KARMA: Kernel-Aided Rational Macromodeling of Noisy Electrical Frequency Response Data

Thijs Ullrick, *Member, IEEE*, Dirk Deschrijver, *Senior Member, IEEE*, Wim Bogaerts, *Fellow, IEEE*, and Tom Dhaene, *Senior Member, IEEE*

***Abstract*—Gaussian processes (GPs) have attracted significant interest for macromodeling frequency-domain responses of linear time-invariant (LTI) systems. Yet, a unified and practical approach to extend their use to the time domain has remained elusive. This article introduces kernel-aided rational macromodeling (KARMA), a novel framework that constructs compact, multiport rational representations suitable for circuit simulation directly from frequency response data. By leveraging the unique properties of complex-valued rational kernels, KARMA fuses the advantages of kernel-based nonparametric methods with the physical properties of traditional rational parametric macromodeling techniques. The framework comprises critical components such as joint hyperparameter optimization, passivity enforcement, and model order reduction (MOR). Moreover, the probabilistic nature of GPs enables KARMA to generate uncertainty-aware macromodels, making the framework particularly effective in noisy or data-scarce settings.**

***Index Terms*—Frequency-domain system identification, Gaussian processes (GPs), linear time-invariant (LTI) systems, macromodeling, physics-informed kernels, rational modeling.**

I. INTRODUCTION

RATIONAL macromodels provide the accuracy and computational efficiency required for high-fidelity circuit simulation and remain an indispensable tool for the development of next-generation microwave and RF systems [1]. Many high-frequency components, such as filters, couplers, antennas, and transmission lines, are inherently characterized in the frequency domain. Their behavior is commonly described by the scattering matrix (S-matrix), which relates the incident and reflected power waves at each port of an electrical network, effectively capturing the complex coupling and interaction between ports. While it might seem straightforward to apply the inverse fast Fourier transform (IFFT) to obtain the corresponding time-domain representation, practical limitations such as band-limited or truncated frequency response data often lead to inaccuracies and violations of physical properties, such as causality and passivity [2].

In this context, rational macromodeling techniques, which approximate the transfer function of an LTI system as a ratio of two polynomials, provide a robust mathematical framework for generating state-space representations, facilitating efficient analysis in both the time and frequency domain. The goal is to derive a rational model in pole–residue form

$$\mathbf{S}(s) = \sum_{i=1}^{M} \frac{\mathbf{R}_i}{s - \tilde{p}_i} + \mathbf{D} \tag{1}$$

where $\mathbf{S}(s)$ denotes the P-port S-matrix, $s = j2\pi f$ denotes the Laplace variable evaluated on the imaginary axis, $\{\tilde{p}_i\}_{i=1}^{M}$ represents the common pole set (with poles either real or occurring in complex-conjugate pairs), $\mathbf{R}_i \in \mathbb{C}^{P\times P}$ are the residue matrices associated with the poles, and $\mathbf{D} \in \mathbb{R}^{P\times P}$ models the asymptotic response at high frequencies.

System identification methods are broadly categorized into parametric and nonparametric approaches. Parametric methods rely on the assumption of a predefined deterministic model structure and focus on estimating the associated parameters. Rational macromodeling techniques fall within this category, as they assume a rational functional form defined by poles and residues. One widely adopted technique is the vector fitting (VF) algorithm, which iteratively relocates the poles and adjusts the residues of the rational model (1) to minimize the least-squares (LSs) error with the data [3], [4], [5], [6]. Another method, the adaptive Antoulas–Anderson (AAA) algorithm, employs barycentric interpolation, and adaptively refines the model parameters through a two-step procedure based on a linear LS problem and a greedy strategy [7]. Although parametric methods such as VF and AAA have proven to be highly effective for modeling multiport S-parameters, they exhibit some limitations. First, their performance is significantly affected by challenges related to model order selection. Moreover, neither VF nor AAA inherently model noise in the data. The AAA algorithm forces the rational fit to pass through the input data values (i.e., interpolant values) making it impractical when the data is contaminated with noise [8]. VF, on the other hand, approximates the system response by minimizing the LSs error, making it naturally more suited for handling noisy data. However, studies show that even small amounts of noise can severely impact the convergence of the VF algorithm [9]. Specifically,

Received 10 July 2025; revised 7 September 2025 and 8 October 2025; accepted 10 October 2025. This work was supported in part by the Flemish Research Foundation (FWO-Vlaanderen) under Grant G031421N and in part by the Flanders Artificial Intelligence (AI) Research Program. *(Corresponding author: Thijs Ullrick.)*

Thijs Ullrick is with the Internet Technology and Data Science Laboratory (IDLab), Department of Information Technology, and the Photonics Research Group, Department of Information Technology, Ghent University-imec, 9000 Ghent, Belgium (e-mail: thijs.ullrick@ugent.be).

Dirk Deschrijver and Tom Dhaene are with IDLab, Department of Information Technology, Ghent University-imec, 9000 Ghent, Belgium (e-mail: dirk.deschrijver@ugent.be; tom.dhaene@ugent.be).

Wim Bogaerts is with the Photonics Research Group, Department of Information Technology, Ghent University-imec, 9000 Ghent, Belgium (e-mail: wim.bogaerts@ugent.be).

Digital Object Identifier 10.1109/TMTT.2025.3621537

the presence of noise can bias the LSs solution, leading to an increased number of iterations or inaccurate rational approximations. To mitigate these issues, several advanced formulations have been proposed, including hard pole relocation [9], mixed VF–Newton iterations [10], and instrumental variable techniques [11]. Although these methods improve robustness against noise, they do not guarantee convergence in all cases, and model order selection remains critical. Another limitation of the VF and AAA algorithms stems from their parametric nature: as the model order increases, the number of parameters to be estimated grows accordingly, leading to increased computational complexity.

Over the past decade, kernel-based methods have gained significant popularity across various engineering domains, demonstrating their effectiveness in solving complex problems [12]. These methods belong to the class of nonparametric machine learning (ML) models, which do not rely on a fixed number of parameters, such as the weights and biases of an artificial neural network (ANN). Instead, they are constructed directly from the data, offering a high degree of expressiveness and enabling accurate predictions even with limited training data. These methods utilize kernel functions to implicitly map input data into a higher dimensional feature space, allowing them to effectively model complex, nonlinear relationships. Within this kernel-based framework, traditional challenges such as model order selection and parameter estimation are reformulated in terms of kernel design and hyperparameter optimization. Hyperparameter tuning, treated as a continuous optimization problem, plays a crucial role in this process by balancing model complexity and generalization across the design space. This method eliminates the need for manual order selection and mitigates the risk of overfitting.

Kernel-based methods have been applied to modeling frequency-domain responses using a variety of techniques, including support vector machines (SVMs) [13], [14], [15], GPs [16], [17], [18], [19], and kernel ridge regression (KRR) [20], [21]. In this work, GPs are the preferred choice since they offer several advantages over SVMs. Specifically, GPs provide a Bayesian probabilistic framework for hyperparameter optimization via marginal likelihood estimation (MLE) [22]. Owing to the likelihood's closed-form formulation, hyperparameter optimization becomes a computationally efficient process, enabling the use of complex composite kernels with many hyperparameters. In contrast, hyperparameter optimization in SVMs typically relies on grid sweeps or cross-validation, limiting the number of hyperparameters in the kernel to one or two. Furthermore, due to their stochastic nature, GPs inherently represent uncertainty in their predictions, making them well-suited for uncertainty quantification, particularly when dealing with limited or noisy data.

GP-based approaches were first introduced in [12] and have since proven effective in the frequency-domain identification of LTI systems [16], [17], [18], [19]. While these models have been extensively studied in the frequency domain, a comprehensive framework for their application in the time domain remains absent, with no significant efforts dedicated to this area. This constitutes a significant gap in the literature: Although passive devices are typically characterized in the frequency domain, accurate and compact circuit models are needed for the characterization of transient effects and interactions with other nonlinear components in complex circuits. Consequently, parametric algorithms such as VF and AAA, whose rational representations naturally yield a system of ordinary differential equations (ODEs), often remain the preferred choice over kernel-based methods [16], [17], [18], [19].

To bridge the existing gap in the literature, a novel rational macromodeling framework is defined, which builds upon the unique properties of the rational kernels presented in [16] and [17]. By adopting a rational kernel, it is demonstrated that the GP's predictive mean can be represented in a rational functional form. This key insight reveals an intricate relationship between the GP model and traditional VF-like formulations. As a result, the extensive body of work originally developed for rational macromodels, including model order reduction (MOR) and passivity enforcement techniques, becomes directly applicable to these kernel-based nonparametric models. This result allows the definition of a new class of rational macromodels, combining the advantages of complex-valued kernel-based methods with the physical properties of VF-like formulations.

II. PRELIMINARY

This section presents an overview of fundamental concepts in kernel-based frequency-domain system identification, as discussed in [16] and [17], forming the basis for the kernel-aided rational macromodeling (KARMA) framework introduced later in this article. The discussion mainly builds on Bect's theoretical foundations [17], and aims to offer an intuitive, tutorial-style overview of complex-valued kernel-based approaches.

A. Problem Statement

Consider a system whose true frequency response is given by $\tilde{\mathcal{H}}(f) : \mathbb{R} \rightarrow \mathbb{C}$, where f denotes the frequency. The evaluation of this system at a frequency f_n can be modeled as a noisy observation, expressed as follows:

$$\tilde{y}\,(f_n) = \tilde{\mathcal{H}}\,(f_n) + \tilde{\varepsilon} \tag{2}$$

where $\tilde{\varepsilon}$ represents the noise term whose real and imaginary components are assumed to be independently Gaussian distributed according to $\sim \mathcal{N}(0, \eta^2)$. Given a vector of noisy observations $\tilde{\mathbf{y}} = [\tilde{y}_1, \ldots, \tilde{y}_N]^{\mathrm{T}}$ at corresponding frequency points $\mathbf{f} = [f_1, \ldots, f_N]^{\mathrm{T}}$, the goal is to construct a macromodel that can predict the system response at previously unseen frequencies, while effectively handling the noise in the observed data. Note that the tilde symbol is used throughout this article to indicate the complex-valued nature of functions and variables.

B. Complex GP Regression

GPs are powerful probabilistic models commonly used for nonparametric regression and function approximation. A GP defines a distribution over functions such that any finite collection of function values follows a multivariate Gaussian distribution.

Formally, a multioutput GP (MOGP) is designed to simultaneously model T outputs. Specifically, in this work, the MOGP is adopted to model the real and imaginary components ($T = 2$) of the complex-valued frequency response. In addition, it is assumed that both real and imaginary outputs share the same input f. In this framework, the real-valued target function, $\boldsymbol{\mathcal{H}}(f) : \mathbb{R} \to \mathbb{R}^2$, representing the real and imaginary parts, is said to follow a GP if

$$\boldsymbol{\mathcal{H}}(f) \sim \mathcal{GP}\left(\mathbf{m}(f), \mathbf{k}\left(f, f'\right)\right) \tag{3}$$

where $\mathbf{m}(f) = \mathbb{E}[\boldsymbol{\mathcal{H}}(f)]$ is the prior mean function, and the covariance function $\mathbf{k}(f, f') \in \mathbb{R}^{2\times 2}$ is defined as follows:

$$k_{tt'}\left(f, f'\right) = \mathbb{E}\left[\left(\mathcal{H}_t(f) - m_t(f)\right)\left(\mathcal{H}_{t'}\left(f'\right) - m_{t'}\left(f'\right)\right)\right] \tag{4}$$

where $t, t' \in \{\mathfrak{R}, \mathfrak{I}\}$ and $k_{tt'}(f, f')$ encodes the degree of correlation between the real and imaginary parts.

A complex GP (CGP) extends the concept of real GPs to complex-valued functions. A CGP is a complex stochastic process, where the real and imaginary components, considered jointly, form a real GP [23]. Given that the frequency-domain system response (2) is complex-valued, CGPs are a natural choice for modeling such data. Let us consider a complex-valued random process $\tilde{\xi}$ defined on $\mathbb{C}$ with real-valued input f, then the CGP is defined by its covariance function $\tilde{k}$ and pseudocovariance function $\tilde{c}$ expressed as follows:

$$\begin{aligned}\tilde{k}\left(f, f'\right) &= \mathbb{E}\left[\left(\tilde{\xi}(f) - \tilde{m}(f)\right)\left(\tilde{\xi}\left(f'\right) - \tilde{m}\left(f'\right)\right)^*\right]\\ \tilde{c}\left(f, f'\right) &= \mathbb{E}\left[\left(\tilde{\xi}(f) - \tilde{m}(f)\right)\left(\tilde{\xi}\left(f'\right) - \tilde{m}\left(f'\right)\right)\right]\end{aligned} \tag{5}$$

where $\tilde{m}(f) = \mathbb{E}[\tilde{\xi}(f)]$ and $(\cdot)^*$ denotes complex conjugation. The pseudocovariance function $\tilde{c}(f, f')$ provides additional information on the dependency between real and imaginary components. If $\tilde{c}(f, f') = 0$, then $\tilde{\xi}$ is said to be *proper* (i.e., circularly symmetric).

