling to system-level high-speed link prediction.

The remainder of the paper is organized as follows. Section II introduces the parametric NARX formulation, the separable-kernel KRR model, and the Nyström-compressed training strategy. Section III presents the numerical applications and discusses the accuracy and computational performance of the proposed approach. Finally, Section IV draws the main conclusions and outlines possible future developments.

II. Parametric System Identification via Kernel Methods

A. Problem Formulation and Parametric NARX Dataset

Let us consider the problem of constructing a behavioral, data-driven model able to reproduce the input-output behavior of a parametric nonlinear dynamical system from a limited set of observations. The system is assumed to depend on a set of parameters collected in the vector

$$\boldsymbol{\theta} = [\theta_1, \dots, \theta_p]^{\mathsf{T}}. \tag{1}$$

A general continuous-time description of such a system can be written as

$$y(t;\boldsymbol{\theta}) = f\Bigg(\boldsymbol{u}(t;\boldsymbol{\theta}), \frac{d\boldsymbol{u}(t;\boldsymbol{\theta})}{dt}, \dots, \frac{d^{r_u}\boldsymbol{u}(t;\boldsymbol{\theta})}{dt^{r_u}}, \frac{dy(t;\boldsymbol{\theta})}{dt}, \dots, \frac{d^{r_y}y(t;\boldsymbol{\theta})}{dt^{r_y}}; \boldsymbol{\theta}\Bigg), \tag{2}$$

where $y(t;\boldsymbol{\theta}) \in \mathbb{R}$ is the scalar output variable, $\boldsymbol{u}(t;\boldsymbol{\theta}) \in \mathbb{R}^{n_u}$ is the input vector, and $f(\cdot;\boldsymbol{\theta})$ is a nonlinear map parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$. The integers r_u and r_y denote the maximum orders of the input and output derivatives, respectively. In the following, different values of r_u and r_y are allowed in order to provide additional flexibility in the representation of systems whose input and output histories may have different effective memory lengths.

To obtain a discrete-time input-output model, the continuous-time relation in (2) is sampled with a uniform sampling period T_s. The time derivatives are then approximated using backward finite differences. For instance, for a generic variable $x(t)$:

$$\left.\frac{dx(t)}{dt}\right|_{t=t_k} \approx \frac{x_k - x_{k-1}}{T_s}, \tag{3}$$

where $x_k = x(t_k)$ and $t_k = (k-1)T_s$. This leads to the following parametric one-step-ahead discrete-time model:

$$\hat{y}_k(\boldsymbol{\theta}) = \tilde{f}\big(\boldsymbol{u}_k(\boldsymbol{\theta}), \dots, \boldsymbol{u}_{k-r_u}(\boldsymbol{\theta}), y_{k-1}(\boldsymbol{\theta}), \dots, y_{k-r_y}(\boldsymbol{\theta}); \boldsymbol{\theta}\big), \tag{4}$$

where $\tilde{f}$ is the discrete-time counterpart of the continuous nonlinear map. Equation (4) defines a parametric NARX-type model [25]. The internal state variables of the system are not explicitly modeled; instead, the relevant dynamic information is embedded in a finite history of past input and output samples [26], [27].

The present formulation considers a scalar target output, i.e., each trained model predicts one selected output quantity. Systems with multiple ports or output quantities can be addressed by constructing a separate model for each selected output. A simultaneous multi-output formulation could in principle be obtained using vector-valued kernel regression [28]; however, its extension to the large-scale recurrent setting considered here requires a dedicated assessment and is outside the scope of this work.

The training data are generated by simulating or measuring the system under M different working conditions, such as different input excitations or load configurations, and for Q different parameter configurations $\boldsymbol{\theta}_q$, with $q = 1, \dots, Q$. For each pair (m, q), a transient response is obtained and uniformly sampled, yielding L time samples.

For each working condition $m = 1, \dots, M$, parameter configuration $q = 1, \dots, Q$, and admissible time index $l = r+1, \dots, L$, where $r = \max(r_u, r_y)$, we define the dynamic NARX regressor of dimension $(1 + r_u)n_u + r_y$:

$$\boldsymbol{\xi}_{l,m,q} = \begin{bmatrix} \boldsymbol{u}_l^{(m)}(\boldsymbol{\theta}_q) \\ \vdots \\ \boldsymbol{u}_{l-r_u}^{(m)}(\boldsymbol{\theta}_q) \\ y_{l-1}^{(m)}(\boldsymbol{\theta}_q) \\ \vdots \\ y_{l-r_y}^{(m)}(\boldsymbol{\theta}_q) \end{bmatrix}. \tag{5}$$

The corresponding parametric regressor is obtained by augmenting the dynamic regressor with the parameter vector:

$$\boldsymbol{z}_{l,m,q} = \begin{bmatrix} \boldsymbol{\xi}_{l,m,q} \\ \boldsymbol{\theta}_q \end{bmatrix}, \qquad y_{l,m,q} = y_l^{(m)}(\boldsymbol{\theta}_q). \tag{6}$$

The triplet (l, m, q) is useful to introduce the local structure of the regression samples associated with each transient response. If the QM waveforms were treated independently, the first r samples of every waveform would have to be

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

discarded, resulting in $QM(L-r)$ regression samples. In the adopted implementation, however, the transient waveforms are concatenated before constructing the NARX regression matrix. All simulations are generated as zero-state responses and include a sufficiently long final zero interval, so that the transient response has decayed to zero before the beginning of the following waveform. Consequently, the concatenation does not introduce an artificial nonzero initial condition, and only the first r samples of the complete concatenated sequence must be discarded. The resulting number of regression samples is therefore:

$$N_{\mathrm{tr}} = QML - r. \tag{7}$$

Compared with treating the waveforms independently, this procedure preserves an additional $r(QM-1)$ training samples and is particularly beneficial when the model order r is not negligible with respect to the waveform length L. The training set can then be written compactly using a single global index as

$$\mathcal{D}_{\mathrm{tr}} = \{(\boldsymbol{z}_n, y_n)\}_{n=1}^{N_{\mathrm{tr}}}. \tag{8}$$

The global index n enumerates the samples of the concatenated sequence, while the local indices (l, m, q) identify the corresponding time sample, operating condition, and parameter configuration. Similarly, $\boldsymbol{z}_n$ can be decomposed as

$$\boldsymbol{z}_n = \begin{bmatrix} \boldsymbol{\xi}_n \\ \boldsymbol{\theta}_n \end{bmatrix}, \tag{9}$$

where $\boldsymbol{\xi}_n$ is the dynamic part and $\boldsymbol{\theta}_n$ is the parameter vector associated with the n-th training sample.

B. Kernel Ridge Regression with a Separable Kernel

The goal of the proposed approach is to approximate the nonlinear parametric map in (4) from the training set $\mathcal{D}_{\mathrm{tr}}$. To this end, KRR is adopted because it provides a nonlinear regression model with a convex training problem [29]–[31].

Given a generic test regressor

$$\boldsymbol{z}_* = \begin{bmatrix} \boldsymbol{\xi}_* \\ \boldsymbol{\theta}_* \end{bmatrix}, \tag{10}$$

the KRR prediction is written as

$$\hat{y}_* = \sum_{n=1}^{N_{\mathrm{tr}}} \alpha_n K(\boldsymbol{z}_n, \boldsymbol{z}_*), \tag{11}$$

where α_n are the regression coefficients and $K(\cdot,\cdot)$ is the kernel function.

Since each regressor contains both dynamic variables and system parameters, a separable kernel is adopted [32], [33]. The kernel is defined as the product of a kernel acting on the dynamic NARX regressors and a kernel acting on the parameter space:

$$K(\boldsymbol{z}_n, \boldsymbol{z}_{n'}) = k_\xi(\boldsymbol{\xi}_n, \boldsymbol{\xi}_{n'})\, k_\theta(\boldsymbol{\theta}_n, \boldsymbol{\theta}_{n'}). \tag{12}$$

Here, k_ξ measures the similarity between two dynamic input-output histories, whereas k_θ measures the similarity between two parameter configurations.

Let $\mathbf{K} \in \mathbb{R}^{N_{\mathrm{tr}} \times N_{\mathrm{tr}}}$ be the Gram matrix associated with the kernel K, whose entries are

$$[\mathbf{K}]_{nn'} = K(\boldsymbol{z}_n, \boldsymbol{z}_{n'}). \tag{13}$$

By using the separable kernel in (12), the Gram matrix can be written as the Hadamard product [34], [35]

$$\mathbf{K} = \mathbf{K}_\xi \odot \mathbf{K}_\theta, \tag{14}$$

where $[\mathbf{K}_\xi]_{nn'} = k_\xi(\boldsymbol{\xi}_n, \boldsymbol{\xi}_{n'})$ and $[\mathbf{K}_\theta]_{nn'} = k_\theta(\boldsymbol{\theta}_n, \boldsymbol{\theta}_{n'})$.

