Bayesian Optimization of Hyperparameters in Kernel-Based Delay Rational Models

Bayesian Optimization of Hyperparameters in Kernel-Based Delay Rational Models

Felipe Treviso
Dept. of Electronics and Telecom.
Politecnico di Torino
Torino, Italy
felipe.treviso@polito.it

Riccardo Trinchero
Dept. of Electronics and Telecom.
Politecnico di Torino
Torino, Italy
riccardo.trinchero@polito.it

Flavio G. Canavero
Dept. of Electronics and Telecom.
Politecnico di Torino
Torino, Italy
flavio.canavero@polito.it

***Abstract*—This paper presents an automatic procedure for the optimization of the hyperparameters of a delay rational model approximating the frequency-domain behavior of high-speed interconnects. The proposed model is built via a kernel-based regression, such as the Least-Square Support Vector Machine (LS-SVM), by considering an ad-hoc kernel with two hyperparameters related to the propagation delays introduced by the system. Such hyperparameters, along with the Tikhonov regularizer used by the LS-SVM regression, are carefully tuned via an automatic approach based on a k-fold cross-validation and Bayesian optimization. The feasibility of the effectiveness of the proposed modeling approach are investigated on a high-speed link.**

***Index Terms*—High-speed link, delay rational model, Machine Learning regression, cross-validation.**

I. Introduction

In high-speed links, the availability of accurate and fast interconnect models is essential for signal and power integrity (SPI) and electromagnetic interference (EMI) predictions through simulations. The simulations of the channel can be performed in an early phase of the system design, and thus represent an useful tool for the design optimization and the assessment of its performance. Due to the inherent linear time invariant nature of the link interconnects, their models are usually based on the so-called rational models. However, accurate rational models of long interconnects usually require a large number of poles. Such large number of poles unavoidably has an impact on the simulation time required by a transient simulation involving systems composed by such interconnects. An alternative to reduce the model complexity and speed up those analyses is the use of delay rational models (DRM) [1]–[3], which account explicitly for the delay present in such structures.

A promising method for the estimation of a DRM is to use the least-squares support vector machine (LS-SVM) [4]. This kernel-based technique allows the manipulation of infinite dimensional spaces, such that the model ensures that the exact delays of the system are accounted. By doing so, it is possible to accurately identify the delays of the original transfer function [5], [6], which is the hardest step to build a DRM. However, some parameters of the model and kernel should be tuned in order to obtain a better performance. The tuning of these so-called hyperparameters is a critical step in the estimation of this approximated model, and the optimal approximation can only be reached through an optimization of such hyperparameters.

Due to the complex structure of the model, the above optimization is a non-convex problem, which may achieve local minima if performed with standard optimization techniques. This paper presents an automatic approach for the optimization of the model hyperparameters and thus of its accuracy, based on the combination of a k-fold cross-validation (CV) [7] scheme and a Bayesian optimization [8]. The strength and the robustness the proposed approach are investigated on a high-speed link.

II. Kernel-Based Delay Rational Model

Delay rational models are a natural choice to represent distributed systems [1], [2]. Those models approximate sampled pairs of data from a transfer function $\{(s_k, H(s_k))\}_{k=1}^{K}$ as a linear combination of basis functions containing poles $p_j \in \mathbb{C}$ and delays $\tau_i \in \mathbb{R}$ as

$$H(s_k) \approx \tilde{H}(s_k) = r_0 + \sum_{i=1}^{n_\tau} \sum_{j=1}^{n_p} \frac{r_{ij}}{s_k - p_j} e^{-s_k \tau_i}, \qquad (1)$$

where the residues $r_{ij} \in \mathbb{C}$ are the coefficients of this linear combination and r_0 is a constant bias term.