Although CGPs offer a powerful framework for modeling complex-valued functions, only a limited number of software tools support this capability. For instance, popular GP libraries like GPyTorch [24] and GPflow [25] do not provide native support for CGPs. However, the CGP covariance function $\tilde{k}$ and pseudocovariance function $\tilde{c}$ may be expressed as follows:

$$\begin{aligned}\tilde{k} &= k_{\mathfrak{R}\mathfrak{R}} + k_{\mathfrak{I}\mathfrak{I}} + j\left(k_{\mathfrak{I}\mathfrak{R}} - k_{\mathfrak{R}\mathfrak{I}}\right)\\ \tilde{c} &= k_{\mathfrak{R}\mathfrak{R}} - k_{\mathfrak{I}\mathfrak{I}} + j\left(k_{\mathfrak{I}\mathfrak{R}} + k_{\mathfrak{R}\mathfrak{I}}\right)\end{aligned} \tag{6}$$

where $k_{\mathfrak{R}\mathfrak{I}}$, $k_{\mathfrak{I}\mathfrak{I}}$, $k_{\mathfrak{R}\mathfrak{I}}$, and $k_{\mathfrak{I}\mathfrak{R}}$ are the covariances and cross-covariances of the real and imaginary parts, respectively. Leveraging this formulation, the CGP may be implemented within the real-valued MOGP framework by treating the real and imaginary parts of the complex-valued function as separate real-valued outputs. The covariance matrix of the resulting MOGP is expressed as follows:

$$\mathbf{k}\left(f, f'\right) = \begin{bmatrix} k_{\mathfrak{R}\mathfrak{R}}\left(f, f'\right) & k_{\mathfrak{R}\mathfrak{I}}\left(f, f'\right)\\ k_{\mathfrak{I}\mathfrak{R}}\left(f, f'\right) & k_{\mathfrak{I}\mathfrak{I}}\left(f, f'\right)\end{bmatrix} \tag{7}$$

where

$$\begin{aligned}k_{\mathfrak{R}\mathfrak{R}} &= \frac{\mathrm{Re}\left[\tilde{k} + \tilde{c}\right]}{2}, & k_{\mathfrak{R}\mathfrak{I}} &= \frac{\mathrm{Im}\left[-\tilde{k} + \tilde{c}\right]}{2}\\ k_{\mathfrak{I}\mathfrak{R}} &= \frac{\mathrm{Im}\left[\tilde{k} + \tilde{c}\right]}{2}, & k_{\mathfrak{I}\mathfrak{I}} &= \frac{\mathrm{Re}\left[\tilde{k} - \tilde{c}\right]}{2}\end{aligned} \tag{8}$$

and the mean function is given by $\mathbf{m}(f) = [\tilde{m}_{\mathfrak{R}}(f), \tilde{m}_{\mathfrak{I}}(f)]$.

By conditioning the GP prior on the noisy observations, the posterior mean is obtained, enabling inference at new test frequencies f. Specifically, the predicted frequency response $\hat{\boldsymbol{\mathcal{H}}}(f) = [\hat{\boldsymbol{\mathcal{H}}}_{\mathfrak{R}}(f), \hat{\boldsymbol{\mathcal{H}}}_{\mathfrak{I}}(f)] = \mathbb{E}[\boldsymbol{\mathcal{H}}(f)|\mathbf{y}, \mathbf{f}]$ at the new frequency f is given by

$$\hat{\boldsymbol{\mathcal{H}}}(f) = \mathbf{m}(f) + K(f, \mathbf{f})\left(K(\mathbf{f}, \mathbf{f}) + \eta^2 I\right)^{-1}(\mathbf{y} - \mathbf{m}(\mathbf{f})). \tag{9}$$

The GP predictive covariance is expressed as follows:

$$\mathbb{V}\left[\boldsymbol{\mathcal{H}}(f)\right] = K(f, f) - K(f, \mathbf{f})\left(K(\mathbf{f}, \mathbf{f}) + \eta^2 I\right)^{-1} K(\mathbf{f}, f). \tag{10}$$

Here, $\mathbf{y} = [\tilde{\mathbf{y}}_{\mathfrak{R}}, \tilde{\mathbf{y}}_{\mathfrak{I}}] \in \mathbb{R}^{2N\times 1}$ is the vector containing the real and imaginary parts of the observed data, $\mathbf{m}(\mathbf{f}) = [\tilde{\mathbf{m}}_{\mathfrak{R}}(\mathbf{f}), \tilde{\mathbf{m}}_{\mathfrak{I}}(\mathbf{f})] \in \mathbb{R}^{2N\times 1}$ is formed by stacking the real and imaginary components of the mean function, respectively, and $K \in \mathbb{R}^{2N\times 2N}$ is the multioutput Gram matrix that encodes the pairwise covariance between observed frequencies, as defined by the kernel function $\mathbf{k}$. The statistical properties of the noise are captured by a single learnable hyperparameter, incorporated in the GP by augmenting the covariance matrix $K(\mathbf{f}, \mathbf{f})$ with a diagonal term $\eta^2 I$. This regularization accounts for Gaussian white noise and allows the GP to distinguish between systematic signal variations and random noise. Consequently, this formulation enhances the model's predictive accuracy and its ability to quantify uncertainty. Remark that this is a major advantage over parametric rational macromodeling techniques such as VF or AAA, which do not inherently account for uncertainty in the observations.

C. Rational Kernels

Bect et al. [17] recently introduced a novel covariance function, referred to as the rational Szegö kernel. This kernel has been designed to represent a space of complex holomorphic functions and incorporates the Hermitian symmetry inherent in the frequency response of dynamical systems. It effectively captures the Kramers–Kronig relations between real and imaginary parts, resulting in a significantly better fit for complex-valued functions compared to standard kernels such as the Matérn 5/2 kernel. In practice, the convergence of the rational Szegö kernel can be significantly slower than that of rational approximation techniques (such as VF or AAA) when the function exhibits a few dominant poles p_i, i.e., poles with small attenuation $\mathfrak{R}[p_i] \approx 0$. To address this issue, Bect suggested using a low-order rational basis as the mean function $\mathbf{m}(x)$ of the GP.

The combination of kernel methods with a small number of rational basis functions has also been considered in [16]. In that work, the authors employ a superposition of a first-order spline kernel to encode smoothness, along with additional kernels that incorporate prior knowledge about resonating poles. These poles are estimated through a local rational modeling (LRM) preprocessing step. By leveraging these prior knowledge kernels, which concentrate statistical power around the frequencies of the dominant poles, the GP model can effectively quantify uncertainty in its posterior predictions, enabling the definition of error bounds on the estimate.

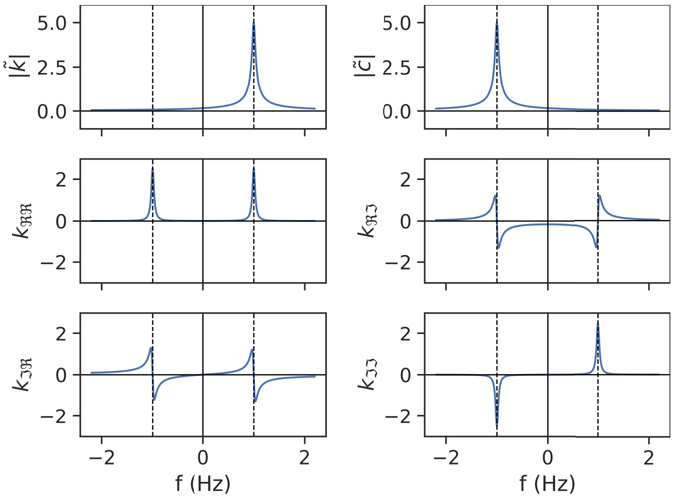


Fig. 1. Magnitude plot of the CGP and corresponding real-valued Szegö covariance functions with $f' = 1$ Hz and $\alpha = 0.1$.

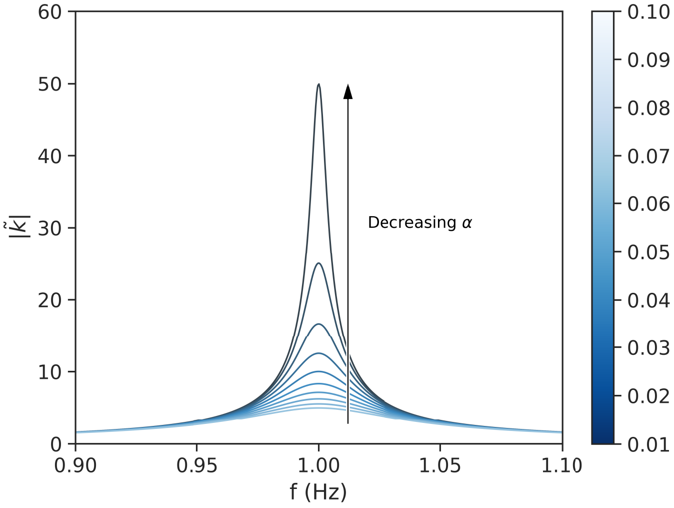


Fig. 2. Magnitude plot of $\tilde{k}(f, f')$ with $f' = 1$ Hz and varying values of α.

Both [16] and [17] introduce kernel-based approaches for frequency-domain system identification. The Szegö kernel, in particular, forms the foundation of this work, as its properties enable the construction of compact time-domain macromodels. The Szegö covariance and pseudocovariance functions are given by

$$\begin{aligned}\tilde{k}_\alpha\left(s, s'\right) &= \frac{1}{2\alpha + s - s'}\\ \tilde{c}_\alpha\left(s, s'\right) &= \frac{1}{2\alpha + s + s'}\end{aligned} \tag{11}$$

where α is a hyperparameter of the rational kernel and $s = j\,2\pi f$ and $s' = j\,2\pi f'$ are the Laplace variables evaluated at different frequencies. Fig. 1 shows a plot of the Szegö covariance functions with $f' = 1\,\text{Hz}$ and $\alpha = 0.1$. The rational kernel operates similarly to traditional kernels such as the Matérn 5/2 or RBF kernel, where points in close proximity to f' exhibit a high correlation, while the correlation rapidly diminishes for more distant points. However, a key distinction from traditional kernels is that points located near $-f'$ remain highly correlated with points at f'. This behavior stems from the (anti)symmetry of the covariance functions with respect to 0 Hz and is a direct implication of the Szegö pseudocovariance function $\tilde{c}$, which enforces Hermitian symmetry in the GP estimate.

Fig. 2 depicts the magnitude of the covariance function $\tilde{k}$ for varying values of α. The hyperparameter α behaves similarly to the length scale parameter in the RBF kernel: smaller values of α result in a kernel that is highly responsive to points near f', with correlations rapidly diminishing for more distant points. This leads to a model that captures local variations more effectively. In contrast, larger values of α extend the kernel's influence, allowing distant points to retain a significant correlation, resulting in a smoother fit and an improved generalization across frequencies.

Following standard kernel design practices, a mixture of Szegö kernels can be adopted to further enhance the GP's flexibility in modeling spatially varying correlation structures in the frequency response data. This enables the model to allocate finer resolution to regions with sharp resonances or rapid variations while maintaining broader smoothing in flatter spectral regions. The proposed rational mixture kernel is given by

$$\begin{aligned}\tilde{k}_{\text{mix}}\left(s, s'\right) &= \sum_{l=1}^{L} \sigma_l^2 \tilde{k}_{\alpha_l}\left(s, s'\right)\\ \tilde{c}_{\text{mix}}\left(s, s'\right) &= \sum_{l=1}^{L} \sigma_l^2 \tilde{c}_{\alpha_l}\left(s, s'\right)\end{aligned} \tag{12}$$

where L is the number of rational Szegö kernels, and σ_l and α_l are learnable hyperparameters [17]. The mixture kernel is highly effective due to its strong generalization properties, but it requires a sufficiently dense set of frequency samples, especially when sharp resonances are present in the S-parameter data.

To mitigate this limitation, a low-order rational mean, as proposed by Bect et al. [17], can be incorporated into the GP. This (optional) method is particularly well-suited for modeling sparsely sampled frequency responses exhibiting sharp resonant peaks, as it can effectively represent the underlying resonant structures. The rational mean, which satisfies the symmetry condition $\tilde{m}^*(s) = \tilde{m}(s^*)$, can be expressed as follows:

$$\tilde{m}\left(s\right) = \sum_{i=1}^{M_\lambda} \frac{\tilde{\rho}_i}{s - \tilde{\lambda}_i} + \rho_0 \tag{13}$$

where $\tilde{\lambda}_i$ are the poles (either real or occurring in complex-conjugate pairs) satisfying the stability conditions $\Re(\tilde{\lambda}_i) < 0$ and $\Im(\tilde{\lambda}_i) \geq 0$ for all i, $\tilde{\rho}_i$ are the residues associated with the poles (also either real or occurring complex-conjugate pairs), and $\rho_0 \in \mathbb{R}$ is a real-valued constant.

Although the theoretical derivations in the following chapter incorporate the rational mixture kernel, all empirical examples in this work employ a rational mean function to model resonant poles. This modeling decision is motivated by two key considerations. First, in noisy settings, kernels with multiple length scales (particularly in systems dominated by sharp resonances) are prone to overfitting. In such cases, the dominant poles effectively force the kernel to adopt smaller length scales, reducing its ability to filter noise. However, by introducing a rational mean function, dominant poles are explicitly represented, allowing the Szegö kernel to focus on smoothing the residual variations. Second, as shown in Section III-A, the use of a mixture kernel often leads to a significantly higher number of poles in the resulting rational representation of the GP.

D. Hyperparameter Learning

The performance of GPs heavily depends on the choice of the kernel hyperparameters, which control the properties of the covariance function, such as smoothness and periodicity. These hyperparameters $\boldsymbol{\theta}$ are typically learned by maximizing the log marginal likelihood of the observed data

$$\log\ p\left(\mathbf{y}|F,\boldsymbol{\theta}\right) = -\frac{1}{2}\mathbf{y}^{\mathrm{T}}K^{-1}\mathbf{y} - \frac{1}{2}\log\ |K| - \frac{N}{2}\log\ 2\pi \tag{14}$$

where K is the covariance matrix constructed from the kernel function. The terms in (14) correspond to the below list.

1) *Data Fit:* The first term measures how well the model fits the data.
2) *Model Complexity Penalty:* The second term penalizes overly complex models by incorporating the determinant of the covariance matrix.
3) *Normalization Constant:* The third term ensures proper scaling of the likelihood.

To find the optimal hyperparameters $\boldsymbol{\theta}$, the negative log marginal likelihood is minimized using numerical optimization techniques such as gradient descent or second-order methods (e.g., L-BFGS). In this work, the hyperparameters of the mixture kernel ($\boldsymbol{\alpha}_{\mathrm{mix}}$, $\boldsymbol{\sigma}_{\mathrm{mix}}$), prior mean ($\tilde{\boldsymbol{\lambda}}$, $\tilde{\boldsymbol{\rho}}$) and likelihood ($\eta$) are optimized using a penalized version of (14), as initially proposed by Bect. Since this falls beyond the scope of our contribution, readers are referred to [17] for further details.