The KRR coefficients are obtained by solving the regularized least-squares problem

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{N_{\mathrm{tr}}}} \|\boldsymbol{y} - \mathbf{K}\boldsymbol{\alpha}\|_2^2 + \lambda \boldsymbol{\alpha}^{\mathsf{T}} \mathbf{K} \boldsymbol{\alpha}, \tag{15}$$

where $\boldsymbol{y} = [y_1, \ldots, y_{N_{\mathrm{tr}}}]^{\mathsf{T}}$ collects the training outputs and $\lambda > 0$ is the Tikhonov regularization parameter. The solution is obtained by solving

$$(\mathbf{K} + \lambda \mathbf{I}_{N_{\mathrm{tr}}})\, \boldsymbol{\alpha} = \boldsymbol{y}, \tag{16}$$

or, equivalently,

$$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I}_{N_{\mathrm{tr}}})^{-1}\, \boldsymbol{y}. \tag{17}$$

Although this formulation is straightforward, its direct application becomes computationally demanding when $N_{\mathrm{tr}} = QML - r$ is large. In the present setting, the number of training samples scales with the product of the number of excitation or load configurations, the number of parameter configurations, and the number of usable time samples extracted from each transient simulation. Consequently, N_{tr} can easily reach several tens or hundreds of thousands. The corresponding full Gram matrix $\mathbf{K}$ in (16) has size $N_{\mathrm{tr}} \times N_{\mathrm{tr}}$, leading to high memory requirements and expensive training. This issue becomes even more critical during hyperparameter tuning, since the KRR problem must be solved repeatedly for different kernel hyperparameters and regularization values.

Moreover, adjacent time samples extracted from transient waveforms are often strongly correlated. As a consequence, several training regressors may be very similar, leading to nearly dependent rows and columns in the Gram matrix. This can deteriorate the conditioning of the linear system in (16). The regularization parameter λ improves numerical stability, but an excessively large value may increase the bias of the estimator and reduce prediction accuracy. Therefore, λ and the kernel hyperparameters must be selected jointly in order to balance accuracy, smoothness, and numerical robustness [15]. These issues motivate the use of a compressed formulation based on the Nyström approximation.

C. Nyström-Compressed Training

The Nyström method provides a structured approximation of the full Gram matrix by using only a subset of representative training samples, referred to as landmarks [36]. Unlike a simple reduction of the training set, the Nyström approximation still exploits all the available training samples through the kernel evaluations between the full dataset and the landmark subset [37], [38].

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

Let $\mathcal{S} = \{s_1, \dots, s_{N_s}\} \subset \{1, \dots, N_{\mathrm{tr}}\}$, with $N_s \ll N_{\mathrm{tr}}$, denote the set of landmark indices. The Nyström approximation of the full Gram matrix is written as

$$\mathbf{K} \approx \widetilde{\mathbf{K}} = \mathbf{K}_{NS} \mathbf{K}_{SS}^{\dagger} \mathbf{K}_{NS}^{\mathsf{T}}, \tag{18}$$

where $\mathbf{K}_{NS} \in \mathbb{R}^{N_{\mathrm{tr}} \times N_s}$ collects the kernel evaluations between all training samples and the landmark samples, while $\mathbf{K}_{SS} \in \mathbb{R}^{N_s \times N_s}$ is the Gram matrix restricted to the landmark samples. Their entries are defined as $[\mathbf{K}_{NS}]_{ij} = K(\boldsymbol{z}_i, \boldsymbol{z}_{s_j})$ and $[\mathbf{K}_{SS}]_{ij} = K(\boldsymbol{z}_{s_i}, \boldsymbol{z}_{s_j})$, respectively. The symbol $(\cdot)^{\dagger}$ denotes the Moore–Penrose pseudo-inverse.

Since the kernel K is separable, the Nyström matrices inherit the same Hadamard-product structure. In particular,

$$\mathbf{K}_{NS} = \mathbf{K}_{\xi,NS} \odot \mathbf{K}_{\theta,NS}, \tag{19}$$

where $[\mathbf{K}_{\xi,NS}]_{ij} = k_{\xi}(\boldsymbol{\xi}_i, \boldsymbol{\xi}_{s_j})$ and $[\mathbf{K}_{\theta,NS}]_{ij} = k_{\theta}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_{s_j})$. Similarly,

$$\mathbf{K}_{SS} = \mathbf{K}_{\xi,SS} \odot \mathbf{K}_{\theta,SS}. \tag{20}$$

The compressed model is expressed as an expansion over the landmark samples only:

$$\hat{y}_* = \sum_{j=1}^{N_s} \beta_j K(\boldsymbol{z}_{s_j}, \boldsymbol{z}_*). \tag{21}$$

Using the separable kernel, this can be written explicitly as

$$\hat{y}_* = \sum_{j=1}^{N_s} \beta_j k_{\xi}(\boldsymbol{\xi}_{s_j}, \boldsymbol{\xi}_*) k_{\theta}(\boldsymbol{\theta}_{s_j}, \boldsymbol{\theta}_*). \tag{22}$$

The vector of compressed coefficients $\boldsymbol{\beta} \in \mathbb{R}^{N_s}$ is obtained by solving the reduced regularized least-squares problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{N_s}} \|\boldsymbol{y} - \mathbf{K}_{NS}\boldsymbol{\beta}\|_2^2 + \lambda \boldsymbol{\beta}^{\mathsf{T}} \mathbf{K}_{SS} \boldsymbol{\beta}. \tag{23}$$

The corresponding solution is

$$\boldsymbol{\beta} = \left(\mathbf{K}_{NS}^{\mathsf{T}} \mathbf{K}_{NS} + \lambda \mathbf{K}_{SS}\right)^{-1} \mathbf{K}_{NS}^{\mathsf{T}} \boldsymbol{y}. \tag{24}$$

In practice, the reduced system may still be affected by ill-conditioning, especially when the transient samples are highly correlated or when product kernels lead to nearly dependent landmark features. Therefore, a diagonal stabilization term is added to the reduced system, leading to

$$\boldsymbol{\beta} = \left(\mathbf{K}_{NS}^{\mathsf{T}} \mathbf{K}_{NS} + \lambda \mathbf{K}_{SS} + \lambda \mathbf{I}_{N_s}\right)^{-1} \mathbf{K}_{NS}^{\mathsf{T}} \boldsymbol{y}. \tag{25}$$

Compared with the full KRR formulation in (16), the Nyström-compressed model requires the estimation of only N_s coefficients, with $N_s \ll N_{\mathrm{tr}}$. Moreover, the main linear system to be solved has size $N_s \times N_s$, while the information from the whole training set is retained through the matrix product $\mathbf{K}_{NS}^{\mathsf{T}} \mathbf{K}_{NS}$. This provides a compact and computationally efficient approximation of the full KRR model, while preserving the separable kernel structure used to account for both dynamic and parametric similarities.

The accuracy of the Nyström approximation depends on the selected landmark samples. A purely random selection is simple, but may introduce additional variability in the model accuracy [13]. In this work, the landmarks are selected through a deterministic farthest-point strategy applied to the training regressors. The first landmark is chosen as the sample closest to the empirical mean of the training regressors. The remaining landmarks are then selected iteratively. At iteration j, the indices $s_1, \dots, s_{j-1}$ have already been selected. For each candidate sample $\boldsymbol{z}_i$, its distance from the set of already-selected landmarks is defined as the distance from the closest selected landmark,

$$d_i^{(j)} = \min_{k=1,\dots,j-1} \|\boldsymbol{z}_i - \boldsymbol{z}_{s_k}\|_2^2 . \tag{26}$$

The next landmark is then chosen as the sample with the largest distance from this selected set,

$$s_j = \arg\max_i d_i^{(j)}, \qquad j = 2, \dots, N_s. \tag{27}$$

This deterministic rule promotes a well-spread coverage of the training input space without requiring additional tuning parameters.

During recursive prediction, the test dynamic regressor is updated using the available input samples $\boldsymbol{u}_{k-r_u}, \dots, \boldsymbol{u}_k$ and the previously predicted output values $\hat{y}_{k-r_y}, \dots, \hat{y}_{k-1}$. To distinguish this quantity from the training regressors built from measured outputs, we denote it by $\hat{\boldsymbol{\xi}}_k$, namely

$$\hat{\boldsymbol{\xi}}_k = \left[\boldsymbol{u}_k^{\mathsf{T}}, \dots, \boldsymbol{u}_{k-r_u}^{\mathsf{T}}, \hat{y}_{k-1}, \dots, \hat{y}_{k-r_y}\right]^{\mathsf{T}}. \tag{28}$$

Within the recursive formulation, the output sample at the k-th time instant is predicted as

$$\hat{y}_k(\boldsymbol{\theta}_*) = \sum_{j=1}^{N_s} \beta_j k_{\xi}(\boldsymbol{\xi}_{s_j}, \hat{\boldsymbol{\xi}}_k) k_{\theta}(\boldsymbol{\theta}_{s_j}, \boldsymbol{\theta}_*). \tag{29}$$

The recursive evaluation requires an initialization of the input and output histories appearing in (28). In the results presented in this work, the first r available samples are used to initialize these histories. We therefore define the initialization lag as $lag = r$ samples, so that the first free-running prediction is obtained at sample $k = lag + 1$. Since $k = 1$ corresponds to $t = 0$, the predicted waveform starts at $t = lag\, T_s = rT_s$. Alternatively, prediction can start directly from $t = 0$ by prescribing the input and output histories preceding the first sample. In particular, for a zero-state response, these past samples can be initialized to zero.