Kernel-based techniques are a class of machine learning (ML) regression models which can be applied to fit the data of a transfer function. For that regression, they use as input $s \in \mathbb{C}$ and output $\tilde{H}(s) \in \mathbb{C}$. The dual space formulation of this model reads:

$$\tilde{H}(s) = \sum_{k=1}^{K} \alpha_k \mathrm{k}(s, s_k) + b, \qquad (2)$$

where $b \in \mathbb{C}$ is a bias constant, $\alpha_k \in \mathbb{C}$ represent the K model coefficients and $\mathrm{k}(s, s_k)$ is the so-called kernel function. This is a non-parametric model where the number of estimated coefficients is always equal to $K + 1$ (the number of training samples plus one), independent of the number and shape of the basis functions that such kernel reproduces. On the other hand, the model in (1) requires the estimation of $n_p n_\tau + 1$ coefficients, a number that changes according to the number of basis functions accounted for by the model.

Considering the LS-SVM framework, the dual formulation model in (2) can be suitably estimated by solving a system of linear equations [9]. The main requirement for this representation is the definition of the kernel function, which can be done through [10]:

$$\mathrm{k}(s, s_k) = \langle \boldsymbol{\varphi}(s), \boldsymbol{\varphi}(s_k) \rangle . \tag{3}$$

where $\boldsymbol{\varphi}(s)$ are the basis functions used in the model.

In order that the above model corresponds to a DRM, the kernel function equates to:

$$\mathrm{k}(s, s_k) = \mathrm{k}_p(s, s_k)\mathrm{k}_\tau(s, s_k), \tag{4}$$

with

$$\mathrm{k}_p(s, s_k) = \sum_{j=1}^{n_p} \frac{|p'_j|}{(s - p_j)\left(s_k^* - p_j^*\right)}, \tag{5}$$

and

$$\mathrm{k}_\tau(s, s_k) = \begin{cases} \frac{\left(e^{-\tau_M(s^*+s_k)} - e^{-\tau_m(s^*+s_k)}\right)}{-(s^*+s_k)}, & s^* + s_k \neq 0 \\ \tau_M - \tau_m, & s^* + s_k = 0 \end{cases}. \tag{6}$$

The above kernel depends on the definition of the n_p poles $p_j = p'_j + jp''_j$, and of an interval from a minimum delay τ_m to a maximum τ_M where the system's propagation delays should be located. Once defined, such kernel represents a space formed by bases in the following form:

$$\varphi_j(s; p_j, \tau) = \frac{|p'_j|^{1/2}}{s - p_j} e^{-s\tau}, \tag{7}$$

which is equivalent to the basis of the DRM in (1). In the specific case of (4), the poles are a discrete set (i.e., $\{p_1, \ldots, p_{n_p}\}$), while $k_\tau(s, s_k)$ accounts for all possible delay terms between τ_m and τ_M. Indeed, the use of such kernel provides a feature space with an infinite number of dimensions, i.e., an infinite number of basis.

Taking into account the kernel definition, the kernel-based model can be equivalently represented in its primal space formulation as

$$\tilde{H}(s) = \sum_{j=1}^{n_p} \int_{\tau_m}^{\tau_M} \mathrm{w}'_j(\tau)\varphi_j(s; p_j, \tau)d\tau + b, \tag{8}$$

where

$$\mathrm{w}'_j(\tau) = \sum_{k=1}^{K} \alpha_k \frac{|p'_j|^{1/2}}{s_k^* - p_j^*} e^{-s_k^*\tau} \tag{9}$$

are constant coefficients defined for every combination of p_j and τ considered in the model.

After estimating the model, the weight $\mathrm{w}'_j(\tau)$ provides information on the values of p_j and specially, τ, that have a larger influence in the model. Indeed, we can define the following total weight $W(\tau)$:

$$W(\tau) = \sqrt{\sum_{j=1}^{n_p} \|\mathrm{w}(\tau, p_j)\|^2}, \tag{10}$$

which sums the contributions of all the poles in the final model, leading to a τ-dependency only. The analysis of (10) allows the accurate identification of the dominant propagation delays of the system [5]. Those delays are the τ values where the peaks of $W(\tau)$ occur. Those few identified delays can be used to estimate compact delay rational models with a small number of poles [1].