III. KERNEL-AIDED RATIONAL MACROMODELING

This section presents the novel KARMA framework, illustrated in Fig. 3, and outlines its key building blocks. At the core of the proposed framework is the analytical transformation of the GP estimate into an equivalent rational pole–residue representation, which preserves physical properties including stability, causality, and Hermitian symmetry. The structure and properties of the resulting rational model are then exploited to construct a multiport macromodel in which all S-parameter elements share a common set of poles. This is achieved by modeling each element of the S-matrix with a distinct GP, where the hyperparameters of the rational kernels (22) are shared across the different GPs. To reduce the size of the model and facilitate efficient postprocessing, a sparse approximation is introduced. Next, a structure-preserving MOR step is applied to derive a compact system of ODEs while preserving the rational form with a common pole set. Although the rational kernel and MOR stage guarantee stable poles by construction, they do not inherently ensure passivity, a critical property for reliable time-domain simulation. To address this, passivity must be explicitly enforced. Following standard practice, the Hamiltonian matrix approach is proposed to identify regions of passivity violations. Passivity of the model is then enforced by modifying the residues of the rational representation. To quantify uncertainty in the presence of noisy observations, an estimate of the mean squared error (mse) relative to the true system response is derived, along with a set of conservative error bounds designed for worst-case analysis (WCA). Each of these components is detailed in Sections III-A–III-H.

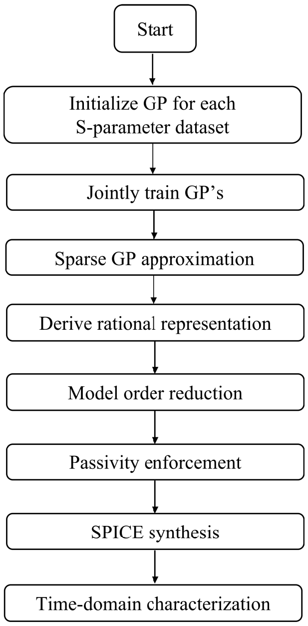


Fig. 3. Flowchart of the KARMA framework.

A. GP Rational Representation

Consider the vector $\mathbf{y} = [\tilde{\mathbf{y}}_{\Re}, \tilde{\mathbf{y}}_{\Im}] \in \mathbb{R}^{2N\times 1}$, which contains the real and imaginary parts of the observed data at the complex frequencies $\mathbf{s} = j2\pi\mathbf{f} = [j2\pi f_1, \ldots, j2\pi f_N]$. By incorporating the rational mixture kernel (12) and rational mean (13) in the CGP, the GP estimate at a new input frequency may be expressed as follows:

$$\hat{\mathcal{H}}\left(s\right) = \mathbf{m}\left(s\right) + K\left(s,\mathbf{s}\right)V \tag{15}$$

where $V \in \mathbb{R}^{2N\times 1}$ is defined as follows:

$$V = \left(K\left(\mathbf{s},\mathbf{s}\right) + \eta^2 I\right)^{-1}\left(\mathbf{y} - \mathbf{m}\left(\mathbf{s}\right)\right). \tag{16}$$

Remark that the Laplace variable S is used instead of the frequency variable f for notational conciseness. Then, by noting that $K(s,\mathbf{s}) \in \mathbb{R}^{2\times 2N}$ has blocks $K_{tt'}(s,\mathbf{s}) = [k_{tt'}(s,s_1),\ldots,k_{tt'}(s,s_N)] \in \mathbb{R}^{N\times 1}$ for $t,t' = \{\Re,\Im\}$, the GP estimate can be formulated as follows:

$$\begin{aligned}\hat{\mathcal{H}}_{\Re}\left(s\right) &= \tilde{m}_{\Re}\left(s\right) + \sum_{n=1}^{N} k_{\Re\Re}\left(s,s_n\right)v_n + \sum_{n=1}^{N} k_{\Re\Im}\left(s,s_n\right)v_{n+N}\\ \hat{\mathcal{H}}_{\Im}\left(s\right) &= \tilde{m}_{\Im}\left(s\right) + \sum_{n=1}^{N} k_{\Im\Re}\left(s,s_n\right)v_n + \sum_{n=1}^{N} k_{\Im\Im}\left(s,s_n\right)v_{n+N}.\end{aligned} \tag{17}$$

Combining the real and imaginary parts, it follows that

$$\begin{aligned}\hat{\mathcal{H}}(s) = \tilde{m}(s) &+ \sum_{n=1}^{N}\left[k_{\mathfrak{R}\mathfrak{R}}(s, s_n) + jk_{\mathfrak{I}\mathfrak{R}}(s, s_n)\right]v_n \\ &+ j\sum_{n=1}^{N}\left[k_{\mathfrak{I}\mathfrak{I}}(s, s_n) - jk_{\mathfrak{R}\mathfrak{I}}(s, s_n)\right]v_{n+N}. \qquad (18)\end{aligned}$$

Using (6), this can be expressed in terms of the Szegö covariance and pseudocovariance functions (12)

$$\begin{aligned}\hat{\mathcal{H}}(s) = \ &\tilde{m}(s) + \frac{1}{2}\sum_{n=1}^{N}\left[\tilde{k}_{\text{mix}}(s, s_n) + \tilde{c}_{\text{mix}}(s, s_n)\right]v_n \\ &+ \frac{j}{2}\sum_{n=1}^{N}\left[\tilde{k}_{\text{mix}}(s, s_n) - \tilde{c}_{\text{mix}}(s, s_n)\right]v_{n+N} \qquad (19)\end{aligned}$$

or

$$\begin{aligned}\hat{\mathcal{H}}(s) = \tilde{m}(s) &+ \frac{1}{2}\sum_{n=1}^{N}\tilde{k}_{\text{mix}}(s, s_n)(v_n + jv_{n+N}) \\ &+ \frac{1}{2}\sum_{n=1}^{N}\tilde{c}_{\text{mix}}(s, s_n)(v_n - jv_{n+N}) \qquad (20)\end{aligned}$$

leading to

$$\begin{aligned}\hat{\mathcal{H}}(s) = \tilde{m}(s) &+ \frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{L}\frac{\sigma_l^2\left(v_n + jv_{n+N}\right)}{2\alpha_l + s - s_n} \\ &+ \frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{L}\frac{\sigma_l^2\left(v_n - jv_{n+N}\right)}{2\alpha_l + s + s_n} \qquad (21)\end{aligned}$$

which can be expressed in a rational pole–residue form (1)

$$\hat{\mathcal{H}}(s) = \ \tilde{m}(s) + \sum_{n=1}^{N}\sum_{l=1}^{L}\left(\frac{\tilde{r}_{l,n}}{s - \tilde{p}_{l,n}} + \frac{\tilde{r}^*_{l,n}}{s - \tilde{p}^*_{l,n}}\right) \qquad (22)$$

where

$$\begin{aligned}\tilde{r}_{l,n} &= \frac{\sigma_l^2\left(v_n + jv_{n+N}\right)}{2} \\ \tilde{p}_{l,n} &= -2\alpha_l + s_n. \qquad (23)\end{aligned}$$

It follows that the rational representation of the GP estimate comprises $M_{\text{GP}} = M_{\text{mix}} + M_{\lambda}$ poles. Among these, $M_{\text{mix}} = 2$ LN poles originate from the rational mixture kernel and are positioned at the sampled frequencies $\{s_n\}_{n=1}^{N}$, while the remaining M_{λ} poles arise from the low-order rational mean.

Unlike VF, which iteratively adjusts pole locations to minimize the LSs error, the poles introduced by the mixture kernel are subject to specific constraints. In particular, the real part of all poles remains constant and is determined by the kernel's length scales $\boldsymbol{\alpha}_{\text{mix}}$, while the imaginary part is fixed at the sampled frequencies. As a consequence, the kernel-based rational model will typically require a larger number of poles compared to VF to achieve similar modeling accuracy. This effect becomes more pronounced when $L > 1$, as the number of poles increases linearly with the number of Szegö kernels. While this does not pose any limitations in the frequency domain, it results in a large system of ODEs, which can be computationally expensive to solve. To mitigate this, an additional MOR step is proposed to enable efficient time-domain simulations.

B. Modeling Multiport S-Parameters

The system identification framework presented in the first part of this article is restricted to single-frequency responses and does not yet support the modeling of LTI systems with multiple inputs and outputs. Therefore, an extension is required to accommodate multi-input, multioutput (MIMO) systems, which are widely used to represent microwave and RF devices. In such networks, each port represents an input/output terminal of the device. The relationship between these ports is characterized using S-parameters, which quantify how signals are reflected or transmitted between ports. These parameters are typically collected in a frequency-dependent matrix, where each element represents the interaction between two ports. The diagonal elements describe the reflection at individual ports, while off-diagonal elements capture the transmission between different ports.

To extend the rational macromodeling framework to multiport devices, an approach similar to VF is applied. By adopting a common pole set to model the various elements in the S-matrix, VF is able to build compact state-space representations. Given the significant advantages of this approach, particularly in the context of time-domain simulations, it is also enforced in the KARMA framework. Exploiting the properties of the GP's rational representation (23), a common pole set can be adopted for all S-parameters. In particular, the real part of the poles is determined by the kernel's hyperparameters $\boldsymbol{\alpha}_{\text{mix}}$, while the imaginary part is determined by the sampled frequencies. This allows to use a separate GP for each S-parameter, where the hyperparameters $\boldsymbol{\alpha}_{\text{mix}}$ are shared across the different GPs and each GP is trained on the same set of input frequencies, as illustrated in Fig. 4. Following this approach, a rational function of the form (1) can be constructed, where the elements of the residue matrices are obtained from the rational representation (23) of the different GPs.

The hyperparameter optimization process involves maximizing the marginal likelihood of the combined models. Assuming that the models are independent but share the length scale hyperparameters $\boldsymbol{\alpha}_{\text{mix}}$, the overall log likelihood is calculated as the sum of the log likelihoods of each individual model

$$\log p\left(\mathbf{y}_1, \ldots, \mathbf{y}_{P^2} \mid \boldsymbol{\phi}\right) = \sum_{i=1}^{P^2}\log p\left(\mathbf{y}_i \mid \boldsymbol{\alpha}_{\text{mix}}, \tilde{\boldsymbol{\lambda}}, \boldsymbol{\phi}_i\right) \qquad (24)$$

where $\mathbf{y}_i$ represents the observed real and imaginary data for the ith GP, and $\boldsymbol{\phi}_i$ denotes the hyperparameters specific to each model i.

C. Sparse Model Approximations

Each sampled frequency in the S-parameter dataset introduces L additional pole-pairs in the GP's rational representation. As a result, the model complexity scales linearly with the number of frequency samples, rather than with the intrinsic dynamics of the system's frequency response. This characteristic is undesirable for several reasons. First, dense sweeps are commonly adopted to improve the model's accuracy in the presence of noise, as they provide redundancy that can help mitigate measurement uncertainties. Second, the

$$\begin{bmatrix}\{f_n, S_{11}(f_n)\}_{n=1}^{N} & \{f_n, S_{12}(f_n)\}_{n=1}^{N}\\ \{f_n, S_{21}(f_n)\}_{n=1}^{N} & \{f_n, S_{22}(f_n)\}_{n=1}^{N}\end{bmatrix} \xrightarrow{\text{model}} \begin{bmatrix}\mathcal{GP}(f;\boldsymbol{\alpha}_{mix},\tilde{\boldsymbol{\lambda}},\boldsymbol{\phi}_1) & \mathcal{GP}(f;\boldsymbol{\alpha}_{mix},\tilde{\boldsymbol{\lambda}},\boldsymbol{\phi}_2)\\ \mathcal{GP}(f;\boldsymbol{\alpha}_{mix},\tilde{\boldsymbol{\lambda}},\boldsymbol{\phi}_3) & \mathcal{GP}(f;\boldsymbol{\alpha}_{mix},\tilde{\boldsymbol{\lambda}},\boldsymbol{\phi}_4)\end{bmatrix} \xrightarrow{\text{convert}} \sum_{i=1}^{M_{GP}} \frac{1}{s-\tilde{p}_i}\begin{bmatrix}\tilde{r}_i^{(1,1)} & \tilde{r}_i^{(1,2)}\\ \tilde{r}_i^{(2,1)} & \tilde{r}_i^{(2,2)}\end{bmatrix}$$

Fig. 4. Transformation of the $2 \times 2S$-matrix data into a multiport rational macromodel using four GPs via the proposed method. The poles $\tilde{p}_i$ in the rational conversion include both those of the kernel (23) and of the rational mean (13).

minimum frequency resolution required for accurate modeling is often not known a priori. In such cases, designers typically perform dense frequency sweeps, either through measurement or EM simulation, to ensure sufficient coverage for reliable model construction.

To mitigate model complexity following initial GP training, a sparse approximation is introduced as a postprocessing step. The corresponding workflow is depicted in Fig. 5. This approach utilizes the concept of inducing points, where the full set of observations is substituted with a smaller, representative set of pseudoobservations, thereby maintaining model accuracy while significantly improving computational efficiency.

1) A GP is first trained on the densely sampled frequency response, resulting in a rational representation with a high number of poles.
2) The learned hyperparameters are then transferred to construct a new GP using a reduced dataset $\{f_n, \mathcal{H}_{\text{ref}}(f_n)\}_{n=1}^{N_s}$, where f_n is uniformly distributed across the bandwidth of interest, and $\mathcal{H}_{\text{ref}}(f_n)$ is sampled from the initial GP.
3) The number of frequency samples is iteratively increased until the error between the approximated GP and the initial dense GP falls below a predefined threshold, ensuring a compact yet accurate model representation.

The initial size of the sparse dataset can be estimated from the optimized length scale α, which represents the correlation length of the data and defines the minimum spacing between samples required for reliable inference. Experimental results indicate that the number of samples necessary for constructing an accurate sparse approximation typically lies within the interval $[\text{BW}_{\text{mod}}/\alpha,\ 2\ \text{BW}_{\text{mod}}/\alpha]$, where BW_{mod} is the electrical modeling bandwidth. To this end, $\text{BW}_{\text{mod}}/\alpha$ is selected as the starting point, and the number of samples is subsequently increased until the desired approximation accuracy is achieved. This strategy enables the construction of a sparse approximation with a minimal number of iterations. Moreover, since the hyperparameters are transferred from the dense model, the computational overhead associated with hyperparameter (re)optimization is minimized, resulting in a highly efficient and computationally fast approach.

The proposed sparse model approximation maintains a high accuracy in a significantly more compact form compared to the initial GP, trained on a dense frequency sweep. Although the presence of the pseudocovariance implies that the kernel is nonstationary, its behavior closely approximates that of a stationary kernel in practice. This observation justifies the use of a uniform distribution of sample points across the frequency range of interest. While more advanced implementations such as variational inference techniques are possible, they are not pursued in this work to maintain simplicity. The sparse approximation is presented here for a single S-parameter, but can be readily extended to a multiport setting by aggregating the modeling error across all S-parameters and jointly retraining the associated models, as described in Section III-B.