The deterministic KRR formulation adopted here is closely related to Gaussian Process Regression (GPR). For a fixed kernel and corresponding regularization/noise parameters, the KRR predictor coincides with the posterior mean of the corresponding GPR model [39]–[43], while GPR additionally provides a posterior covariance. Extending the present formulation to probabilistic prediction is, however, nontrivial in the free-running NARX setting, since the uncertainty associated with previously predicted outputs would have to be propagated through the subsequent recursive regressors. Moreover, the landmark-selection strategy should then account for both posterior-mean accuracy and predictive uncertainty.

More generally, the dynamic kernel representation introduced above is not intrinsically restricted to KRR, and alternative kernel regression methods can in principle be employed within the same NARX framework. Compactness can

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

also be achieved through alternative sparsification mechanisms, including relevance-vector selection [44] and inducing-point sparse Gaussian Processes, which are closely related to Nyström kernel approximations [45]. A systematic extension to probabilistic prediction and comparison with alternative sparse-kernel formulations in the large-scale recurrent setting are left for future investigation.

III. Results

This section assesses the performance and generality of the proposed parametric identification framework through different application scenarios. The selected examples cover both device-level and system-level modeling problems and involve different numbers and types of parameters, input excitations, and output quantities.

The first example addresses a device-level modeling problem involving an IC buffer. In this case, the proposed approach is used to model the output current as a function of the input bit string, output voltage, and temperature, thereby enabling the analysis of the device behavior under different operating and load conditions. The second example addresses the parametric input-output modeling of a linear high-speed interconnect, where the model describes the relation between the input voltage excitation and the far-end output voltage as a function of two geometrical parameters of the link. Finally, the third example deals with a more realistic nonlinear high-speed interconnect with IBIS-based driver and receiver terminations. In this case, the objective is to model the relation between the input voltage excitation and the far-end voltage as a function of five link parameters. For the last application, additional analyses of the scalability with respect to the parameter-space dimensionality and the sensitivity to noise in the training data are presented in Appendix B.

For all the considered applications, the training and test waveforms are generated by transient circuit simulations carried out with HSPICE U-2023.03-SP1. HSPICE transient simulations use adaptive time stepping and therefore provide waveforms on nonuniform time grids. Before constructing the NARX datasets, the simulated responses are interpolated onto the uniform time grids specified for each application. The kernel models are trained in MATLAB R2024b on a workstation equipped with an Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz and 16 GB of RAM. The same workstation is used to run the HSPICE simulations. The kernel hyperparameters and the regularization parameter are selected using a validation set obtained by randomly extracting 40% of the available training samples. The hyperparameter search is carried out by Bayesian optimization, using the mean squared error computed on the validation set as the objective function.

For each application, the HSPICE cost required to generate the training data is reported separately from the MATLAB model-training time, together with a comparison between the evaluation time of the trained model and the corresponding HSPICE transient simulation. This comparison should be interpreted considering the different purposes of the two approaches. HSPICE performs a complete circuit-level solution and inherently provides multiple circuit quantities, whereas the present behavioral formulation predicts one selected scalar output. Nevertheless, the proposed approach learns the selected parametric nonlinear dynamic relation directly from input-output data without requiring explicit knowledge of the circuit topology or component models; once trained, its evaluation cost is therefore not tied to the complexity of the original circuit description, and the offline data-generation and identification costs can be amortized over repeated evaluations for new input excitations and parameter configurations.

A. Application 1: Parametric IC Buffer Modeling

The first application focuses on the transient modeling of the output current of the commercial Texas Instruments transceiver SN74ALVCH16973. The device operates at a nominal supply voltage $V_{DD} = 1.8\,\mathrm{V}$. The reference data are obtained from the transistor-level HSPICE model provided by the manufacturer.

Similarly to [15], the operating temperature is treated as an external parameter, i.e., $p = 1$, so that the parameter vector reduces to the scalar quantity θ. The objective is to construct a parametric dynamic model able to predict the output current $i_o(t;\theta)$ for unseen temperature values, bit sequences, and load conditions, using the input excitation and the output voltage across the load as input variables of the dynamic regressor. This example, previously considered in [15], is used here to demonstrate the benefits of the improved formulation in terms of accuracy and compactness.

The training data are generated for $Q = 6$ uniformly spaced temperature values in $[0, 70]^{\circ}\mathrm{C}$. For each temperature, two input bit strings, namely "01010001100" and "01000010100", are considered, with a $40\,\mathrm{ns}$ bit period and $1\,\mathrm{ns}$ rise/fall time. Four load configurations are used: i) an open-ended transmission line with $60\ \Omega$ characteristic impedance and 10 ns delay; ii) an open-ended transmission line with $75\ \Omega$ characteristic impedance and 4 ns delay; iii) a lumped $50\ \Omega$ resistor connected in parallel with a 10 pF capacitor; and iv) a lumped $150\ \Omega$ resistor. Therefore, the overall number of operating conditions, intended as combinations of input excitation and load configuration, is $M = 2 \times 4 = 8$.

Each transient simulation is performed over the interval $[0, 440]$ ns. The waveform duration is selected so that the transient response has decayed to zero before the end of each simulation, thereby avoiding artificial discontinuities when consecutive training responses are concatenated. The simulated waveforms are uniformly resampled with sampling period $T_s = 285\,\mathrm{ps}$, leading to $L = 1541$ discrete-time samples.

As in [15], the dynamic orders $r_u = r_y = 4$ are used for the input and output regressors. The resulting dataset contains $QML = 6 \times 8 \times 1541 = 73968$ samples, corresponding to $N_{\mathrm{tr}} = QML - r = 73964$ regression samples after accounting for the dynamic order. It is worth noting that these 73964 regression samples are generated from only $QM = 48$ independent HSPICE transient simulations, since each simulation contributes a complete time sequence to the dynamic regression dataset. The HSPICE simulations required to generate the complete training dataset take approximately $203\,\mathrm{s}$. The training set is used to train four models: a plain KRR model,

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

corresponding to the approach presented in [15], and three Nyström KRR models based on the proposed formulation in Section II-C, with increasing numbers of landmarks N_s. All models use isotropic Matérn $5/2$ kernels for both $k_{\boldsymbol{\xi}}$ and k_θ. The explicit kernel expressions are provided in Appendix A.

The test set consists of unseen waveforms generated with a new load configuration, namely a transmission line with $50\ \Omega$ characteristic impedance, a 3 ns delay, and a 5 pF termination capacitance. Three unseen input bit streams, namely "0101000000", "0110100000", and "0011000000", are simulated at 15 uniformly spaced temperature values in $[0, 70]^\circ$C. During prediction, the dynamic regressor is updated recursively using the available input samples and the previously predicted output-current samples, so that the model is evaluated in free-run mode.

Table I compares the considered models in terms of the number of coefficients, training time, evaluation time, and relative L_2-error. The plain KRR model requires one coefficient per training sample, namely $N_{\mathrm{tr}} = 73964$ coefficients in this example. This results in a large training cost and may lead to numerical conditioning issues due to the strong correlation among neighboring time samples, as discussed in Section II-B. As a consequence, the plain KRR model requires a much larger training time and achieves a relative L_2-error of 15.3% on the test set. In contrast, the Nyström approximation considerably reduces the number of coefficients and training time while providing improved prediction accuracy, with the $N_s = 400$ model achieving a relative L_2-error of 3.1%. The trained compact models require substantially less time to evaluate a new test case than a full HSPICE transient simulation, demonstrating the computational benefit of replacing repeated circuit simulations with the proposed behavioral model.

On the other hand, the proposed Nyström-compressed formulation replaces the full expansion over all training samples with an expansion over $N_s \ll N_{\mathrm{tr}}$ landmark samples, while still exploiting all available training data through (25). This significantly reduces the number of model coefficients and the training time. In particular, the training time ranges from 66 s for $N_s = 200$ to 100 s for $N_s = 400$. The test error decreases as the number of landmarks increases, showing that the proposed formulation improves its accuracy as the number of model coefficients grows. The best performance is obtained with $N_s = 400$, yielding a relative L_2-error of 3.1% with a training time of 100 s. This represents a clear improvement over the plain KRR model, both in terms of accuracy and computational cost.