The above format of model was shown to be accurate without the use of optimal poles [5]. However, similar to the standard formulation of the DRM, the considered delay interval has a large impact in the performance of the model, and therefore must be carefully tuned.

III. Tuning of the Model Hyperparametters via Cross validation and Bayesian Optimization

The accuracy of the modeling scheme presented in the previous section unavoidably depends on the tuning of the kernel hyperparameters τ_m and τ_M, which provide information on the minimum and maximum candidate delays of the system, and of the Tikhonov regularizer γ used by the LS-SVM regression. As an example, if a small value for the hyperparmetr τ_M is used, the model may be unable to follow the dynamic phase variations produced by the system. On the other hand, if a more conservative strategy is considered, as an example by using the largest delay interval allowed by the frequency sampling (i.e., $\tau_m = 0\,\mathrm{s}$ and $\tau_M = 1/\Delta_f$), the delay identification procedure can be rather cumbersome, since the values of $W(\tau)$ must be analyzed in a large interval. The regularizer γ provides a trade-off between the model error on the training samples and the 2-norm of the model coefficients $\|\mathrm{w}'_j(\tau)\|_2^2$, thus preventing over-fitting.

Techniques such as k-fold CV [7] and Bayesian optimization [8] can be seen as promising candidates to overcome the above issues, since they are widely used within ML techniques to select, without any manual tuning, the best configuration of the model hyperparmeters. In the k-fold CV [7], the training set is split into k smaller sets, called folds. Then, for each of the k folds, the model is trained using $k-1$ folds as training data and by using a given configuration of the hyperparameters $\boldsymbol{\lambda} = \{\gamma, \tau_m, \tau_M\}$, while the remaining fold is kept as a validation set (i.e., to evaluate the model accuracy on data which were not used during the training). The above scheme is iterated for all the k-folds. Then, for each analyzed combination of the hyperparameters $\boldsymbol{\lambda}$, the overall model performance is assessed by the k-fold CV error $CV_{\text{error}}^{(\boldsymbol{\lambda})}$, which is the average of the values computed during the k iterations, according to

$$CV_{\text{error}}^{(\boldsymbol{\lambda})} = \frac{1}{k} \sum_{n=1}^{k} CV_{\text{error},n}^{(\boldsymbol{\lambda})}, \tag{11}$$

where $CV_{\text{error},n}^{(\boldsymbol{\lambda})}$ is the mean squared error of the model on the n-th test fold.

The common choice for k is usually 5 or 10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance. An illustration of the k-fold CV for the case

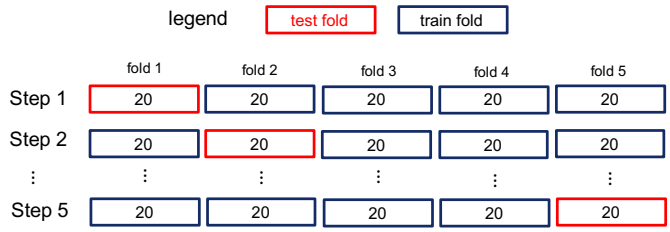

Fig. 1. Example: a k-fold cross-validation with $k = 5$ and $K = 100$ training samples.

with $K = 100$ training samples and 5 folds (i.e., $k = 5$) is shown in Fig. 1.