The sparse approximation offers several advantages: it is computationally efficient, conceptually elegant, requires no additional libraries beyond those used for the initial GP training, and can be readily extended to multivariate settings. Nevertheless, the order of the sparse model often remains higher than necessary. This arises because the poles and residues of the sparse model do not converge to those of the underlying physical system, but are instead constrained by the sampling process and the rational covariance function. In applications where compact models are critical—such as large-scale interconnected circuits—this limitation can become problematic. To this end, an additional reduction step based on balanced truncation (BT) is presented later in this chapter. Guidelines for when to apply each reduction method, or how to combine them, are also discussed.

D. State-Space Representation

The multiport rational function representation of the GP estimates (21) can be analytically transformed into an equivalent system of ODEs in state-space form as follows;

$$\begin{cases}\dfrac{\partial \mathbf{x}(\mathbf{t})}{\partial t} = \mathbf{A}\mathbf{x}(\mathbf{t}) + \mathbf{B}\mathbf{a}(\mathbf{t})\\ \mathbf{b}(\mathbf{t}) = \mathbf{C}\mathbf{x}(\mathbf{t}) + \mathbf{D}\mathbf{a}(\mathbf{t})\end{cases} \tag{25}$$

where $\mathbf{a}(\mathbf{t}) \in \mathbb{R}^{P\times 1}$ and $\mathbf{b}(\mathbf{t}) \in \mathbb{R}^{P\times 1}$ are the forward and backward traveling waves of the P-port system, $\mathbf{x}(t) \in \mathbb{R}^{v\times 1}$ with $v = MP$ represents the state variables, $\mathbf{A} \in \mathbb{R}^{v\times v}$ is a block-diagonal matrix with $\tilde{p}_i$ at its nonzero entries, $\mathbf{B} \in \mathbb{R}^{v\times P}$ is a matrix that only has zeros or ones, $\mathbf{C} \in \mathbb{R}^{P\times v}$ is formed by horizontally stacking the residue matrices $\mathbf{R}_i$, and $\mathbf{D} \in \mathbb{R}^{P\times P}$ is the same matrix as in (1). The matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$ are called state-space matrices. It is possible to express (1) in terms of the state-space matrices

$$\mathbf{S}(s) = \mathbf{C}(s\mathbf{I}_v - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}. \tag{26}$$

E. Model Order Reduction

The proposed mixture kernel introduces L complex conjugate pole–residue pairs for each sampled frequency f_n. If the number of sampled frequencies N is high, this results in a rational approximation (21) with high pole count. To mitigate this effect, an additional MOR step is motivated to reduce the size of the rational approximation, resulting in a compact state-space representation. MOR refers to a set of techniques aimed at approximating large-scale, high-dimensional systems with simpler, lower dimensional models while preserving key system characteristics such as accuracy and stability.

MOR has been a central topic in design automation for several decades. Among the most widely studied techniques are Krylov subspace methods [26], [27], [28], [29], [30] and BT [31], [32], [33], [34], [35]. Krylov-based approaches are particularly effective for analyzing large-scale systems, but the resulting reduced models are often less compact, retaining more states than necessary. In contrast, BT is better suited for systems with a few hundred states, yielding superior reduced models along with computable bounds on the reduction error. Since model size has a direct impact on simulation performance, especially in the context of large, interconnected circuits, BT is the method of choice in this work.

BT reduces the system order by eliminating states with minimal energy contribution. This is achieved by computing the controllability and observability Gramians and transforming the system into a balanced realization in which these Gramians are equal and diagonal. States associated with smaller Hankel singular values are truncated, yielding guaranteed upper and lower bounds on the H_∞-norm approximation error. In KARMA, an iterative scheme is adopted where the reduced model order is initially selected based on the lower bound and then iteratively increased until some desired accuracy is met. The increment steps are defined based on the range between the error bounds, thus ensuring convergence within a guaranteed minimum number of iterations.

Significant efforts have been devoted to developing MOR techniques that preserve the port-Hamiltonian structure inherent to passive impedance (Z) and admittance (Y) representations. However, to the best of the authors' knowledge, no related efforts have been made to explicitly preserve the rational structure with a common pole set in the state-space system (25). The loss of this structure complicates the derivation of equivalent circuit representations, as many synthesis methods explicitly rely on the availability of the pole–residue form and its associated state-space realization [36], [37]. Furthermore, preserving the reciprocity inherent to passive systems, implicitly enforced by the rational model structure, is critical to ensure physically meaningful reduced models.

To address this limitation, a procedure is proposed that allows the adoption of any standard BT (or Krylov) method, yet preserves the model's rational structure. The rational functions $\hat{\mathcal{H}}_{u,v}(s) = \sum_{i=1}^{M} \tilde{r}_i^{(u,v)}/(s - \tilde{p}_i)$ representing the P-port electrical network are used to construct the state-space realization of a single-input multioutput system. In this configuration, the transmission paths from the single input to each output correspond to the distinct rational functions $\hat{\mathcal{H}}_{u,v}(s)$. The transformation of a two-port system into its corresponding single-input multiple output (SIMO) representation is illustrated in Fig. 6. The resulting state-space matrices are given by

$$\mathbf{A}_{\text{SIMO}} = \begin{bmatrix} \tilde{p}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \tilde{p}_M \end{bmatrix}, \quad \mathbf{B}_{\text{SIMO}} = \mathbf{1}_M$$
$$\mathbf{C}_{\text{SIMO}} = \begin{bmatrix} \tilde{\mathbf{r}}^{(1,1)} \\ \vdots \\ \tilde{\mathbf{r}}^{(P,P)} \end{bmatrix}, \quad \mathbf{D}_{\text{SIMO}} = \begin{bmatrix} D^{(1,1)} \\ \vdots \\ D^{(P,P)} \end{bmatrix}. \tag{27}$$

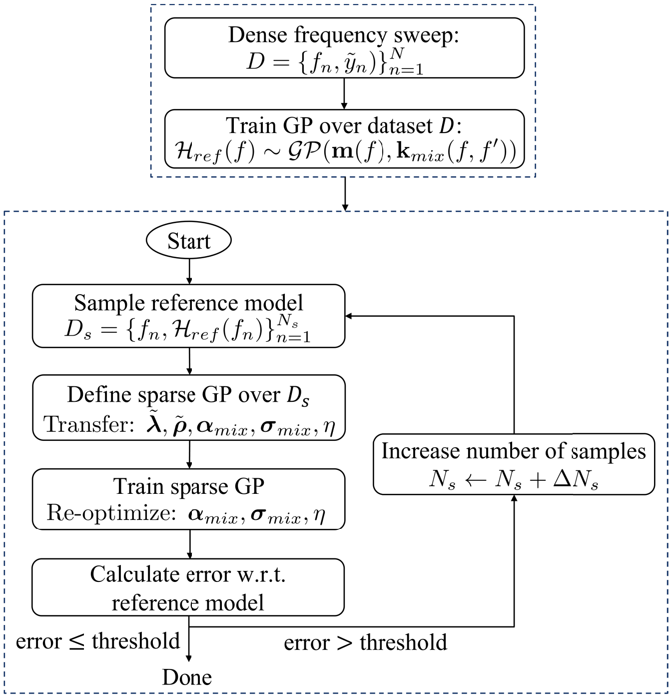


Fig. 5. Flowchart of the proposed sparse model approximation.

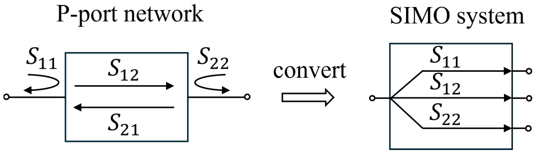


Fig. 6. Transformation of a two-port electrical network into its SIMO representation. Due to the symmetry of the S-parameters, only the upper triangular elements of the S-matrix are considered in the transformation.

The matrix $\mathbf{A}_{\text{SIMO}}$ is a diagonal matrix holding the poles $\tilde{p}_i$ at its nonzero entries. The input matrix $\mathbf{B}_{\text{SIMO}}$ is a column vector of all ones, reflecting the single input structure. The output matrix $\mathbf{C}_{\text{SIMO}}$ is constructed by vertically stacking the residue vectors $\tilde{\mathbf{r}}^{(u,v)} = [\tilde{r}_1^{(u,v)}, \ldots, \tilde{r}_M^{(u,v)}]$, and $\mathbf{D}_{\text{SIMO}}$ contains the corresponding real-valued feedthrough terms. BT is then applied to the system described by (27), yielding an ROM characterized by the matrices $\mathbf{A}_{\text{red}}$, $\mathbf{B}_{\text{red}}$, $\mathbf{C}_{\text{red}}$, and $\mathbf{D}_{\text{red}}$. This step reduces the number of states from M to a smaller dimension M_r.

BT performs a similarity transformation that balances the controllability and observability Gramians, resulting in a transformed system with a loss of the original diagonal structure. Nevertheless, since $\mathbf{A}_{\text{red}}$, $\mathbf{B}_{\text{red}}$, $\mathbf{C}_{\text{red}}$, and $\mathbf{D}_{\text{red}}$ represent a SIMO system, it is possible to extract an equivalent rational representation by diagonalizing $\mathbf{A}_{\text{red}}$. Specifically, there exists a nonsingular matrix $\mathbf{T}$ such that

$$\mathbf{A}'_{\text{red}} = \mathbf{T}^{-1}\mathbf{A}_{\text{red}}\mathbf{T} = \operatorname{diag}\left(\tilde{p}_1^{\text{red}}, \tilde{p}_2^{\text{red}}, \ldots, \tilde{p}_{M_{\text{red}}}^{\text{red}}\right)$$
$$\mathbf{B}'_{\text{red}} = \mathbf{T}^{-1}\mathbf{B}_{\text{red}}, \quad \mathbf{C}'_{\text{red}} = \mathbf{C}_{\text{red}}\mathbf{T}, \quad \mathbf{D}'_{\text{red}} = \mathbf{D}_{\text{red}} \tag{28}$$

where $\tilde{p}_i^{\text{red}}$ are the poles of the reduced system. The residue vectors corresponding to the equivalent rational representation of the SIMO system are then found as follows:

$$\mathbf{C}'_{\text{red}} \circ \begin{bmatrix} \mathbf{B}'^{T}_{\text{red}} \\ \vdots \\ \mathbf{B}'^{T}_{\text{red}} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}^{(1,1)}_{\text{red}} \\ \vdots \\ \tilde{\mathbf{r}}^{(P,P)}_{\text{red}} \end{bmatrix} \tag{29}$$

where $\circ$ denotes the Hadamard (elementwise) product, and $\mathbf{B}'^{T}_{\text{red}}$ is a row vector horizontally repeated l times, with l representing the number of outputs in the SIMO system. The rational representation of the reduced SIMO system is subsequently mapped back to construct a reduced-order model for the original P-port electrical network.

A key limitation of BT lies in its computational complexity, which scales as $\mathcal{O}(n^3)$ with the number of states n in the LTI model. An additional benefit of the proposed approach, however, is that the intermediate SIMO formulation enables MOR on a substantially smaller state-space system. Specifically, instead of reducing the full PM_{GP} states of the original MIMO system, only the M_{GP} states of the SIMO system–corresponding to the common poles of the rational model–must be considered during reduction. This yields significant computational cost savings, particularly for systems with a large number of ports.

In noisy settings, the frequency response is often oversampled to improve noise suppression, which can still result in a high number of poles $M_{\text{GP}} = 2N + M_{\lambda}$. Once the GP with rational mean is trained, however, the kernel is only required to model the smooth residual, which can be represented with far fewer samples. In this case, the optimized length scale α typically satisfies $\alpha \gg \Delta f$, where Δf is the frequency spacing. This implies that a sparse model can be constructed using $N_s \ll N$ samples. Given that $N_s \ll N$, the computational cost of retraining the hyperparameters of the sparse GP is minimal relative to that of the initial GP training. Consequently, for highly oversampled responses, an effective strategy is to first compute a sparse model approximation and subsequently apply the structure-preserving BT method, operating on a reduced number of poles. When the data is undersampled, and the length scale is on the same order as the sample spacing, the sparse approximation does not offer significant computational benefits and BT can be applied directly.

F. Computational Complexity

The dominant computational cost in the KARMA framework arises from GP training. Exact GP inference requires inversion of the covariance matrix of the training data, resulting in a time complexity of $\mathcal{O}(N^3)$ and memory complexity of $\mathcal{O}(N^2)$, where N is the number of training samples. While this cubic scaling places practical limits on dataset size, it remains well suited for problems involving up to a few thousand samples on a standard workstation equipped with 16–32 GB of RAM. Due to the data efficiency and stochastic nature of GPs, this limitation is often not critical, as an accurate model can typically be built with a dataset of moderate size, even in the presence of noise. More advanced solutions such as multi-GPU parallelization and optimized linear algebra routines can push this limit considerably, albeit at the expense of substantial computational resources.

For multiport systems, constructing the complete S-matrix of a P-port network requires joint training of P^2 GPs, thus increasing the overall complexity to $\mathcal{O}(P^2N^3)$. Consequently, the quadratic dependence on the number of ports restricts the applicability of the framework to small- and medium-scale systems, typically with $P < 10$.

To reduce the model order after initial GP training, two distinct approaches were introduced: sparse model approximation and a structure-preserving MOR step. Following the initial hyperparameter training, the computational time required to compute the sparse GP model becomes negligible. Additionally, for the proposed MOR method, the worst-case complexity scales cubically, at $\mathcal{O}(N^3)$, when the data is undersampled. Nevertheless, the time required for model reduction was found to be relatively small compared to hyperparameter optimization, which currently represents the main computational bottleneck of the KARMA framework.

G. Stability and Passivity

When considering passive devices, like RF couplers, filters, or antennas, it is crucial that the equivalent models are both stable and passive to ensure reliable time-domain simulations. If these conditions are not met, the model can be the root cause of nonphysical numerical instabilities during circuit simulations, especially when these circuits would contain feedback loops and/or nonlinear components [1], [38]. A complete discussion (including the derivation) of the conditions for stability and passivity of rational models is given in [1] and [38]. In this section, an overview is provided of the key concepts necessary for evaluating the stability and passivity of the proposed kernel-based rational models.

A state-space model is stable if the real part of all the eigenvalues of the $\mathbf{A}$ matrix is negative. Since the positiveness of α in the Szegö kernel is enforced during MLE, all complex conjugate poles have a negative real part, and stability is preserved by construction.