For each test waveform containing 1541 time samples, the evaluation of the trained model with $N_s = 400$ landmarks requires 0.19 s, compared with 4.15 s for the corresponding HSPICE transient simulation. It is important to remark that this application was designed primarily to demonstrate the modeling capabilities of the proposed framework rather than its computational efficiency.

Figure 1 shows a parametric comparison between the model predictions obtained with $N_s = 400$ and the reference output-current waveforms $i_o(t;\theta_*)$ for the input bit stream "0110100000" at $\theta_* = 25^\circ$C, 45°C, and 65°C. The upper panels report the full transient responses and a magnified view of a representative time window, highlighting the influence of temperature on the output-current dynamics and the close agreement between reference and predicted responses. The bottom panel reports the absolute prediction error, which confirms that the largest discrepancies are localized around the switching transients.

TABLE I
PERFORMANCE COMPARISON FOR THE IC BUFFER EXAMPLE IN SEC. III-A.

Model	Coeff.	Model		rel. L_2-err. (Waveform)
		Training time	Evaluation time/test case	
Plain KRR	73964	2 h	9.03 s	15.3%
Nyström KRR	$N_s = 200$	66 s	0.17 s	13.8%
	$N_s = 300$	91 s	0.18 s	11.7%
	$N_s = 400$	100 s	0.19 s	3.1%
HSPICE data gen. time: 203 s — Test HSPICE time/test case: 4.15 s				

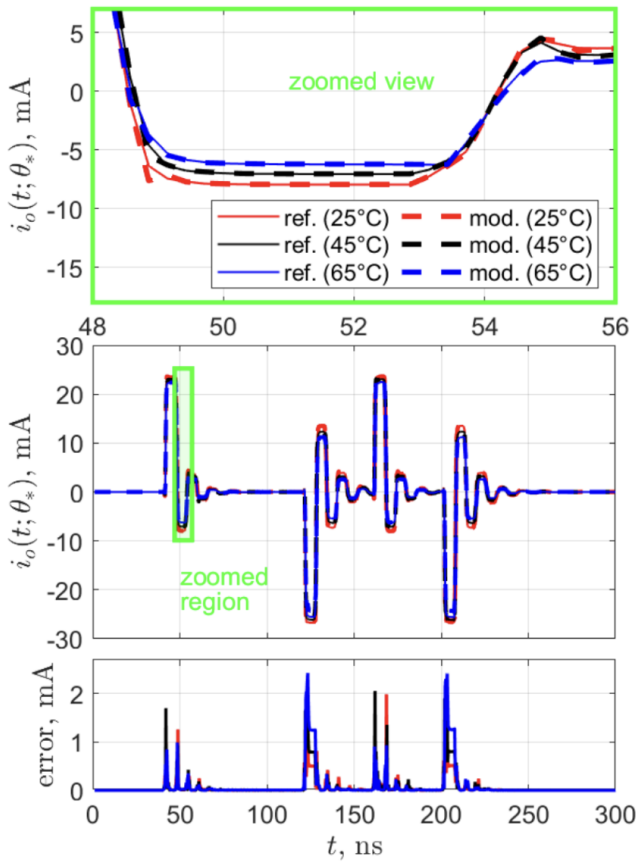


Fig. 1. Application 1: validation results for the unseen test load configuration. Comparison between the reference output current $i_o(t;\theta_*)$ and the model predictions at $\theta_* = 25^\circ$C, 45°C, and 65°C. The two upper panels show the complete output-current response and a magnified view of a representative time window. The bottom panel shows the absolute prediction error.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

TABLE II
NOMINAL PARAMETERS OF THE TRANSMISSION-LINE CROSS SECTION.

Parameter	Nominal value	Parameter	Nominal value
g	150 μm	h	200 μm
d_1	100 μm	d_2	140 μm
d_3	70 μm	w_1	150 μm
w_2	130 μm	w_3	170 μm
t_1	30 μm	t_2	20 μm
t_3	40 μm	$\tan\delta$	0.02
σ	58×10^6 S/m	–	–

B. Application 2: Parametric Modeling of a Linear High-Speed Link

The second application considers the parametric input-output modeling of a high-speed interconnect with linear terminations. The objective is to predict the far-end voltage response of the link from the input voltage excitation, corresponding to the bit stream feeding the interconnect, while accounting for the dependence on two geometrical parameters. The predicted far-end responses are then used to generate the corresponding eye diagrams.

The considered structure, shown in Fig. 2, is adapted from [17], [18] and consists of an interconnection of distributed transmission-line sections. In this application, the drivers and receivers in the schematic of Fig. 2 are replaced by linear terminations. Specifically, drivers $D1$ and $D3$ are modeled by high-impedance terminations through 1 kΩ resistors, whereas driver $D2$ is modeled by a Thevenin equivalent circuit composed of the voltage source $v_s(t)$ connected in series with a 30 Ω resistor. All receivers at the far end of the interconnect are represented by 1 pF capacitive loads. The reference data are generated by HSPICE transient simulations, and the link is excited by an idealized transmitter with supply voltage $V_{DD} = 1.2$ V.

The nominal geometrical values defining the transmission-line cross section are reported in Table II. In this application, the varied parameters are the gap g and the height h. Therefore, the parameter vector is $\boldsymbol{\theta} = [\theta_1, \theta_2]^\mathsf{T}$, with $p = 2$, where $\theta_1 = g$ and $\theta_2 = h$. Both parameters are varied by $\pm 40\%$ around their nominal values. The target output is the far-end voltage $v_3(t;\boldsymbol{\theta})$, measured at the node located at the end of the upper lane of the second transmission-line structure, while the input is the voltage $v_s(t)$.

The training data are generated by considering $M = 10$ different 7-bit excitation sequences, including for example "0110110", "0010100", and "0101010". Each excitation waveform starts with a zero bit and has a rise/fall time of 0.1 ns and a bit rate of 1 Gbit/s. For each excitation sequence, $Q = 100$ parameter configurations are simulated, resulting in $MQ = 1000$ transient simulations. The parameter samples are generated as $\theta_1 = 150(1 + 0.4x_1)$ μm and $\theta_2 = 200(1 + 0.4x_2)$ μm, where $x_1, x_2 \in [-1, 1]$ are uniformly distributed random variables.

Each transient simulation is performed up to 20 ns. The resulting waveforms are uniformly sampled with sampling period $T_s = 50$ ps, leading to $L = 401$ discrete-time samples per waveform. The 1000 HSPICE transient simulations required to generate the complete training dataset take approximately 4580 s.

Due to the propagation delay introduced by the distributed structures of the link, the dynamic orders are selected as $r_u = 80$ and $r_y = 20$, with $r = 80$. The resulting number of training samples is $N_{\mathrm{tr}} = 1000 \times 401 - 80 = 400920$.

This dataset size makes plain KRR training impractical for repeated hyperparameter tuning, since it would require forming and solving a regression problem involving a 400920×400920 Gram matrix. Therefore, the plain KRR model is not considered in this example, which is instead used to assess the benefits of the proposed Nyström-compressed formulation.

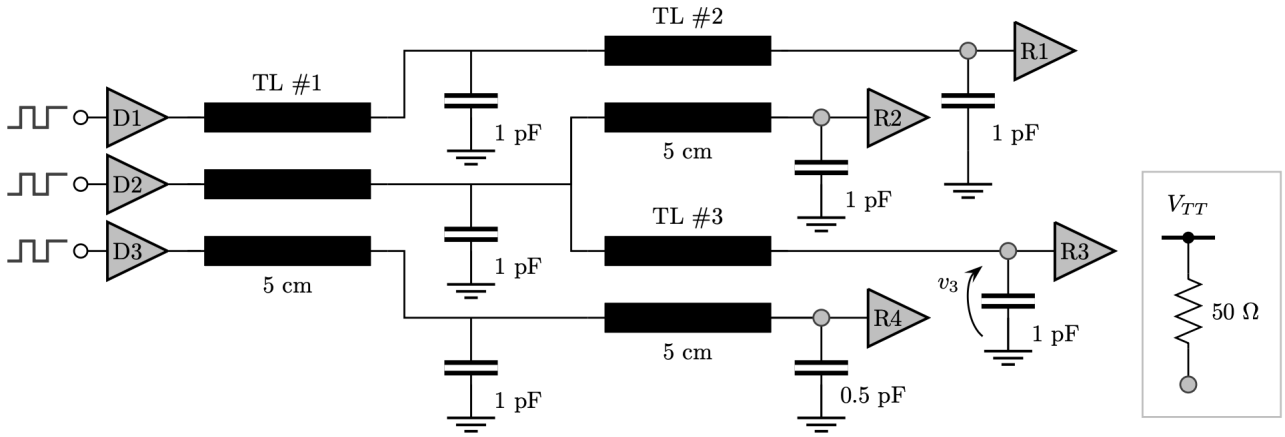
The compressed scalar KRR model is built using the separable kernel defined in (12). An isotropic Matérn $5/2$ kernel is used for $k_{\boldsymbol{\xi}}$, whereas an anisotropic Matérn $5/2$ kernel is used for $k_{\boldsymbol{\theta}}$, as described in Appendix A. The isotropic structure assigns a common length-scale kernel hyperparameter to all 101 components of the relatively high-dimensional NARX regressor, thereby limiting the number of hyperparameters to be optimized. In contrast, the anisotropic kernel assigns a different length-scale hyperparameter to each physical parameter, allowing the model to account for different sensitivities of the response to the individual geometrical parameters. The Matérn $5/2$ kernel provides a compromise between smoothness and modeling flexibility; unlike the squared-exponential/RBF kernel, the Matérn family does not impose infinite differentiability of the underlying function [28], [40]. Three compressed models are trained using $N_s = 100$, 300, and 500 landmarks.