The optimum value of the hyperparameters $\boldsymbol{\lambda}^\star = \{\gamma^\star, \tau_m^\star, \tau_M^\star\}$ is selected as the one that minimizes the corresponding overall CV error $CV_{\text{error}}^{(\boldsymbol{\lambda})}$, i.e.,

$$\boldsymbol{\lambda}^\star = \arg\min_{\boldsymbol{\lambda}} CV_{\text{error}}^{(\boldsymbol{\lambda})}. \tag{12}$$

Together with the CV, a Bayesian optimization algorithm is used to solve the minimization problem in (12). This algorithm specifically selects the next evaluated configuration of the hyperparameters $\boldsymbol{\lambda}$ in order to maximize the reward towards finding the global optimum of a non-convex function [8]. By using this sampling scheme, the search converges to the optimum solution in fewer iterations than with standard search schemes, while keeping the capacity to avoid local minima. An open-source implementation of this optimization scheme is available in the scikit-optimize Python library [11].

IV. Application Example

The proposed methodology is exemplified by considering the transfer function $H(s) = V_{out}(s)/E(s)$ for $s = j\omega = j2\pi f$ of the high-speed link circuit in Fig. 2. The link consists of three transmission lines based on microstrips, together with lumped elements that represent the parasitic effects of the link, in order to approximate the structure of a realistic interconnect. This structure has been implemented and simulated in HSPICE in a bandwidth from 0 to 20 GHz. Simulation results consists of $K = 1001$ frequency points with a frequency spacing $\Delta_f = 20$ MHz. A subset containing 101 samples, randomly selected among the available data, has been used as a validation dataset, whilst the remaining 900 samples are used as training set to construct the LS-SVM model.

A DRM based on the modeling approach presented in Sec. II is constructed to approximate $H(s)$. The set of poles $\boldsymbol{p}$ is defined by drawing its real and imaginary parts randomly and independently from a normal distribution $\mathcal{N}(0, 4\pi \times 10^9)$ with zero mean and standard deviation of $4\pi \times 10^9$ rad/s. In the cases where the real part of the pole was positive, corresponding to an unstable pole, it was forced to be negative by flipping its sign.

During the model training, the optimum set of hyperparameters γ, τ_m and τ_M is selected via the CV-based Bayesian optimization scheme presented in Sec. III using 5 folds and 50 iterations. The parameters search space is restricted to

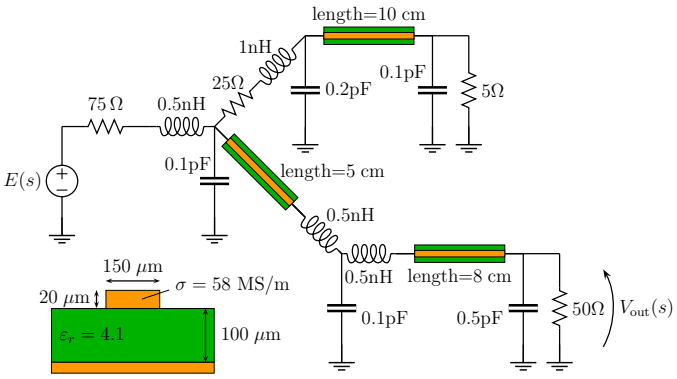

Fig. 2. Schematic of the circuit modeled in the application example.

the intervals $\gamma \in [10^3, 10^{25}]$ and $\tau_m, \tau_M \in [0, 50]$ ns. The constraint $\tau_M > \tau_m$ is not enforced, however it is expected as the logical outcome of the optimization. After the 50 cycles, the obtained optimized parameters are $\gamma^\star = 6.56 \times 10^{21}$, $\tau_m^\star = 5.68 \times 10^{-8}$ and $\tau_M^\star = 7.39$ ns. Figures 3 and 4 show that these parameters provide a very accurate model when applying it to the validation data, where the model output almost perfectly matches the original points of $H(s)$.