In [1], the passivity definition and conditions for linear dynamical systems are presented. In particular, there are three passivity constraints that the scattering parameters $\mathbf{S}(s)$ must satisfy.

1) $\mathbf{S}(s)$ is analytic in $\Re(s) > 0$;
2) $\mathbf{I}_n - \mathbf{S}^{\mathrm{T}}(s)\mathbf{S}(s)$ is a nonnegative matrix for all S such that $\Re(s) > 0$; and
3) $\mathbf{S}(s^*) = \mathbf{S}(s)^*$.

If these conditions are satisfied, the S-parameters $\mathbf{S}(s)$ are said to be bounded real. An efficient and accurate method to assess the passivity of state-space models in the form (25) is based on the Hamiltonian matrix $\mathbf{M}$, given by [39] and [40]

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \tag{30}$$

where

$$\mathbf{M}_{11} = \mathbf{A} - \mathbf{B}\mathbf{L}^{-1}\mathbf{D}^{\mathrm{T}}\mathbf{C}$$
$$\mathbf{M}_{12} = -\mathbf{B}\mathbf{L}^{-1}\mathbf{B}^{\mathrm{T}}$$
$$\mathbf{M}_{21} = \mathbf{C}^{\mathrm{T}}\mathbf{Q}^{-1}\mathbf{C}$$

$$\begin{aligned}\mathbf{M}_{22} &= -\mathbf{A}^{\mathrm{T}} + \mathbf{C}^{\mathrm{T}}\mathbf{D}\mathbf{L}^{-1}\mathbf{B}^{\mathrm{T}}\\ \mathbf{L} &= \mathbf{D}^{\mathrm{T}}\mathbf{D} - \mathbf{I}_n, \mathbf{Q} = \mathbf{D}\mathbf{D}^{\mathrm{T}} - \mathbf{I}_n\end{aligned} \tag{31}$$

A state-space model is passive if its Hamiltonian matrix has no purely imaginary eigenvalues, since any imaginary eigenvalue indicates a crossover frequency where a singular value of the S-matrix changes from being smaller to larger than unity, or vice versa [41]. Frequency bands of passivity violations are next identified by checking the singular values at the midpoint between the crossover frequencies [42]. Then, within each violating band, the local maxima can be identified by a frequency sweep [43].

Once violations have been identified, passivity can be enforced as a postprocessing step to ensure numerically stable simulations. A great number of powerful passivity enforcement routines have been presented in the literature and are broadly classified into three categories: 1) direct methods through positive real lemma (PRL) or bounded-real lemma (BRL) constraints [44], [45]; 2) perturbation of the Hamiltonian eigenvalues [40], [46], [47]; and 3) iterative pole–residue perturbation schemes [40], [43], [48], [49], [50], [51]. The shared goal of these methods is to ensure passivity (1) such that the singular values $\sigma(\mathbf{S}(j\omega))$ are unitary bounded for all frequencies, while minimizing the overall change with respect to the original model response. Although each method has its distinct advantages and disadvantages, the method [43] is adopted in this work for its computational efficiency and ability to minimize model deviations within a specified bandwidth.

H. Uncertainty Quantification

Traditional system identification techniques, such as VF, typically assess model accuracy using a hold-out test set. This approach aids in selecting an appropriate trade-off between model complexity and accuracy, while mitigating the risk of overfitting. However, obtaining a separate test dataset often incurs additional experimental cost, and both training and test measurements are generally contaminated by noise. Under such conditions, the empirical test error becomes an unreliable indicator of the model's true accuracy, which should ideally be evaluated with respect to the underlying noiseless system response—information that is fundamentally unobservable. As a result, conventional test-set-based evaluation offers limited insight into true model accuracy in noisy measurement scenarios.

The predictive covariance of the GP quantifies the uncertainty associated with its prediction at a test input f. This covariance can be used to infer the expected squared error between the true function value $\tilde{\mathcal{H}}(f)$ and the GP predictive mean. Specifically, the expected squared error at a test input f can be expressed as follows:

$$\mathbb{E}\left[\left|\tilde{\mathcal{H}}(f) - \hat{\mathcal{H}}(f)\right|^2\right] = \sigma^2_{\Re}(f) + \sigma^2_{\Im}(f) \tag{32}$$

where $\sigma^2_{\Re}(f)$ and $\sigma^2_{\Im}(f)$ are the predictive variances corresponding to the real and imaginary parts, respectively, and appear as the diagonal entries in the predictive covariance matrix. Leveraging these variances, one can infer the expected mse between the GP posterior mean and the true frequency response

$$\mathrm{mse}^{\mathrm{GP}}_{\mathrm{Bayes}} = \frac{1}{N_e}\sum_{n=1}^{N_e}\left(\sigma^2_{\Re}(f_n) + \sigma^2_{\Im}(f_n)\right) \tag{33}$$

where $\{f_n\}_{n=1}^{N_e}$ denotes a set of closely spaced frequency samples over the bandwidth of interest. Additional errors are introduced by subsequent postprocessing such as the sparse approximation, passivity enforcement, and MOR. Although the true mse with respect to the unknown ground truth cannot be directly computed due to the presence of noise in the observations, it is possible to quantify the discrepancy between the GP predictive mean and the final passive reduced circuit model $\hat{\mathcal{H}}^{\mathrm{KARMA}}(f)$, via

$$\mathrm{mse}^{\mathrm{KARMA}}_{\mathrm{post}} = \frac{1}{N_e}\sum_{n=1}^{N_e}\left[\left|\hat{\mathcal{H}}(f_n) - \hat{\mathcal{H}}^{\mathrm{KARMA}}(f_n)\right|^2\right]. \tag{34}$$

Assuming the errors introduced by the postprocessing are independent zero-mean Gaussian variables, the expected mse between the KARMA circuit model and the true frequency response can be approximated by the root sum of squares (RSSs)

$$\widehat{\mathrm{mse}}^{\mathrm{KARMA}}_{\mathrm{true}} \approx \mathrm{mse}^{\mathrm{GP}}_{\mathrm{Bayes}} + \mathrm{mse}^{\mathrm{KARMA}}_{\mathrm{post}}. \tag{35}$$

This metric is adopted to define the error threshold in the sparse approximation loop and to determine the upper and lower bound on the model order during the MOR step. By setting the thresholds in both stages to be equal to the estimated true error $\widehat{\mathrm{mse}}^{\mathrm{KARMA}}_{\mathrm{true}}$, the overall accuracy of the final circuit model is preserved. Moreover, the use of this uncertainty estimate enables automated model order selection and offers a systematic alternative to heuristic order tuning.

The GP predictive variances can also be used to establish a set of error bounds, which are particularly valuable in WCAs. Specifically, pointwise confidence intervals for the squared error of the response can be constructed by scaling the predictive variance using an appropriate standard deviation multiplier. For instance, a 95% confidence upper bound on the squared error is expressed as follows:

$$\mathbb{E}\left[|\tilde{\mathcal{H}}(f) - \hat{\mathcal{H}}^{\mathrm{KARMA}}(f)|^2\right] \leq 4\sigma^2_{\Re}(f) + 4\sigma^2_{\Im}(f) + \mathrm{MAE}^2_{\mathrm{post}} \tag{36}$$

where $\mathrm{MAE}_{\mathrm{post}}$ is the max absolute error over frequency between the GP predictive mean and the passive reduced circuit model. Because $\mathrm{MAE}_{\mathrm{post}}$ represents a worst-case estimate over the entire frequency range rather than a probabilistic measure, this approach generally leads to an overestimation of the actual error introduced during postprocessing. Consequently, the resulting frequency-dependent upper bounds provide conservative estimates of the local squared error.

Within the proposed framework, the noise is represented by a single learnable hyperparameter, assuming uncorrelated Gaussian white noise for the real and imaginary parts. Although treated as uncorrelated, the noise level is assumed to be of the same magnitude for both components across all frequency responses. In practice, GPs can still yield reliable predictions even when the white noise assumption is not strictly valid. In such cases, however, the estimated error variance may not capture the true error at every frequency, so caution is needed when interpreting the GP error bounds. On the other hand, the frequency-averaged GP variance ($\text{mse}_{\text{GP}}^{\text{Bayes}}$) generally offers a reliable measure of overall model accuracy.

Caution must be taken when incorporating a rational mean function (13) into the GP model. In this setting, the uncertainty associated with the rational coefficients is not represented in the predictive covariance. Consequently, the model may exhibit overconfidence, particularly when the coefficients of the rational mean deviate from the true pole locations. This discrepancy leads the GP to underestimate the squared error, resulting in an overly optimistic assessment of model accuracy. While it is theoretically possible to incorporate the contribution of the rational mean function into the covariance and pseudocovariance structures [16], it is beyond the scope of this study and is deferred to future work.

IV. APPLICATION EXAMPLES

The KARMA framework, implemented using GPyTorch, is demonstrated through several application examples. To evaluate performance against a state-of-the-art commercial tool, KARMA is compared with Keysight Broadband SPICE (BBS). Many commercial VF implementations, including Keysight BBS, are primarily developed for clean, simulated EM data; it is worth noting, however, that VF is well known to be sensitive to noise in the training data. To mitigate this issue, several advanced methods have been proposed [9], [10], [11]. While these approaches are effective, they are closed-source and require substantial modifications to the core VF algorithm and implementation. A more recent and simpler alternative augments the VF cost function with a penalty term on the model's second derivative, effectively acting as a regularizer [52]. In this work, KARMA is also compared with the VFIT3 toolbox [53], which has been modified to include this smoothing regularization. Despite its simplicity and effectiveness, this approach requires specifying two hyperparameters: the noise standard deviation and a parameter that controls the smoothness of the VF macromodel. In the experiments presented below, these parameters are either inferred directly from the data or tuned for optimal LSs performance. Across all experiments, both Keysight BBS and the noise-robust VFIT3 implementation were run for 30 iterations with coefficient relaxation enabled to further improve robustness against noise. While the number of poles is manually optimized in VFIT3, the model order is automatically selected in Keysight BBS.

The KARMA framework is evaluated in terms of model order, computational efficiency (CPU time), and accuracy. Model accuracy is assessed using the following mse metrics: $\text{mse}_{\text{train}}$, the error with respect to the noisy training data; mse_{test}, the error with respect to the noisy test data; and mse_{true}, the error with respect to the true (simulated) frequency response. The reported mse metrics are computed on the models prior to passivity enforcement. This ensures that the observed differences in accuracy between KARMA and VF reflect the intrinsic modeling performance rather than artifacts introduced by the passivity enforcement step. The modeling and numerical simulations are carried out on a personal computer equipped with an Intel Core i7 processor and 16 GB of RAM.

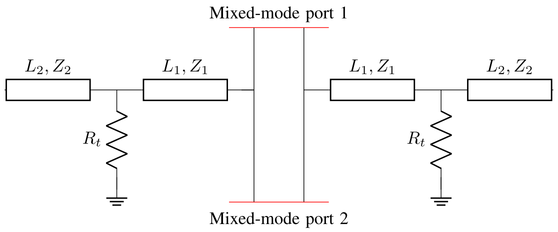
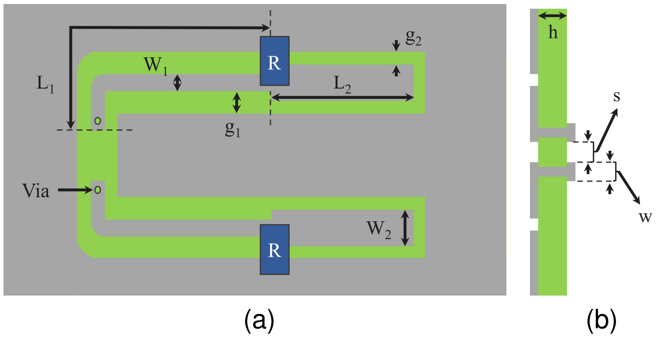


Fig. 7. PDE: Schematic representation of the differential equalizer combining two single-ended equalizers.



(a) (b)

Fig. 8. PDE: Geometry of the PDE. (a) Bottom view of the microstrip layout showing metallization (gray), slots, and dielectric substrate (green). Key features include the open-circuited stubs of length L_2 (the design parameter) and width W_2, transmission-line segments of length L_1 and width W_1, shunt resistors R, and the two via connections. (b) Cross-sectional view of the structure: the left side illustrates the bottom plane with slots, the right side shows two striplines characterized by substrate thickness h, conductor width w, and spacing S. Component dimensions and electrical parameters are listed in Table I.

TABLE I
PDE: DESIGN PARAMETERS OF THE PDE

Parameter	Value	Parameter	Value
W_1	0.35 mm	g_2	0.675 mm
W_2	1.0 mm	R	45 Ω
L_1	18 mm	w	0.9 mm
g_1	1.0 mm	s	0.32 mm
h	1.0 mm	L_2	15.0 mm

A. Passive Differential Equalizer

The KARMA framework is applied to model a passive equalizer embedded in a high-speed differential interconnect, designed to enhance signal integrity (SI) in data transmission systems. At these high data rates, PCB traces exhibit frequency-dependent losses, including the skin effect, substrate losses, and radiation losses. Consequently, higher frequencies are attenuated more than lower frequencies, leading to pulse distortion. To mitigate this effect, Pattyn et al. [54] proposed a passive differential equalizer (PDE) comprising open-circuited stubs and shunt resistors, designed to shape the differential-mode S-parameters to produce a flat magnitude response and linear phase across the signal bandwidth. The equalizer's differential-mode performance is modeled over the frequency

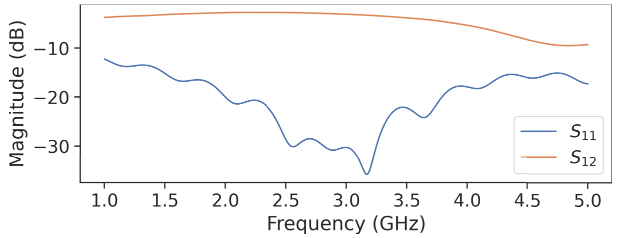


Fig. 9. PDE: *S*-parameters of the passive equalizer simulated in CST Microwave Studio.