The performance of the obtained models is assessed under unseen operating conditions using a 512-bit pseudo-random bit sequence (PRBS) with $V_{DD} = 1.2$ V, rise/fall time equal to 0.1 ns, and bit rate equal to 1 Gbit/s. This excitation is simulated for 100 newly generated parameter configurations, obtained with the same nominal values and variability range used for training. The resulting transient responses, with total duration of 500 ns, are used to construct 100 reference eye diagrams.

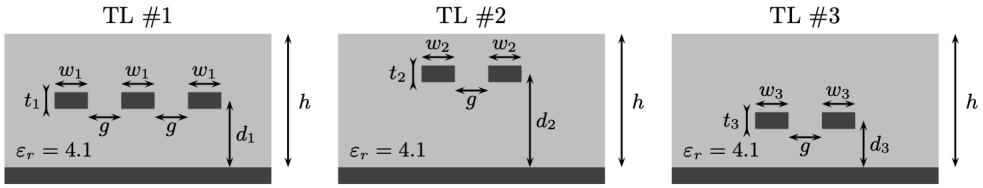
Table III reports the quantitative results for the long-sequence validation in terms of the relative L_2-error of the transient waveforms and the eye-diagram features, namely eye height and eye area. Increasing the number of landmarks improves the eye-diagram accuracy while keeping the training time within a few minutes. The table also reports the model evaluation time per test case, with each test case corresponding to one excitation and one parameter configuration. For comparison, the last row reports the total HSPICE time required to generate the training dataset and the HSPICE simulation time per test case. For the 512-bit validation sequence, the model with $N_s = 500$ landmarks requires 1.91 s per test case, compared with 25.68 s for the corresponding HSPICE transient simulation, corresponding to a speed-up of approximately $13.4\times$. It is worth noting that this speed-up refers only to the model evaluation stage and does not account for the offline costs associated with training-data generation and model training. However, these costs are incurred only once; after training, the model can be repeatedly evaluated for arbitrary input bit strings and parameter configurations within

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459



(a) high-speed interconnect



(b) transmission line cross-sections

Fig. 2. Schematic of the high-speed link considered in the second and third applications (top panel) and transmission-line cross sections with the involved geometrical parameters (bottom panel). The additional pull-up termination shown in the box on the right is connected to the far-end gray nodes at the receiver side for the third application only.

TABLE III
PERFORMANCE COMPARISON FOR THE LINEAR HIGH-SPEED LINK EXAMPLE IN SEC. III-B.

N_s	Model		Rel. L_2-error		
	Training time	Evaluation time/test case	Waveform	Eye height	Eye area
100	96 s	1.09 s	4.9%	10.1%	5.1%
300	242 s	1.47 s	3.3%	7.1%	3.7%
500	387 s	1.91 s	3.5%	6.4%	2.9%
HSPICE data gen. time: 4580 s — Test HSPICE time/test case: 25.68 s					

the considered parameter domain.

Figure 3 compares the reference eye diagram obtained from HSPICE with the corresponding eye diagram predicted by the model trained with $N_s = 500$ landmarks for a representative unseen parameter configuration. The agreement confirms that the Nyström-compressed model accurately reproduces the eye-diagram behavior while using only a small number of model coefficients.

Figure 4 compares the corresponding transient far-end voltage waveforms in the time window $[0, 40]$ ns. The top panel reports the input bit stream. The plot highlights the propagation delay introduced by the distributed structures of the link and shows that the prediction starts after $r = 80$ samples, corresponding to the dynamic memory used by the model. The comparison confirms the capability of the proposed model to reproduce the dynamic behavior of the output voltage.

C. Application 3: Parametric Modeling of a Nonlinear High-Speed Link

The third application considers a more realistic high-speed interconnect based on the same structure used in the example before and shown in Fig. 2. Unlike the previous linear case, the drivers and receivers are now represented by nonlinear IBIS models, and the bit rate is increased. This example therefore extends the validation to a nonlinear link with realistic I/O-buffer behavior and five external geometrical parameters. The objective is again to predict the far-end voltage response from the input voltage excitation and to use the predicted waveforms to generate the corresponding eye diagrams.

IBIS models supporting the SSTL-3 Class II I/O standard are employed for the three drivers and four receivers shown in Fig. 2. The models are taken from the Intel Cyclone V device specifications provided in the vendor documentation. The drivers and receivers are powered by a common supply voltage $V_{DD} = 3.3$ V. The input bit-stream excitation is applied through the voltage source $v_s(t)$ at the driver input. According to the SSTL-3 Class II standard, the driver output impedance is combined with a $25\ \Omega$ resistor, while the receiver side is connected to a series connection of a $V_{TT} = 1.5$ V dc source and a $50\ \Omega$ resistor, as shown in the box on the right-hand side of Fig. 2. The reference data are generated by HSPICE transient simulations.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

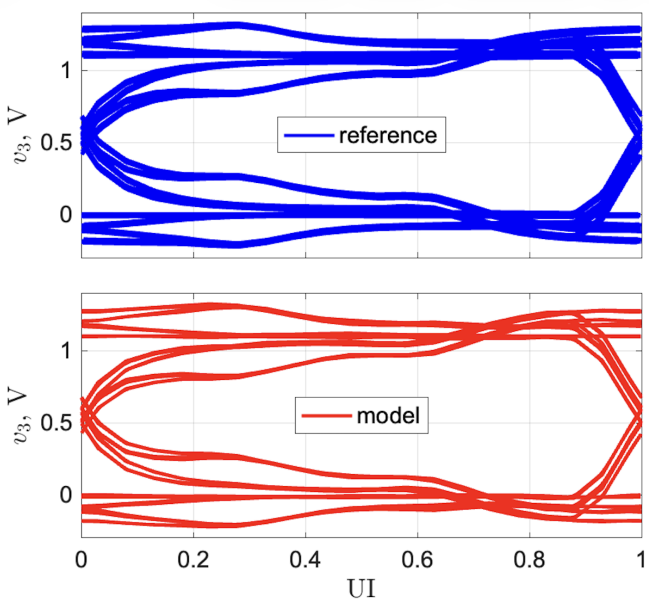


Fig. 3. Application 2: comparison of the reference eye diagram obtained from HSPICE with the prediction of the proposed Nyström KRR model, trained using $N_s = 500$ landmarks, for a representative unseen test configuration.

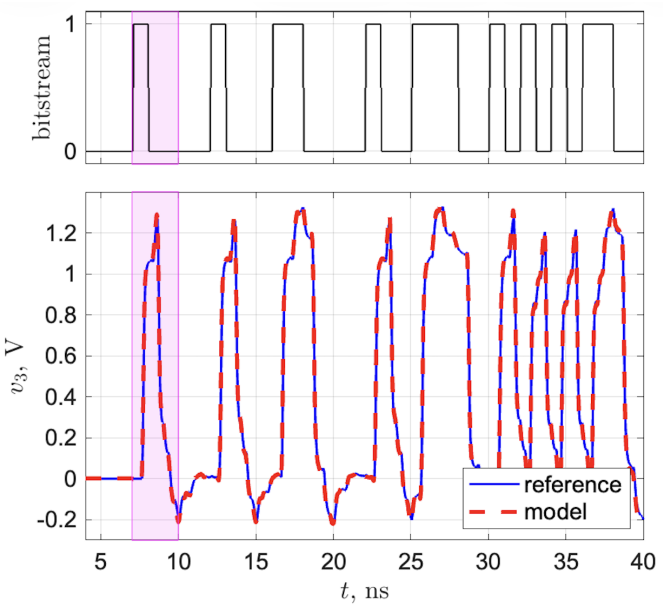


Fig. 4. Application 2: comparison between the reference and predicted far-end voltage waveforms obtained with the proposed model trained using $N_s = 500$ landmarks. The time window $[0, 40]$ ns is shown for readability. The top panel reports the input bit stream.