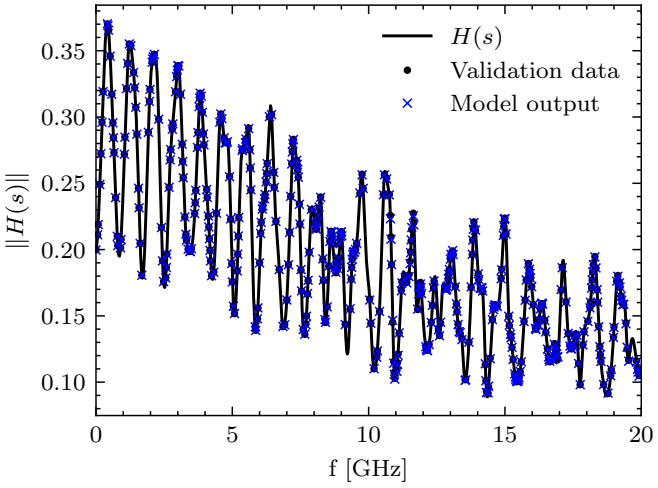

Fig. 3. Magnitude comparison of the LS-SVM model output and the original transfer function $H(s)$.

Additionally, the proposed LS-SVM model of $H(s)$ can be used to identify the dominating propagation delays of the system of Fig. 2. These delays should be searched only within the optimized delay interval $\tau_m^\star$ to $\tau_M^\star$, making it possible to perform a more fine discretization without incurring into an unreasonable computational time. The delays are identified from $W(\tau)$, which is computed according to (10). All this computational procedure took only 940.7 s, of which 937.2 s were used for the estimation of the optimized model and 3.5 s for the delay identification. The plot of $W(\tau)$ is shown in Fig. 5. In this figure, the black curve provided by the optimized model is compared with the blue one obtained by means of a basic tuning of the parameters, where τ_m is set to its minimum

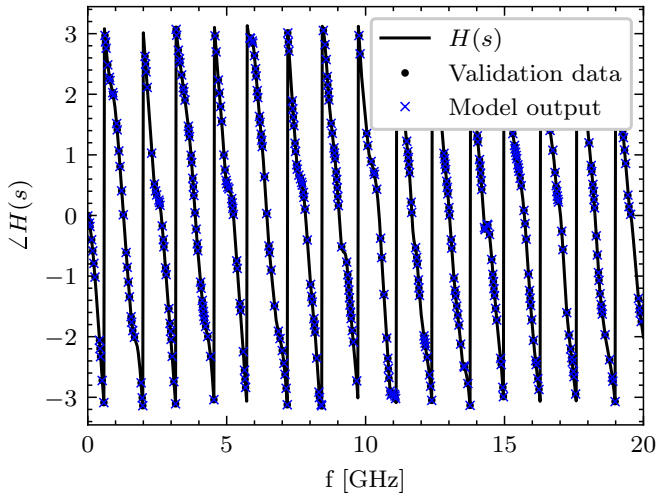

Fig. 4. Phase comparison of the LS-SVM model output and the original transfer function $H(s)$.

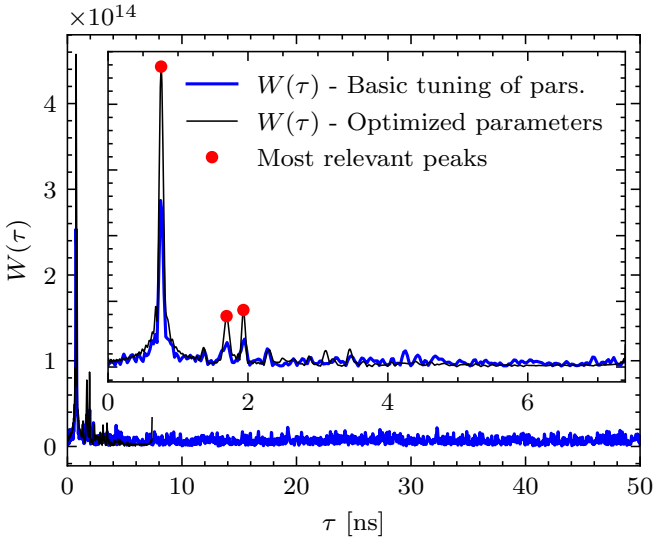

Fig. 5. Plot of $W(\tau)$ obtained from the LS-SVM model of $H(s)$ indicating the dominating propagation delays of the system of Fig. 2. The detail amplifies it in the interval from $\tau_m^\star$ to $\tau_M^\star$.