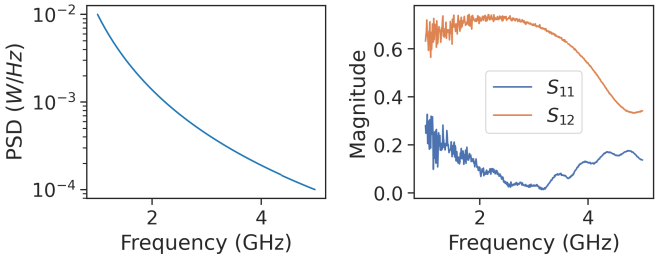


Fig. 10. PDE: (Left) power spectral density (PSD) of the colored noise and (right) corresponding noisy training data.

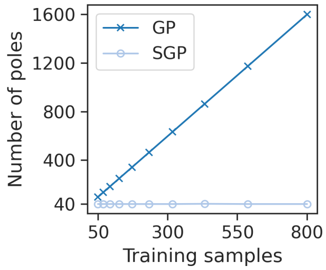


Fig. 11. PDE: Number of poles in the GP and sparse GP rational representations.

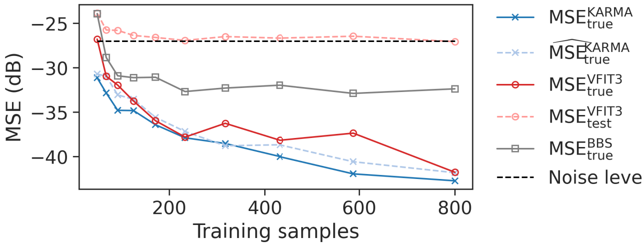


Fig. 12. PDE: mse comparison of KARMA, VFIT3, and BBS under circularly symmetric complex Gaussian noise.

band $\Omega = [1.0, 5.0]$ GHz. A schematic circuit representation of the equalizer is shown in Fig. 7, while its corresponding layout is illustrated in Fig. 8. The equalizer is realized on an FR-4 substrate with a relative permittivity of $\epsilon_r = 3.9$ and a loss tangent of $\tan\delta = 0.022$.

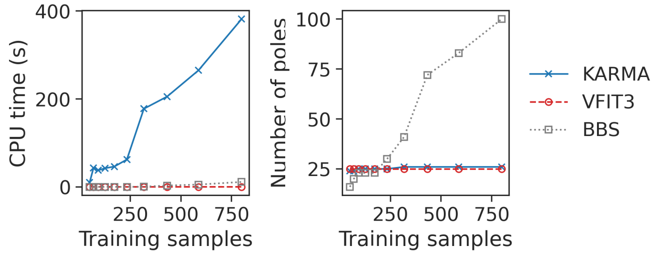


Fig. 13. PDE: Comparison of KARMA, VFIT3, and BBS performance in terms of model order and CPU time.

The frequency response of the equalizer, illustrated in Fig. 9, is simulated using CST Microwave Studio's frequency-domain solver over uniformly spaced frequency points in the range [1.0, 5.0] GHz. In a first experiment, noisy training data are generated by adding circularly symmetric complex Gaussian noise with a variance of 2×10^{-3} to the simulated frequency response. This process is repeated at the same noise level for varying numbers of training samples to assess the performance of the KARMA framework under both sparse and dense data acquisition scenarios. Similarly, in a second experiment, colored noise with spectral profile shown in Fig. 10 is added to the real and imaginary components of the response. To evaluate the model performance, a separate test set of 200 samples over the frequency range of interest is randomly generated using stratified sampling.

In accordance with the KARMA methodology, a compact state-space representation is constructed for each noisy *S*-parameter dataset. Leveraging the symmetry properties of the two-port network, i.e., $S_{11} = S_{22}$ and $S_{12} = S_{21}$, only two GPs are required to represent the equalizer's behavior. Each MOGP incorporates a rational Szegö kernel and a constant prior mean for the real output. A zero mean is adopted for the imaginary part to ensure physically consistent behavior at dc. These GP models are then jointly trained on the artificially generated noisy data of the two-port electrical network, with hyperparameters optimized via the log marginal likelihood, as described in Section II-D. Following training, a sparse approximation is computed as the initial model reduction step. Subsequently, the size of the model is further reduced using BT. All resulting reduced models were inherently passive, and no additional passivity enforcement was required. The error thresholds for the sparse approximation and the MOR step are set to $\widehat{\mathrm{mse}}^{\mathrm{KARMA}}_{\mathrm{true}}$. While this approach enables automated model order selection, it also ensures that the final model maintains accuracy comparable to the original GP representation.

The initial number of poles in the multiport rational representation derived from the GP models follows from the number of frequency samples in the training dataset. Specifically, each frequency sample introduces a complex conjugate pole–residue pair, resulting in a total of $M_{\mathrm{GP}} = 2N$ poles, where N denotes the number of frequency points. This linear relationship is clearly illustrated in Fig. 11. Since computational complexity of BT scales cubically with the number of states, it can become expensive for large datasets. To this end, a sparse model approximation is first computed, which utilizes only 40 poles across all experiments.

KARMA is benchmarked against Keysight BBS and the VFIT3 toolbox enhanced with the smoothing regularization. In the first experiment, the VFIT3 noise hyperparameter was set

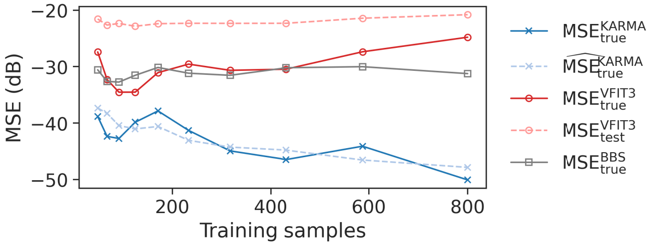


Fig. 14. PDE: mse comparison of KARMA, VFIT3, and BBS under colored noise conditions.

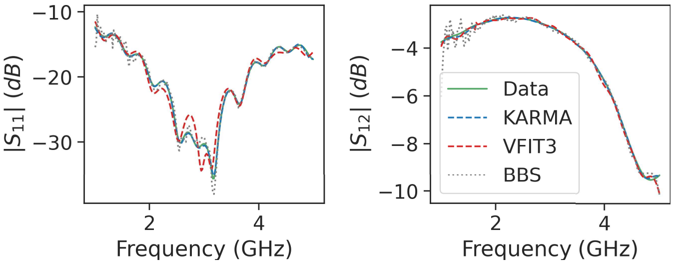


Fig. 15. PDE: *S*-parameter predictions obtained from the KARMA, VFIT3, and BBS models trained on the noisy data shown in Fig. 10, compared with the true system response simulated in CST Microwave Studio.

according to the standard deviation of the circularly symmetric complex Gaussian noise, while the regularization hyperparameter was manually tuned to 0.01 for optimal performance. The results of this experiment are shown in Figs. 11 and 12. Both VFIT3 and KARMA achieve comparable accuracy under white noise conditions. Keysight BBS, on the other hand, exhibits reduced robustness against noise, as evidenced by the nonconverging mse and its inability to properly select the number of poles in the VF model.

The estimated test error, derived from the GP's predictive variance, closely tracks the true test error and decreases as the volume of training data increases. In contrast, VF lacks a stochastic error metric, so model accuracy can only be assessed relative to the noisy measurements. However, this error estimate is fundamentally limited by the noise floor and does not accurately reflect the true model error, as evidenced in Fig. 12.

In terms of computational efficiency, VF outperforms KARMA. This advantage stems from VF's LSs formulation, which is both fast and well-conditioned. In contrast, hyperparameter training in KARMA necessitates the inversion of a Gram matrix (K), whose dimensions scale linearly with the size of the training dataset. Since matrix inversion is a computationally expensive operation, the runtime of KARMA increases as the volume of training data grows, a trend clearly observed in Fig. 13.

The accuracies of VF and KARMA for data contaminated with colored noise are shown in Fig. 14. KARMA yields accurate macromodels with an mse of about −40 dB, even when the dataset size is limited to only 50 samples. Furthermore, even though the white noise assumption no longer holds, the KARMA test error closely follows the true test error. In contrast, both VFIT3 and BBS fail to produce a satisfactory fit. The VFIT3 results in Fig. 14 correspond to a configuration with 25 poles, a noise hyperparameter of 0.01 (taken from the GP-optimized value), and a control hyperparameter of 0.01. Other noise-robust variants of VF may achieve better performance but are not considered in this work.

Fig. 15 presents the *S*-parameter predictions obtained using the VF and KARMA models, each trained on 800 frequency samples contaminated with colored noise. The KARMA model yields smooth predictions that closely match the true, noise-free simulated response across the frequency range of interest. VF, on the other hand, has more difficulty distinguishing noise from system dynamics. While the noise-robust VFIT3 model predicts smooth *S*-parameters, the Keysight BBS model introduces spurious, nonphysical artifacts at the lower end of the frequency range, where the noise variance is highest. This behavior exemplifies a well-documented limitation of the conventional VF approach, where even small amounts of noise can severely impact its convergence.

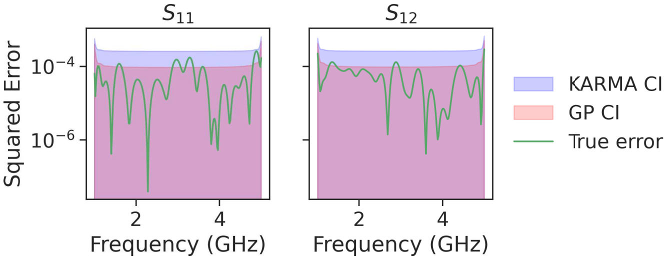


Fig. 16. PDE: The true squared error, the 95% confidence intervals of the GP models, and the corresponding KARMA confidence intervals, for a dataset of 800 samples corrupted by white noise.

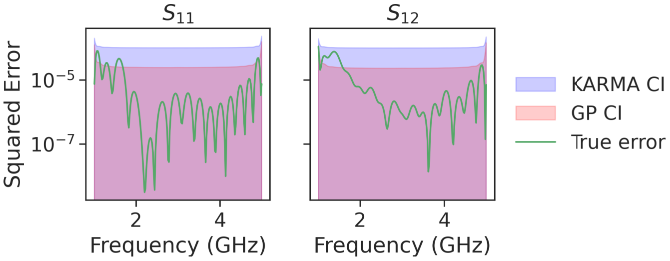


Fig. 17. PDE: The true squared error, the 95% confidence intervals of the GP models, and the corresponding KARMA confidence intervals, for a dataset of 800 samples corrupted by colored noise.

While the expected mse $\widehat{\mathrm{mse}}_{\mathrm{true}}^{\mathrm{KARMA}}$ is a reliable metric for assessing model accuracy, frequency-dependent error bounds can provide additional insight in certain applications. Accordingly, a pointwise upper bound on the squared error, as defined in (36), is derived from the GP predictive variance. For illustration, Fig. 16 shows the 95% confidence bounds of the model trained on 800 white-noise-contaminated frequency samples, along with the actual squared error relative to the simulated data. A similar plot for the model trained on 800 samples corrupted by colored noise is shown in Fig. 17.

In both cases, the KARMA upper bound provides a conservative worst-case estimate, obtained by incorporating the maximum absolute error between the GP model and the ROM. For white noise, the true squared error remains below this threshold across most of the modeling bandwidth, as

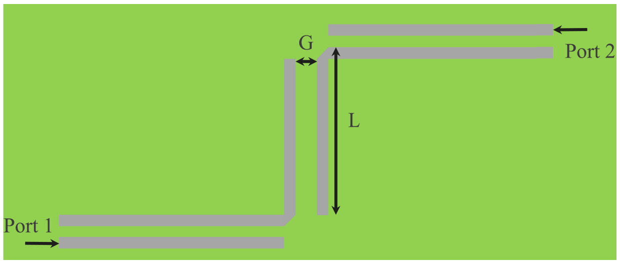


Fig. 18. Zig-zag filter (ZZF): Top-view of the microstrip ZZF layout.

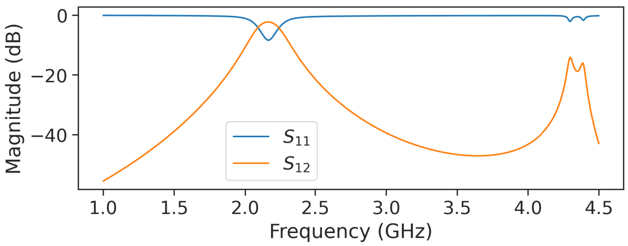


Fig. 19. ZZF: Simulated S-parameter of the microstrip ZZF.

illustrated in Fig. 16. In contrast, for data corrupted by colored noise, the bounds tend to underestimate the error at lower frequencies while overestimating it at higher frequencies. This behavior arises because the GP models the colored noise as an equivalent white noise process, overestimating the error where the true noise variance is low and underestimating it where the noise variance is higher. Moreover, the underestimation of the error is most pronounced near the lower edge of the modeling range, where both the noise variance and the epistemic uncertainty are highest. Nevertheless, despite the violation of the white noise assumption, the bounds still provide a reliable conservative estimate of the maximum true error.

B. Microstrip ZZF

The KARMA framework is tested and benchmarked on a microstrip bandpass filter configuration, as detailed in [55]. The substrate material is 0.5 mm thick, with a relative permittivity of $\varepsilon_r = 2.2$ and a loss tangent of $\tan\delta = 0.0009$. The top-layer geometry of the device under test (DUT) is depicted in Fig. 18. The horizontal conductor gap is 0.3 mm, and all conductors have a uniform width of 0.4 mm, resulting in a narrowband frequency response. The length of the vertical conductors is set to $L = 24.56$ mm, and the gap between them is set to $G = 1.0$ mm. The S-parameters of the DUT, depicted in Fig. 19, are computed using ADS Momentum [56] across 500 uniformly spaced frequency points in the range $f \in [1, 4.5]$ GHz. Among these, 400 uniformly spaced frequency points are selected for training, whereas the remaining 100 points are reserved for testing. Noisy training data are subsequently generated by adding independently distributed Gaussian noise to the real and imaginary components of the simulated data. This process is repeated for multiple noise variances to evaluate the performance of the framework under different noise conditions.

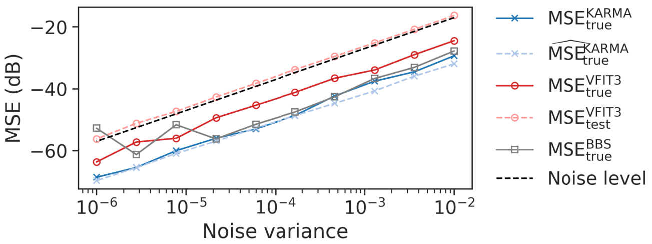


Fig. 20. ZZF: Comparison of KARMA, VFIT3, and BBS performance based on mse metrics.