The target output is the far-end voltage $v_3(t;\boldsymbol{\theta})$, measured at the node where the upper lane of the second transmission-line structure is connected to the third receiver buffer $R3$. In this example, the parameter space is extended from the two-parameter testcase of Sec. III-B to a five-parameter case. The first two parameters are the gap g and the height h, while the remaining three parameters are the distances d_1, d_2, and d_3 shown in Fig. 2. Therefore, the external parameter vector is defined as $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]^\mathsf{T}$, with $p = 5$, where $\theta_1 = g$, $\theta_2 = h$, $\theta_3 = d_1$, $\theta_4 = d_2$, and $\theta_5 = d_3$. The gap and height are varied by $\pm 40\%$ around their nominal values, as in the previous application, while d_1, d_2, and d_3 are varied by $\pm 20\%$ around their nominal values reported in Table II.

The training data are generated by considering $M = 10$ different 7-bit excitation sequences. Each excitation waveform has a rise/fall time of 0.03 ns and a bit rate of 3 Gbit/s. For each excitation sequence, $Q = 100$ parameter configurations are simulated, resulting in $MQ = 1000$ transient simulations. The parameter samples are generated as $\theta_1 = 150(1 + 0.4x_1)$ μm, $\theta_2 = 200(1+0.4x_2)$ μm, $\theta_3 = 100(1+0.2x_3)$ μm, $\theta_4 = 140(1 + 0.2x_4)$ μm, and $\theta_5 = 70(1 + 0.2x_5)$ μm, where $x_1, \ldots, x_5 \in [-1, 1]$ are uniformly distributed random variables.

Each transient simulation is performed over a time interval of 6 ns. The resulting waveforms are uniformly sampled with sampling period $T_s = 37$ ps, leading to $L = 163$ discrete-time samples per waveform. Due to the propagation delays and the nonlinear behavior introduced by the IBIS terminations, the model orders are set to $r_u = 90$ and $r_y = 60$, with $r = 90$. These values account for the higher bit rate, the different sampling period, and the different termination characteristics with respect to the previous application. The resulting number of training samples is $N_{\text{tr}} = 1000 \times 163 - 90 = 162910$. The 1000 HSPICE transient simulations required to generate the complete training dataset take approximately 4726 s.

Also in this case, the use of plain KRR is impractical because it would require constructing and solving a regression problem involving a 162910×162910 Gram matrix. The compressed scalar KRR model is built using an isotropic Matérn $5/2$ kernel for $k_{\boldsymbol{\xi}}$, and an anisotropic Matérn $5/2$ kernel for $k_{\boldsymbol{\theta}}$. The same rationale discussed in Sec. III-B is adopted here: a common length scale is used in the high-dimensional dynamic-regressor space, whereas separate length scales are retained for the physical parameters. Three compressed models are trained using $N_s = 100$, 500, and 900 landmarks.

To validate the obtained models, a long-sequence eye-diagram validation is performed. A 1024-bit pseudo-random bit sequence is generated with $V_{DD} = 3.3$ V, rise/fall time equal to 0.03 ns, and bit rate equal to 3 Gbit/s. This excitation is simulated for 100 newly generated parameter configurations, obtained with the same nominal values and variability ranges used for training. The resulting transient responses, with total duration of approximately 400 ns, are used to assess the prediction accuracy and to construct 100 reference eye diagrams under unseen operating conditions.

Table IV reports the quantitative results for the long-sequence validation in terms of the relative L_2-error of the transient waveforms and the eye-diagram features, namely

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

TABLE IV
PERFORMANCE COMPARISON FOR THE HIGH-SPEED LINK EXAMPLE WITH NON-LINEAR TERMINATIONS IN SEC. III-C.

N_s	Model		Rel. L_2-error		
	Training time	**Evaluation time/test case**	**Waveform**	**Eye height**	**Eye area**
100	51 s	1.32 s	4.8%	11.5%	10.4%
500	200 s	2.08 s	4.8%	9.6%	11.9%
900	454 s	5.61 s	3.0%	4.5%	4.5%
HSPICE data gen. time: 4726 s — Test HSPICE time/test case: 29.07 s					

eye height and eye area. Increasing the number of landmarks improves the overall prediction accuracy while keeping the training time within a few minutes. The table also reports the model evaluation time per test case, with each test case corresponding to one excitation and one parameter configuration. For comparison, the last row reports the total HSPICE time required to generate the training dataset and the HSPICE simulation time per test case. For the 1024-bit validation sequence at a fixed parameter configuration, evaluation of the trained model with $N_s = 900$ landmarks requires $5.61\,\mathrm{s}$, compared with $29.07\,\mathrm{s}$ for the corresponding HSPICE transient simulation. In particular, the model with $N_s = 900$ landmarks achieves the best accuracy, with a waveform error of 3.0% and eye-height and eye-area errors of 4.5%.

The model evaluation time increases from $1.32\,\mathrm{s}$ for $N_s = 100$ to $2.08\,\mathrm{s}$ and $5.61\,\mathrm{s}$ for $N_s = 500$ and 900, respectively, reflecting the expected increase in computational cost with the number of landmarks. Even for the most computationally demanding case, corresponding to $N_s = 900$, the model provides a speed-up of approximately $5\times$ with respect to the corresponding HSPICE transient simulation. As in the previous application, this speed-up refers only to the model evaluation stage and does not account for the offline costs associated with training-data generation and model training. However, these costs are incurred only once; after training, the model can be repeatedly evaluated for arbitrary input bit strings and parameter configurations within the considered parameter domain.

Figure 5 compares the reference eye diagram with the corresponding prediction obtained using the model trained with $N_s = 900$ landmarks for a representative unseen parameter configuration. Despite the nonlinear behavior introduced by the IBIS driver and receiver models and the presence of five varying geometrical parameters, the predicted eye diagram is in good agreement with the reference result.

Similar to the previous example, Fig. 6 compares the reference and predicted far-end voltage waveforms over the time window $[0, 20]$ ns. For this application, $lag = r = 90$ samples and $T_s = 37$ ps. Therefore, the free-running prediction shown in Fig. 6 starts at $t = lag\, T_s \simeq 3.33$ ns.

Two additional studies complement the present application by assessing: (i) the scalability of the proposed formulation with respect to the number of varying physical parameters, and (ii) its sensitivity to noisy training data. Their detailed setup and results are reported in Appendix B.

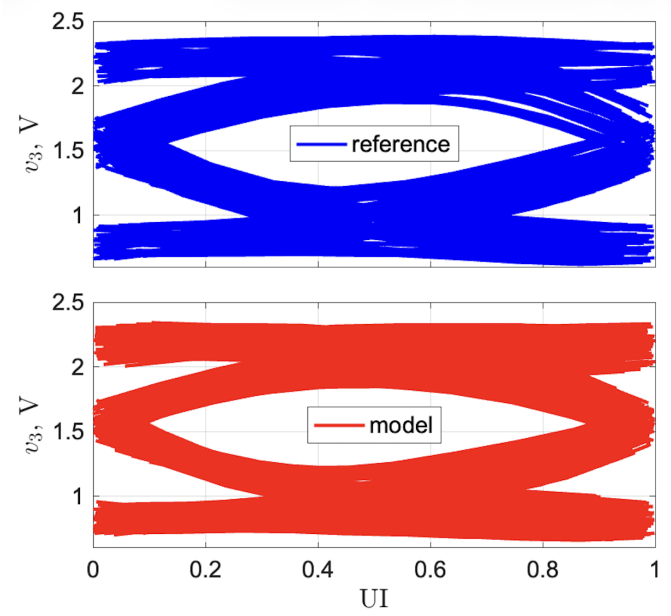


Fig. 5. Application 3: comparison of the reference eye diagram obtained from HSPICE with the prediction of the proposed Nyström KRR model, trained using $N_s = 900$ landmarks, for a representative unseen test configuration.

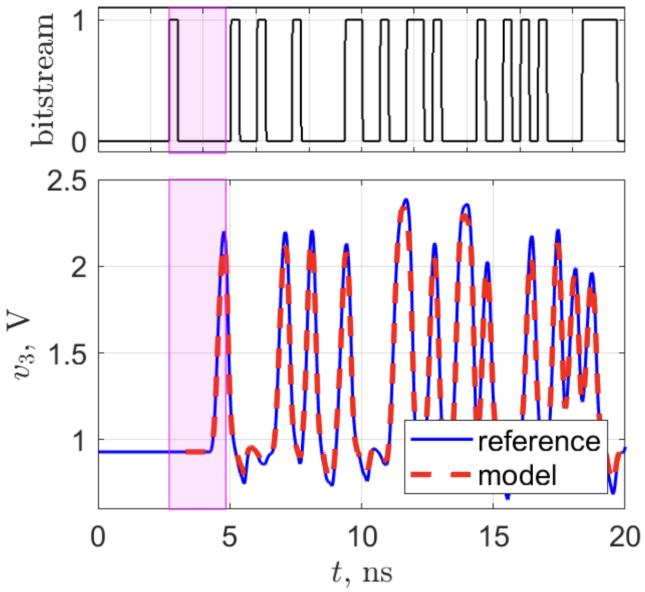


Fig. 6. Application 3: comparison between the reference and predicted far-end voltage waveforms obtained with the proposed model trained using $N_s = 900$ landmarks. The time window $[0, 20]$ ns is shown for readability.