and τ_M and γ are set to their maximum possible values. It is observed that the peaks of the black curve are much more identifiable than in the blue curve, while it is also less noisy. Moreover, τ_M in the blue curve goes up to 50 ns, which results in a worse resolution of the τ-axis if a fixed number of points is considered in the discretization. The peaks of such plot correspond to the propagation delays produced by the original transfer function. For example, the first marked peak occurs at 0.76 ns, while the two transmission lines in the main signal path have a total length of 13 cm. Those values represent a propagation speed of 1.71×10^8 m/s, which is compatible with the real speed in such structures. The additional peaks are also clear in the plot, together with smaller peaks that can be identified if necessary. Such obtained delays are useful to the estimation of compact and accurate DRMs.

V. Conclusion

Kernel-based techniques are a very flexible tool to deal with systems with delays. They provide a way to model the data using a very large or infinite number of basis. This fact makes it possible to ensure that the true propagation delays of the distributed system are accounted in the estimated model. However, the added flexibility comes at the cost of tuning some extra parameters. Those parameters were optimized via a Bayesian method together with a 5-fold cross validation. Such optimization provided an accurate kernel-based model after few iterations. The optimized model provides an easy way to identify the system dominant propagation delays, where the delays should be searched in a small interval given by the optimized parameters, making it a simpler task than if the whole possible interval would be considered.

References

[1] A. Chinea, P. Triverio, S. Grivet-Talocia, *"Delay-Based Macromodelling of Long Interconnects From Frequency-Domain Terminal Responses"*, IEEE Trans. on Advanced Packaging, Vol. 33, No. 1, Feb. 2010.

[2] A. Chinea *et al.*, *"Signal Integrity Verification of Multichip Links Using Passive Channel Macromodels"*, IEEE Trans. on Components, Packaging and Manufact. Tech., vol. 1, no. 6, pp. 920-933, Jun. 2011.

[3] E. R. Samuel, L. Knockaert and T. Dhaene, *"Model Order Reduction of Time-Delay Systems Using a Laguerre Expansion Technique,"* in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 61, no. 6, pp. 1815-1823, June 2014.

[4] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002 (ISBN 981-238-151-1).

[5] F. Treviso, R. Trinchero and F. G. Canavero, *"Machine Learning Applied to the Blind Identification of Multiple Delays in Distributed Systems,"* 2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science, Rome, Italy, 2020.

[6] F. Treviso, R. Trinchero and F. G. Canavero, *"Multiple Delay Identification in Long Interconnects via LS-SVM Regression,"* in IEEE Access, vol. 9, pp. 39028-39042, 2021, doi: 10.1109/ACCESS.2021.3063713.

[7] B. Ghojogh and M. Crowley, *"The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial"*, arXiv preprint arXiv:1905.12787, 2019.

[8] E. Brochu, M. Cora,and N.de Freitas, *"A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning"*, arXiv preprint arXiv:1012.2599, 2010.

[9] R. Trinchero, *et al.*, *"Machine Learning and Uncertainty Quantification for Surrogate Models of Integrated Devices With a Large Number of Parameters,"* in IEEE Access, vol. 7, pp. 4056-4066, 2019.

[10] N. Cristianini, J. Shawne-Taylor, *An Introduction to Support Vector Machines and Other Kernel Based Learning Methods*, Cambridge University Press, UK, 2000 (ISBN 0-521-78019-5).

[11] Head, Tim, Kumar, Manoj, Nahrstaedt, Holger, Louppe, Gilles, and Shcherbatyi, Iaroslav, *"scikit-optimize/scikit-optimize"*. Zenodo, 04-Sep-2020.