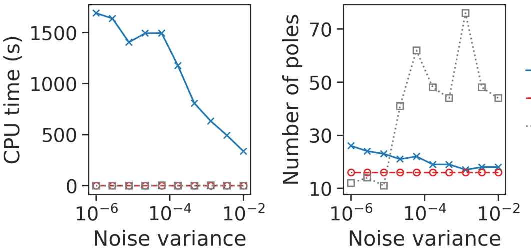


Fig. 21. ZZF: Comparison of KARMA, VFIT3, and BBS performance based on model order and CPU time.

Leveraging the symmetry properties of the two-port network, specifically $S_{11} = S_{22}$ and $S_{12} = S_{21}$, only two GPs are required to represent the filter's behavior. Each GP incorporates a rational Szegö kernel and a low-order rational mean function. Although a statistical pole selection strategy is described in [17], the number of rational poles is fixed to four in this study for consistency. The GP models are trained on the noisy frequency response data of the two-port electrical network, with hyperparameters optimized via the log marginal likelihood, as described in Section II-D. During training, the kernel length scale hyperparameters α and the poles of the rational mean function $\tilde{\lambda}$ are shared across both models to enforce a common pole set. Subsequently, a compact circuit model is synthesized by sequentially applying each step shown in Fig. 3 up to and including the passivity enforcement stage. The performance of KARMA, VFIT3, and BBS is illustrated in Figs. 20 and 21. Both the KARMA and VF macromodels exhibited minor passivity violations. Nevertheless, passivity could be successfully enforced for each model without introducing significant errors relative to the original response within the target frequency range.

As shown in Fig. 20, KARMA and VFIT3 demonstrate comparable accuracy for similar model orders. Although Keysight BBS achieves an mse comparable to that of KARMA, its ability to accurately determine the number of poles deteriorates when the noise variance exceeds 10^{-5}.

In terms of computational efficiency, VF outperforms KARMA in this example. KARMA primarily incurs a higher computational cost due to hyperparameter optimization. In particular, the poles in the GP's rational mean increase the number of hyperparameters, slowing down gradient-based optimization. Furthermore, each iteration of the optimization requires recomputing the residues of the rational mean by solving a linear system, which adds to the overall

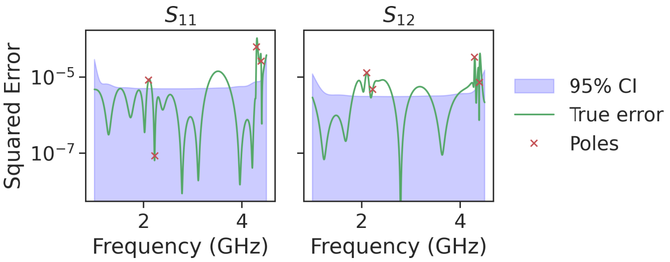


Fig. 22. ZZF: True squared error and 95% confidence intervals of the GP models trained on the dataset with noise variance 6.0×10^{-5}. Red markers indicate the poles of the optimized rational mean function.

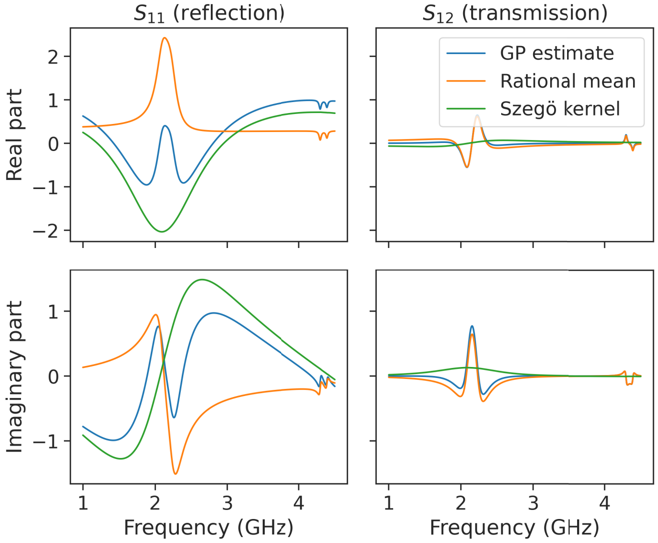
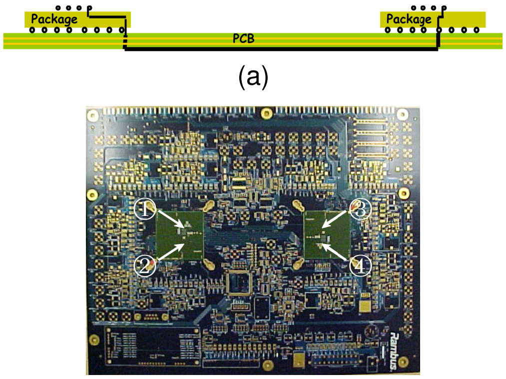


Fig. 23. ZZF: Plot illustrating how the rational mean captures the dominant system poles, while the kernel accounts for the underlying smooth behavior.

computational cost. KARMA is currently implemented within the GPyTorch framework without advanced computational optimizations such as Lazy Tensors or the exploitation of its inherent block covariance structure. Optimizing computational efficiency was not the primary focus of this work but is currently under active investigation. Future improvements leveraging advanced numerical techniques are expected to significantly enhance KARMA's performance, making it more competitive with approaches like VF.

Fig. 22 shows the 95% confidence intervals of the GP models trained on the dataset with noise variance 6.0×10^{-5}. Since the uncertainty in the coefficients of the rational mean is not included in the predictive covariance, the model is overconfident in regions where the rational mean contributes most—i.e., near the pole frequencies. This leads the GP to underestimate the squared error in these areas. As noted in Section III-H, this uncertainty could be incorporated into the covariance structure, but it is not considered here. Nevertheless, as shown in Fig. 20, the estimated mse still closely tracks the true mse, effectively indicating model accuracy.

Fig. 23 illustrates how the rational mean, consisting of four complex conjugate pole pairs, accurately models the resonances around 2 and 4.5 GHz. Since the dominant poles are captured by the rational basis, the kernel models a smoothly varying function. Consequently, the length scale parameter α is assigned a significantly larger value compared to the sample spacing, allowing the model to effectively filter out noise.



(a)

(b)

Fig. 24. Differential interconnect system (DIS): Passive chip-to-chip interconnect system and its scattering parameter measurement setup. (a) Side view of the interconnect. (b) Test board with solder-down transmitter and receiver packages.

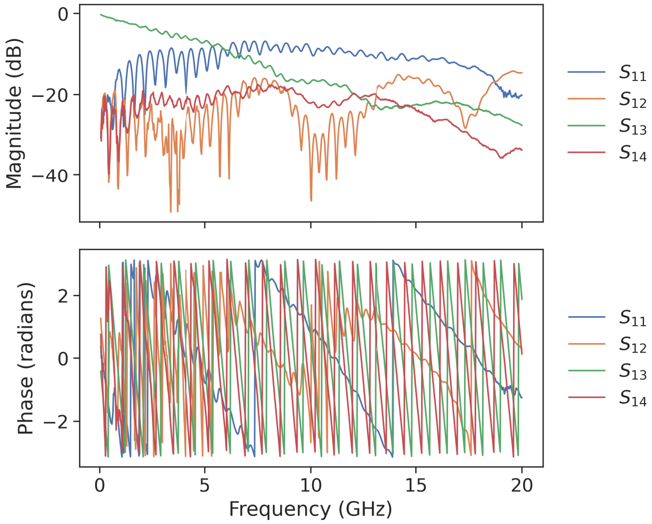


Fig. 25. DIS: Magnitude and phase plots of some of the measured S-parameters.

C. Differential Interconnect System (DIS)

A compact circuit model is computed for the passive differential interconnect structure depicted in Fig. 24. The S-parameters of the four-port electrical device, shown in Fig. 25, were measured using a vector network analyzer (VNA) and are taken from [57]. The magnitude response exhibits dynamic frequency-dependent behavior with numerous resonant peaks across a broad frequency range. Furthermore, the phase response indicates the presence of one or more propagation delays, while the overall frequency response is affected by noise, making this a particularly challenging test case.

The measurement consists of 799 equidistant frequency points over the range $[0, 20]$ GHz. From this dataset, 640 samples are selected for training and the remaining 159 samples are used for validation. Due to measurement noise

TABLE II
DIS: PERFORMANCE COMPARISON OF KARMA AND VF

Method	Model Order	CPU time [s]	MSE [dB]	MAE [dB]
KARMA	166	1236	−47.2	−30.0
VFIT3	170	226	−45.2	−24.2
BBS	149	88	−40.2	−25.8

and other nonidealities, the S-matrix is no longer perfectly symmetric. Although reciprocity could be enforced, e.g., by averaging the upper and lower triangular elements, this step is deliberately omitted. Instead, each S-parameter is modeled using a separate GP.

To ensure physically consistent behavior at dc, the GP models are configured with a zero-mean prior for the imaginary part and a trainable bias for the real part. Once initialized, the models are trained and subsequently transformed into compact circuit representation using the KARMA methodology. As each frequency sample introduces an additional complex conjugate pole pair, the initial rational representation derived from the GP models comprises a total of 1280 poles. Following the sparse approximation step, this number is reduced to 240 poles. Subsequent MOR further compresses the system, resulting in a final rational model with 166 poles.

Leveraging the Bayesian estimate of the mse, i.e., $\mathrm{mse}_{\mathrm{Bayes}}^{\mathrm{GP}}$, along with the BT error bounds, the KARMA model can be computed in a single pass without user intervention. Like KARMA, Keysight BBS automates both model computation and order selection. In contrast, the VFIT3 toolbox requires manual selection of the model order, which in this experiment was performed by iteratively increasing the number of poles from 100 to 200 in steps of 5 and selecting the model that minimizes the training error. Furthermore, both the noise and control hyperparameters of the smoothing regularization were manually optimized and fixed at 0.001. The resulting performance metrics for all three methods are summarized in Table II. It should be noted that the reported accuracies are with respect to the test set.

While all three methods yield a similar number of poles, KARMA achieves the lowest test error in terms of both mse and MAE. An additional benefit of KARMA is its ability to provide a stochastic estimate of the true mse, which offers a direct qualitative measure of model accuracy and can be used to guide iterative data acquisition schemes. For the model under study, this error estimate is $\widehat{\mathrm{mse}}_{\mathrm{true}}^{\mathrm{KARMA}} = -52.3$ dB. In contrast, traditional methods such as VF rely on the test or training error relative to measured data, the accuracy of which is fundamentally limited by noise present in the measurements.

The S-parameter predictions of the two VF models, along with those of the KARMA model, are shown in Fig. 26. By tuning only a small set of hyperparameters, KARMA effectively distinguishes noise from the underlying system dynamics, making it inherently robust to overfitting and resulting in smooth predictions. Both VF models also yield smooth responses in this example; however, larger discrepancies are observed between their predictions and the measured data.

To demonstrate the compatibility of the proposed modeling framework with established circuit synthesis techniques and commercial simulators, the KARMA macromodel is converted into an equivalent electrical netlist through a direct implementation of its state-space representation [1]. A transient simulation is then performed in LTSpice [58] by applying a 5-Gb/s differential on–off keying (OOK) signal to the input terminals of the passive interconnect model. The same transient simulation is performed using an electrical netlist derived from the VFIT3 macromodel. As shown in Fig. 27, the differential output response obtained with both VF and KARMA are in excellent agreement, with a time-averaged relative mse of 4.3×10^{-5}. The CPU time required to simulate the KARMA and VF netlists is approximately 2 s for both, owing to their comparable model orders.

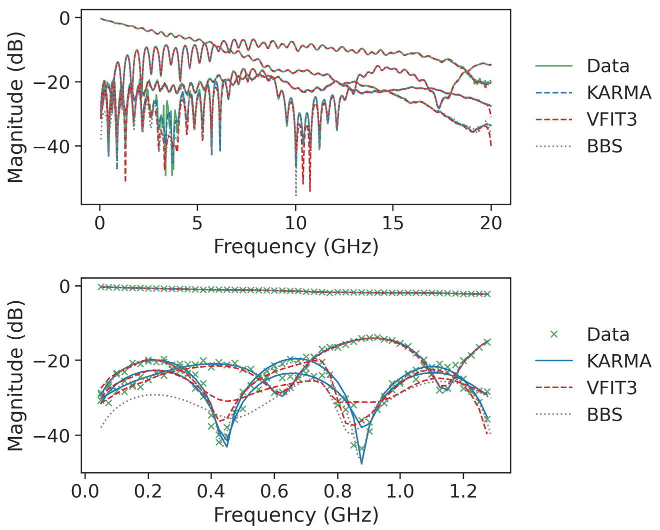


Fig. 26. DIS: S-parameter predictions obtained from the KARMA, VFIT3, and BBS models, compared with measured data over the full modeling bandwidth (top) and in a zoomed-in view over the range [0.0, 1.3] GHz (bottom).

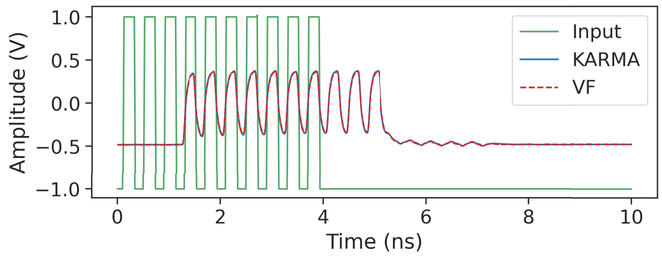


Fig. 27. DIS: Transient SPICE simulation of the interconnect's differential output using the KARMA and VFIT3 models.

This example demonstrates the end-to-end capability of the KARMA framework, transforming S-parameter measurements into compact circuit models suitable for accurate and stable time-domain simulation in standard EDA environments.

V. CONCLUSION

This article introduced KARMA, a novel and unified framework that leverages GPs and rational kernels to construct compact circuit models directly from frequency-domain data. KARMA effectively bridges the gap between statistical, data-driven modeling and classical systems and control theory, combining the advantages of both approaches. Its core innovation lies in the derivation of a rational pole–residue representation from the GP predictive mean, enabling compatibility with traditional circuit simulation workflows. Its probabilistic foundation provides a robust mechanism for assessing model reliability, particularly in scenarios with noise or limited data, offering potential benefits over traditional deterministic methods such as VF. By incorporating features like joint hyperparameter optimization, passivity enforcement, and MOR, KARMA delivers high accuracy and physical consistency with minimal user intervention. Extensive validation on a range of microwave and RF components demonstrates KARMA's superior accuracy and robustness in noisy environments. Moreover, this framework enables accurate and reliable modeling without the need for explicit model order selection, making it highly suitable for complex and noisy datasets.