IV. CONCLUSIONS

This paper presented a compact parametric dynamic modeling framework for high-speed electronic systems based on Nyström-compressed Kernel Ridge Regression. The proposed approach extends the preliminary formulation in [15] by introducing independent input and output memory orders and by enabling the use of large transient datasets while keeping

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

the number of model coefficients limited.

The method was validated on three examples of increasing complexity, ranging from device-level IC-buffer modeling to system-level prediction in high-speed links with linear and nonlinear terminations. The results show that the compressed formulation provides accurate transient and eye-diagram predictions with significantly reduced training time and model complexity compared with the plain KRR formulation. The additional analyses also demonstrate the applicability of the approach to higher-dimensional parameter spaces and its robustness to moderate levels of noise in the training data.

Overall, the proposed formulation provides a flexible data-driven approach for constructing compact parametric dynamic models directly from transient input-output data. Future developments will focus on extending the framework to probabilistic and multi-output dynamic modeling.

APPENDIX A
MATÉRN–5/2 KERNEL FORMULAS

The isotropic Matérn $5/2$ kernel adopted in this paper is defined as

$$k_{\boldsymbol{\xi}/\boldsymbol{\theta}}(\mathbf{r},\mathbf{r}';\sigma_{\boldsymbol{\xi}/\boldsymbol{\theta}}) = \left(1+\sqrt{5}d+\frac{5}{3}d^2\right)e^{-\sqrt{5}d}, \tag{30}$$

where $d = \|\mathbf{r}-\mathbf{r}'\|/\sigma_{\boldsymbol{\xi}/\boldsymbol{\theta}}$. The vector $\mathbf{r}$ denotes either the input regressor $\boldsymbol{\xi}$, for $k_{\boldsymbol{\xi}}$, or the parameter vector $\boldsymbol{\theta}$, for $k_{\boldsymbol{\theta}}$. The quantity $\sigma_{\boldsymbol{\xi}/\boldsymbol{\theta}}$ represents the kernel hyperparameter optimized during model training within the interval $[10^{-2}, 10^{4}]$.

The anisotropic Matérn $5/2$ kernel adopted in this paper is defined as

$$k_{\boldsymbol{\xi}/\boldsymbol{\theta}}(\mathbf{r},\mathbf{r}';\boldsymbol{\Lambda}_{\boldsymbol{\xi}/\boldsymbol{\theta}}) = \left(1+\sqrt{5}d+\frac{5}{3}d^2\right)e^{-\sqrt{5}d}, \tag{31}$$

where $d = \sqrt{(\mathbf{r}-\mathbf{r}')^{\mathsf{T}}\boldsymbol{\Lambda}_{\boldsymbol{\xi}/\boldsymbol{\theta}}^{-2}(\mathbf{r}-\mathbf{r}')}$ denotes the anisotropic ellipsoidal distance metric. The vector $\mathbf{r}$ represents either the input regressor $\boldsymbol{\xi}$, for $k_{\boldsymbol{\xi}}$, or the parameter vector $\boldsymbol{\theta}$, for $k_{\boldsymbol{\theta}}$. Here, $\boldsymbol{\Lambda}_{\boldsymbol{\xi}/\boldsymbol{\theta}}$ is a diagonal matrix whose entries are optimized during model training within the interval $[10^{-1}, 10^{4}]$.

APPENDIX B
ADDITIONAL SCALABILITY AND NOISE-SENSITIVITY ANALYSES

This Appendix complements the nonlinear high-speed link application described in Sec. III-C with two additional analyses assessing the scalability and robustness of the proposed modeling framework.

A. Scalability with the Number of Physical Parameters

To investigate the effect of parameter-space dimensionality while keeping the physical test case unchanged, three configurations with 2, 5, and 10 varying geometrical parameters are considered. Specifically:

- **2 parameters:** g and h, both varied by $\pm 40\%$ around their nominal values;
- **5 parameters:** g and h varied by $\pm 40\%$, and d_1, d_2, and d_3 varied by $\pm 20\%$, corresponding to the original configuration of Application 3;
- **10 parameters:** g and h varied by $\pm 40\%$, and d_1, d_2, d_3, t_1, t_2, t_3, w_2, and w_3 varied by $\pm 20\%$.

TABLE V
SCALABILITY ANALYSIS FOR APPLICATION 3 AS A FUNCTION OF THE NUMBER OF VARYING PHYSICAL PARAMETERS AND NYSTRÖM LANDMARKS.

No. of parameters	N_s	Waveform rel. L_2 error	Training time (s)
2	100	4.6%	53
	500	3.2%	178
	900	2.4%	318
5	100	4.8%	51
	500	4.8%	200
	900	3.0%	454
10	100	6.5%	57
	500	5.6%	165
	900	5.2%	341

For each configuration, models with $N_s = 100$, 500, and 900 landmarks are trained and evaluated. The same long-sequence validation procedure adopted in Application 3 is used, considering a 1024-bit pseudo-random bit sequence and 100 unseen parameter configurations generated within the corresponding parameter domains. The resulting transient responses are used to assess the waveform prediction accuracy. The results are summarized in Table V.

The results show that, for the considered test case, increasing the number of varying parameters generally requires a richer landmark representation to retain comparable prediction accuracy. This behavior should not be interpreted as a direct dependence of the dominant Nyström computational cost on the number of physical parameters, since this cost is primarily governed by the number of training samples and landmarks. This is also reflected in the training times, which do not increase monotonically with parameter-space dimensionality. When an anisotropic/ARD kernel is adopted, increasing the number of physical parameters also increases the number of kernel hyperparameters to be optimized.

The need for additional landmarks is instead primarily related to the complexity of the parametric dynamic input-output map to be learned. Parameter-space dimensionality alone does not determine this complexity: the considered variability ranges and the underlying system behavior also play an important role. For example, increasing the variability range of a given set of parameters may expose a larger and potentially more complex portion of the parametric map, whereas a larger number of parameters with limited variability may still be represented accurately with a comparatively small landmark set. Therefore, the number of landmarks required to achieve a prescribed accuracy is ultimately governed by the complexity of the parametric map rather than by parameter-space dimensionality alone.

B. Sensitivity to Noisy Training Data

The sensitivity of the proposed formulation to noisy training data is investigated by corrupting the output samples used for model identification according to

$$\widetilde{y}_n = y_n + \varepsilon_n, \tag{32}$$

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

TABLE VI
NOISE-SENSITIVITY ANALYSIS FOR APPLICATION 3 WITH $N_s = 500$. THE NOISE LEVEL IS DEFINED AS $\gamma = \sigma_\varepsilon/\sigma_y$.

Noise level	Waveform rel. L_2 error	Eye-height rel. error	Eye-area rel. error	Training time (s)
0%	4.8%	9.6%	11.9%	200
1%	5.0%	11.5%	12.7%	207
5%	5.6%	14.1%	16.3%	211
10%	5.7%	15.3%	17.4%	193
20%	7.1%	25.3%	26.5%	274

where $\varepsilon_n \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is a zero-mean Gaussian random variable representing the additive noise, with standard deviation

$$\sigma_\varepsilon = \gamma \sigma_y, \tag{33}$$

where σ_y denotes the standard deviation of the clean training-output data and γ controls the noise level. The cases $\gamma = 0$, 0.01, 0.05, 0.10, and 0.20 are considered, where $\gamma = 0$ corresponds to the original noise-free training dataset. The two highest noise levels are included to stress-test the proposed approach under severe corruption of the training data. The test waveforms are kept noise-free and used as reference, while the number of landmarks is fixed to $N_s = 500$ in all cases to isolate the effect of training-data noise. The resulting performance is reported in Table VI.

The results show a gradual degradation of prediction accuracy as the noise level increases. The waveform relative L_2 error increases from 4.8% for the noise-free training data to 5.7% at a 10% noise level, while a more pronounced degradation is observed at 20%, where the error reaches 7.1%. The eye-diagram metrics exhibit a stronger sensitivity to training-data corruption, particularly at the highest noise levels. Nevertheless, the proposed formulation retains good waveform prediction accuracy for the considered low-to-moderate noise levels.

Measurement noise affects both the NARX regressors, through the past output samples used to construct them, and the distance-based Nyström landmark-selection procedure. Although the present work primarily considers training data generated by circuit simulations, the results obtained for low-to-moderate noise levels suggest the potential applicability of the proposed framework to training datasets including experimental measurements. If required, additional noise-robust identification and preprocessing strategies could be incorporated when dealing with noisy measured data.

REFERENCES

[1] K. T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments*. London, U.K.: Taylor & Francis Group, 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] G. Signorini, C. Siviero, M. Telescu, and I. S. Stievano, "Present and future of I/O-buffer behavioral macromodels," *IEEE Electromagnetic Compatibility Magazine*, vol. 5, no. 3, pp. 79–85, 2016.