REFERENCES

[1] S. Grivet-Talocia and B. Gustavsen, *Passive Macromodeling* (Wiley series in microwave and optical engineering). Hoboken, NJ, USA: Wiley, 2016.

[2] J. Becerra, F. Vega, and F. Rachidi, "Extrapolation of a truncated spectrum with Hilbert transform for obtaining causal impulse responses," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 2, pp. 454–460, Apr. 2017, doi: 10.1109/TEMC.2016.2613044.

[3] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Trans. Power Del.*, vol. 14, no. 3, pp. 1052–1061, Jul. 1999, doi: 10.1109/61.772353.

[4] B. Gustavsen, "Improving the pole relocating properties of vector fitting," *IEEE Trans. Power Del.*, vol. 21, no. 3, pp. 1587–1592, Jul. 2006, doi: 10.1109/TPWRD.2005.860281.

[5] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, "Macromodeling of multiport systems using a fast implementation of the vector fitting method," *IEEE Microw. Wireless Compon. Lett.*, vol. 18, no. 6, pp. 383–385, Jun. 2008, doi: 10.1109/LMWC.2008.922585.

[6] D. Deschrijver, B. Gustavsen, and T. Dhaene, "Advancements in iterative methods for rational approximation in the frequency domain," *IEEE Trans. Power Del.*, vol. 22, no. 3, pp. 1633–1642, Jul. 2007, doi: 10.1109/TPWRD.2007.899584.

[7] Y. Nakatsukasa, O. Sète, and L. N. Trefethen, "The AAA algorithm for rational approximation," *SIAM J. Scientific Comput.*, vol. 40, no. 3, pp. A1494–A1522, Jan. 2018, doi: 10.1137/16m1106122.

[8] R. Živanović, "An automatic algorithm in the AAA framework for fitting noisy frequency responses," in *Proc. 9th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Jul. 2023, pp. 1876–1881, doi: 10.1109/codit58514.2023.10284369.

[9] S. Grivet-Talocia and M. Bandinu, "Improving the convergence of vector fitting for equivalent circuit extraction from noisy frequency responses," *IEEE Trans. Electromagn. Compat.*, vol. 48, no. 1, pp. 104–120, Feb. 2006, doi: 10.1109/TEMC.2006.870814.

[10] S. Lefteriu and A. C. Antoulas, "On the convergence of the vector-fitting algorithm," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 4, pp. 1435–1443, Apr. 2013, doi: 10.1109/TMTT.2013.2246526.

[11] A. Beygi and A. Dounavis, "An instrumental variable vector-fitting approach for noisy frequency responses," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 9, pp. 2702–2712, Sep. 2012, doi: 10.1109/TMTT.2012.2206399.

[12] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, Mar. 2014, doi: 10.1016/j.automatica.2014.01.001.

[13] N. Soleimani and R. Trinchero, "Compressed complex-valued least squares support vector machine regression for modeling of the frequency-domain responses of electromagnetic structures," *Electronics*, vol. 11, no. 4, p. 551, Feb. 2022, doi: 10.3390/electronics11040551.

[14] N. Soleimani, R. Trinchero, and F. Canavero, "Vector-valued kernel ridge regression for the modeling of high-speed links," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jul. 2022, pp. 1–4, doi: 10.1109/NEMO51452.2022.10038963.

[15] N. Mohammed, W. Zhou, B. Bahrani, and D. J. Hill, "Support vector machines for predicting the impedance model of inverter-based resources," *IEEE Trans. Power Syst.*, vol. 39, no. 6, pp. 7359–7375, Nov. 2024, doi: 10.1109/TPWRS.2024.3378200.

[16] N. Hallemans, R. Pintelon, B. Joukovsky, D. Peumans, and J. Lataire, "FRF estimation using multiple kernel-based regularisation," *Automatica*, vol. 136, Feb. 2022, Art. no. 110056, doi: 10.1016/j.automatica.2021.110056.

[17] J. Bect, N. Georg, U. Römer, and S. Schöps, "Rational kernel-based interpolation for complex-valued frequency response functions," *SIAM J. Scientific Comput.*, vol. 46, no. 6, pp. A3727–A3755, Dec. 2024, doi: 10.1137/23m1588901.

[18] 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, doi: 10.1109/TEMC.2023.3317917.

[19] T. Ullrick, D. Deschrijver, W. Bogaerts, and T. Dhaene, "Modeling microwave *S*-parameters using frequency-scaled rational Gaussian process kernels," in *Proc. IEEE 33rd Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2024, pp. 1–3, doi: 10.1109/EPEPS61853.2024.10754263.

[20] N. Soleimani and R. Trinchero, "Efficient implementation of the vector-valued kernel ridge regression for the parametric modeling of the frequency-response of a high-speed link," in *Proc. Gen. Assem. Sci. Symp. Int. Union Radio Sci. (URSI GASS)*, Aug. 2023, pp. 1–4, doi: 10.23919/ursigass57860.2023.10265393.

[21] R. Trinchero and F. Canavero, "Machine learning regression techniques for the modeling of complex systems: An overview," *IEEE Electromagn. Compat. Mag.*, vol. 10, no. 4, pp. 71–79, 2021, doi: 10.1109/MEMC.2021.9705310.

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (Adaptive computation and machine learning). Cambridge, MA, USA: MIT Press, 2006.

[23] R. Boloix-Tortosa, J. J. Murillo-Fuentes, F. J. Payán-Somet, and F. Pérez-Cruz, "Complex Gaussian processes for regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5499–5511, Nov. 2018, doi: 10.1109/TNNLS.2018.2805019.

[24] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," in *Proc. NeurIPS*, 2018, pp. 1–7. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/27e8e17134dd7083b050476733207ea1-Abstract.html

[25] A. Matthews et al., "GPflow: A Gaussian process library using tensorflow," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 1299–1304, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-537.html

[26] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. 4, pp. 352–366, Apr. 1990, doi: 10.1109/43.45867.

[27] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 5, pp. 639–649, May 1995, doi: 10.1109/43.384428.

[28] Odabasioglu, Celik, and Pileggi,, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," in *Proc. IEEE Int. Conf. Comput. Aided Design (ICCAD)*, 1997, pp. 58–65, doi: 10.1109/ICCAD.1997.643366.

[29] A. Vandendorpe and P. Van Dooren, "Model reduction of interconnected systems," in *Model Order Reduction: Theory, Research Aspects and Applications*. Berlin, Germany: Springer, 2008, pp. 305–321.

[30] P. J. Heres, D. Deschrijver, W. H. A. Schilders, and T. Dhaene, "Combining Krylov subspace methods and identification-based methods for model order reduction," *Int. J. Numer. Model., Electron. Netw., Devices Fields*, vol. 20, no. 6, pp. 271–282, Nov. 2007, doi: 10.1002/jnm.644.

[31] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Autom. Control*, vol. AC-26, no. 1, pp. 17–32, Feb. 1981, doi: 10.1109/TAC.1981.1102568.

[32] C. Mullis and R. Roberts, "Synthesis of minimum roundoff noise fixed point digital filters," *IEEE Trans. Circuits Syst.*, vol. CS-23, no. 9, pp. 551–562, Sep. 1976, doi: 10.1109/TCS.1976.1084254.

[33] L. Pernebo and L. Silverman, "Model reduction via balanced state space representations," *IEEE Trans. Autom. Control*, vol. AC-27, no. 2, pp. 382–387, Apr. 1982, doi: 10.1109/TAC.1982.1102945.

[34] D. Enns, "Model reduction with balanced realizations: An error bound and a frequency weighted generalization," in *Proc. 23rd IEEE Conf. Decis. Control*, Dec. 1984, pp. 127–132, doi: 10.1109/CDC.1984.272286.

[35] K. Glover, "All optimal hankel-norm approximations of linear multivariable systems and their $L,^{\infty}$ error bounds," *Int. J. Control*, vol. 39, no. 6, pp. 1115–1193, Jun. 1984, doi: 10.1080/00207178408933239.

[36] R. Neumayer, F. Haslinger, A. Stelzer, and R. Weigel, "Synthesis of SPICE-compatible broadband electrical models from n-port scattering parameter data," in *Proc. IEEE Int. Symp. Electromagn. Compat.*, vol. 1, Apr. 2002, pp. 469–474, doi: 10.1109/ISEMC.2002.1032524.

[37] G. Antonini, "Spice equivalent circuits of frequency-domain responses," *IEEE Trans. Electromagn. Compat.*, vol. 45, no. 3, pp. 502–512, Aug. 2003, doi: 10.1109/TEMC.2003.815528.

[38] S. Grivet-Talocia, "On driving non-passive macromodels to instability," *Int. J. Circuit Theory Appl.*, vol. 37, no. 8, pp. 863–886, Oct. 2009, doi: 10.1002/cta.499.

[39] D. Saraswat, R. Achar, and M. S. Nakhla, "Global passivity enforcement algorithm for macromodels of interconnect subnetworks characterized by tabulated data," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 7, pp. 819–832, Jul. 2005, doi: 10.1109/TVLSI.2005.850098.

[40] S. Grivet-Talocia, "Passivity enforcement via perturbation of Hamiltonian matrices," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 9, pp. 1755–1769, Sep. 2004, doi: 10.1109/TCSI.2004.834527.

[41] B. Gustavsen and A. Semlyen, "Fast passivity assessment for *S*-parameter rational models via a half-size test matrix," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 12, pp. 2701–2708, Dec. 2008, doi: 10.1109/TMTT.2008.2007319.

[42] B. Gustavsen, "Passivity enforcement of rational models via modal perturbation," *IEEE Trans. Power Del.*, vol. 23, no. 2, pp. 768–775, Apr. 2008, doi: 10.1109/TPWRD.2008.916764.

[43] B. Gustavsen, "Fast passivity enforcement for *S*-parameter models by perturbation of residue matrix eigenvalues," *IEEE Trans. Adv. Packag.*, vol. 33, no. 1, pp. 257–265, Feb. 2010, doi: 10.1109/TADVP.2008.2010508.

[44] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (Studies in Applied and Numerical Mathematics). Society for Industrial and Applied Mathematics, Jan. 1994.

[45] G. C. Calafiore, A. Chinea, and S. Grivet-Talocia, "Subgradient techniques for passivity enforcement of linear device and interconnect macromodels," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 10, pp. 2990–3003, Oct. 2012, doi: 10.1109/TMTT.2012.2211610.

[46] Z. Ye, L. M. Silveira, and J. Phillips, "Extended Hamiltonian pencil for passivity assessment and enforcement for *S*-parameter systems," in *Proc. Design, Automat. Test in Eur. Conf. Exhib.*, Mar. 2010, pp. 1148–1152, doi: 10.5555/1870926.1871204.

[47] L. Gobbato, A. Chinea, and S. Grivet-Talocia, "A parallel Hamiltonian eigensolver for passivity characterization and enforcement of large interconnect macromodels," in *Proc. Design, Autom. Test Eur.*, Mar. 2011, pp. 1–6, doi: 10.1109/DATE.2011.5763011.

[48] T. Dhaene, D. Deschrijver, and N. Stevens, "Efficient algorithm for passivity enforcement of *S*-parameter-based macromodels," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 2, pp. 415–420, Feb. 2009, doi: 10.1109/TMTT.2008.2011201.

[49] D. Deschrijver and T. Dhaene, "Fast passivity enforcement of *S*-parameter macromodels by pole perturbation," *IEEE Trans. Microw. Theory Techn.*, vol. 57, no. 3, pp. 620–626, Mar. 2009, doi: 10.1109/TMTT.2009.2013309.

[50] A. Lamecki and M. Mrozowski, "Equivalent SPICE circuits with guaranteed passivity from nonpassive models," *IEEE Trans. Microw. Theory Techn.*, vol. 55, no. 3, pp. 526–532, Mar. 2007, doi: 10.1109/TMTT.2006.890520.

[51] Y.-C. Wang et al., "Accurate passivity-enforced macromodeling for RF circuits via iterative zero/pole update based on measurement data," in *Proc. 20th Asia South Pacific Design Autom. Conf.*, Jan. 2015, pp. 441–446, doi: 10.1109/ASPDAC.2015.7059046.

[52] A. Carlucci, A. Zanco, R. Trinchero, and S. Grivet-Talocia, "Vector fitting of noisy frequency responses via smoothing regularization," in *Proc. IEEE 26th Workshop Signal Power Integrity (SPI)*, May 2022, pp. 1–3, doi: 10.1109/SPI54345.2022.9874941.

[53] SINTEF.(2023). *Vector Fitting Toolbox*. [Online]. Available: https://www.sintef.no/en/software/vector-fitting/

[54] T. Pattyn, A. Moerman, M. Huynen, and D. Vande Ginste, "Differential interconnects with integrated equalization and common-mode filtering for broadband signal integrity enhancement in high-speed PAM-4 signaling," in *Proc. IEEE 28th Workshop Signal Power Integrity (SPI)*, May 2024, pp. 1–4, doi: 10.1109/SPI60975.2024.10539219.

[55] D. Puttadilok, D. Eungdamrong, and W. Tanacharoenwat, "A study of narrow-band and compact size microstrip bandpass filters for wireless communications," in *Proc. SICE Annu. Conf.*, Sep. 2007, pp. 1418–1421, doi: 10.1109/sice.2007.4421205.

[56] Keysight.(2024). *ADS Momentum*. [Online]. Available: https://www.keysight.com/us/en/products/software/pathwave-advanced-design-system.html

[57] W. T. Beyene, J. Feng, N. Cheng, and X. Yuan, "Performance analysis and model-to-hardware correlation of multigigahertz parallel bus with transmit pre-emphasis equalization," *IEEE Trans. Microw. Theory Techn.*, vol. 53, no. 11, pp. 3568–3577, Nov. 2005, doi: 10.1109/TMTT.2005.858375.

[58] Analog Devices, Inc., Norwood, MA, USA,. (2023). *LTspice XVII Simulator*. [Online]. Available: https://www.analog.com/en/resources/design-tools-and-calculators/ltspice-simulator.html