[4] 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.

[5] I. S. Stievano, F. G. Canavero, and 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.

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

[7] 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. no. 831, 2020.

[8] Y. Cao, R. Ding, and Q.-J. Zhang, "State-space dynamic neural network technique for high-speed IC applications: modeling and stability analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 6, pp. 2398–2409, Jun. 2006.

[9] 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, 2020.

[10] T. Chen, H. Andersen, M. Chitre, and L. Ljung, "System identification via sparse multiple kernel-based regularization," *IEEE Transactions on Automatic Control*, vol. 63, no. 11, pp. 3863–3870, Nov. 2018.

[11] J. Campos Ordoñez and P. Ferguson, "Comparative analysis of machine learning regression models for unknown dynamics," *IEEE Journal of Radio Frequency Identification*, vol. 7, pp. 441–450, 2023.

[12] M. Atlante, R. Trinchero, I. S. Stievano, M. Telescu, and N. Tanguy, "SPICE-based behavioral models of IC buffers via compact kernel regressions," in *Proc. IEEE 29th Workshop on Signal and Power Integrity (SPI)*, May 2025.

[13] M. Atlante, R. Trinchero, I. S. Stievano, M. Telescu, and 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.

[14] A. Kadochnikova, Y. Zhu, Z.-Q. Lang, and V. Kadirkamanathan, "Integrated identification of the nonlinear autoregressive models with exogenous inputs (NARX) for engineering systems design," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 1, pp. 394–401, Jan. 2023.

[15] D. Yuldashev, M. Atlante, I. S. Stievano, and R. Trinchero, "A preliminary study on Kernel Ridge Regression for parametric behavioral modeling of IC buffers," in *Proc. 30th IEEE Workshop on Signal and Power Integrity (SPI)*, Turin, Italy, Jun. 14–17, 2026.

[16] 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, Art. ID 1356538, 9 pages, 2018.

[17] M. Ahadi and S. Roy, "Sparse linear regression (SPLINER) approach for efficient multidimensional uncertainty quantification of high-speed circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 10, pp. 1640–1652, Oct. 2016.

[18] P. Manfredi and R. Trinchero, "Nonparametric formulation of polynomial chaos expansion based on least-square support-vector machines," *Engineering Applications of Artificial Intelligence*, vol. 133, Art. no. 108182, 2024.

[19] R. Trinchero and F. G. Canavero, "Modeling of eye diagram height in high-speed links via support vector machine," in *Proc. IEEE 22nd Workshop on Signal and Power Integrity (SPI)*, May 2018, pp. 1–4.

[20] T. Lu, J. Sun, K. Wu, and Z. Yang, "High-speed channel modeling with machine learning methods for signal integrity analysis," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 6, pp. 1957–1964, Dec. 2018.

[21] Y. Zhao, T. Nguyen, H. Ma, E.-P. Li, A. C. Cangellaris, and J. E. Schutt-Ainé, "Modular neural network-based models of high-speed link transceivers," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 13, no. 10, pp. 1603–1612, Oct. 2023.

[22] C. H. Goay, A. Abd Aziz, N. S. Ahmad, and P. Goh, "Eye diagram contour modeling using multilayer perceptron neural networks with adaptive sampling and feature selection," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, no. 12, pp. 2427–2441, Dec. 2019.

[23] D. Jiao, Y. Dou, J. Yan, J. Zhu, and A. Norman, "Method for accurate and efficient eye diagram prediction of nonlinear high-speed links," *IEEE Transactions on Electromagnetic Compatibility*, vol. 63, no. 5, pp. 1574–1583, Oct. 2021.

[24] M. Telescu, R. Trinchero, N. Tanguy, and I. S. Stievano, "Surrogate eye modeling for the statistical assessment of a smart textile interconnect," in *Proc. IEEE 27th Workshop on Signal and Power Integrity (SPI)*, 2023, pp. 1–4.

[25] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. Singapore: World Scientific, 2002.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Components, Packaging and Manufacturing Technology. This is the author's version which has not been fully edited content may change prior to final publication. Citation information: DOI 10.1109/TCPMT.2026.3752459

[26] J. A. K. Suykens and J. Vandewalle, "Recurrent least squares support vector machines," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 7, pp. 1109–1114, Jul. 2000.
[27] I. Goethals, K. Pelckmans, J. A. K. Suykens, and B. De Moor, "Identification of MIMO Hammerstein models using least squares support vector machines," *Automatica*, vol. 41, no. 7, pp. 1263–1272, 2005.
[28] N. Soleimani, I. S. Stievano and R. Trinchero, "Small-Data Modeling of Electromagnetic Structures via Compressed Vector-Valued Kernel Ridge Regression," in *IEEE Access*, vol. 14, pp. 105178-105194, 2026.
[29] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge University Press, 2004.
[30] B. Scholkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proc. 14th Annual Conf. Computational Learning Theory*, 2001, pp. 416–426.
[31] A. Rudi, L. Carratino, and L. Rosasco, "Falkon: An optimal large scale kernel method," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
[32] 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 Transactions on Microwave Theory and Techniques*, vol. 71, no. 6, pp. 2319–2332, 2023.
[33] 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," in *Proc. IEEE MTT-S Int. Conf. Numerical Electromagnetic and Multiphysics Modeling and Optimization (NEMO)*, Winnipeg, MB, Canada, 2023.
[34] F. Alavi and S. Hashemi, "An element-wise kernel learning framework," *Applied Intelligence*, vol. 53, no. 8, pp. 9531–9547, Jun. 2023.
[35] M. G. Genton, "Classes of kernels for machine learning: A statistics perspective," *Journal of Machine Learning Research*, vol. 2, pp. 299–312, 2001.
[36] P. Drineas and M. W. Mahoney, "On the Nyström method for approximating a Gram matrix for improved kernel-based learning," *Journal of Machine Learning Research*, vol. 6, pp. 2153–2175, 2005.
[37] A. J. Smola and B. Schölkopf, "Sparse greedy matrix approximation for machine learning," in *Proc. Seventeenth International Conference on Machine Learning (ICML)*, pp. 911–918, 2000.
[38] L. Wang, H. Wang, and G. Fu, "Multi-Nyström method based on multiple kernel learning for large scale imbalanced classification," *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–12, 2021.
[39] P. Manfredi, "Conservative Gaussian process models for uncertainty quantification and Bayesian optimization in signal integrity applications," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 14, no. 7, pp. 1261–1272, Jul. 2024.
[40] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.
[41] F. Garbuglia, T. Reuschel, C. Schuster, D. Deschrijver, T. Dhaene, and D. Spina, "Modeling electrically long interconnects using physics-informed delayed Gaussian processes," *IEEE Trans. Electromagn. Compat.*, vol. 65, no. 6, pp. 1715–1723, Dec. 2023.
[42] F. Garbuglia, D. Deschrijver, and T. Dhaene, "On the role of Bayesian learning for electronic design automation: A survey," *IEEE Electromagn. Compat. Mag.*, vol. 12, no. 4, pp. 77–84, Fourth Quarter 2023.
[43] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, "Gaussian processes and kernel methods: A review on connections and equivalences," *arXiv preprint arXiv:1807.02582*, 2018.
[44] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
[45] V. Wild, M. Kanagawa, and D. Sejdinovic, "Connections and equivalences between the Nyström method and sparse variational Gaussian processes," *arXiv preprint arXiv:2106.01121*, 2023.



Dilyorjon Yuldashev (S'25) received the master's degree mechatronic engineering in control and computer Engineering from Politecnico di Torino, Turin, Italy, in 2019. He is currently pursuing the Ph.D. degree in Electrical Engineering with the Department of Electronics and Telecommunications at Politecnico di Torino. His research interests include modeling and simulation of electrical and electronic systems, statistical methods, and machine learning techniques.



Igor S. Stievano (M'98–SM'08) received the master's degree in electronic engineering and the Ph.D. degree in electronics and communication engineering from Politecnico diTorino, Turin, Italy, in 1996 and 2001, respectively. He is currently a professor of electrical engineering with the Department of Electronics and Telecommunications, Politecnico di Torino. He has authored or co-authored more than140 papers published in international journals and conference proceedings. His research interests include modeling and simulation of electrical and electronic systems, with an emphasis on behavioral modeling of digital integrated circuits, signal, and power integrity, energy networks, statistical, and machine learning methods.



Riccardo Trinchero (M'16) received the M.Sc. and the Ph.D. degrees in Electronics and Communication Engineering from Politecnico di Torino, Torino, Italy, in 2011 and 2015, respectively. He is currently an Associate Professor within the EMC Group with the Department of Electronics and Telecommunications at the Politecnico di Torino. His research interests include the analysis of switching DC-DC converters, machine learning, and statistical simulation of circuits and systems.

Authorized licensed use limited to: Politecnico di Torino. Downloaded on November 12,2026 at 07:43:11 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.