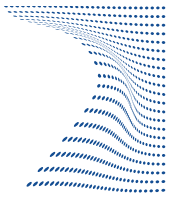



Doctoral Dissertation
Doctoral Program in Electrical, Electronics and Communications Engineering
(34^{th} cycle)

Modeling for the Computer-Aided Design of Long Interconnects

Felipe Treviso
* * * * *

Supervisors
Prof. Flavio G. Canavero, Supervisor
Prof. Riccardo Trinchero, Co-supervisor

Doctoral Examination Committee:
Prof. Christos Christopoulos, University of Nottingham
Prof. Flavia Grassi, Referee, Politecnico di Milano
Prof. Luca Lussardi, Politecnico di Torino
Prof. Paolo Manfredi, Politecnico di Torino
Prof. Giovanni Spagnuolo, Referee, University of Salerno

Politecnico di Torino
November 14, 2022

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . . . .

Felipe Treviso
Turin, November 14, 2022

Summary

The currently used high-speed interconnect structures consist of distributed components such as cables, circuit board strips and package or chip interconnects. Those structures are combined with connectors, vias, etc., resulting in a complex channel for which it is challenging to obtain accurate and compact models so that signal and power integrity (SPI) and electromagnetic interference (EMI) along it can be predicted via simulations. SPI and EMI simulations are desirable because they are cheaper and faster in comparison to experimental alternatives, and therefore, they are an invaluable tool in the early phases of the design of an electronic system.

This thesis discusses techniques suitable for the modeling and simulation of generic long electric interconnects. When taking into account all the electromagnetic effects such as propagation delay, losses, reflections, crosstalk, etc. which are present in those structures, standard models are hard to obtain and/or might have a large complexity which slows down the simulations that must be carried out with them. Such models can be achieved in two different ways:

(i)- Via physical-based models: the level of detail required for an accurate model is not well know. High detailed models are slow to be solved via 3D full-wave simulations, and simplifications might lead to less accurate models, specially when dealing with high-frequency applications. Furthermore, sometimes the detailed design of a component of a high speed link is intellectual property of some companies, and therefore not readily available for engineers which use those parts in their design;

(ii)- Via data-driven modeling approach: in this approach, the use of models without the explicit representation of the delays present in the interconnect structure might lead to models which are excessively complex, requiring a very large number of terms in order to capture the phase variations of systems with delay. On the other hand, if the delay characteristics of the interconnect are explicitly represented, the accurate estimation of the delay values that should be used within the model is very difficult.

The first part of this work follows a physical-based approach to produce a model which predicts the behavior of a complex cable link. It models individually each component of such link, and simulates the complete system in order to achieve the response of the cable link. The modeled system is then validated via experimental data obtained from scattering parameters measurements of the channel, establishing the level of detail needed in those models.

Furthermore, the second part of this work presents a novel approach for the estimation of a surrogate model of a generic long interconnect. The proposed approach is based on a powerful and flexible machine learning technique called the least-square support vector machine (LSSVM). The LSSVM regression is used to construct a metamodel of the transfer function describing a generic linear time-invariant system in a delayed-rational form, but without a-priori specifying the time-delays which should be used by the system, side-stepping this critical part of the estimation of delayed rational models.

By manipulating the estimated metamodel, useful information about it can be extracted. Specifically, the estimated metamodel leads to the accurate estimation of multiple time-delays from the frequency response of the original system. The essential steps and critical criteria for the delay identification procedure are carefully discussed throughout this thesis. By optimizing the LSSVM model via standard techniques used for the tuning of machine learning models, the delays can be searched in a small interval rather than in an extended possible space as is necessary in available techniques for the identification of multiple delays, therefore making this task simpler.

Numerical examples are presented to illustrate the feasibility and performance of the proposed technique and to compare its performances with what is provided by state-of-the-art techniques. The results clearly highlight the capability of the proposed approach to identify the dominant delays in distributed systems, thus allowing to construct compact delayed rational models.

Acknowledgements

I would like to thanks first my supervisors Profs. Riccardo Trinchero and Flavio Canavero for the valuable guidance and effort put into developing this work. I need to thank also all the colleagues from the EMC Group at Politecnico di Torino, and specially Prof. Igor Stievano for first giving me the opportunity to work within this group.

Many people contributed to the successful completion of this work. I would like to acknowledge the help of Fabio Zanoni, Hande Elibol and Enrico Sapienza on the experimental activities performed along these years; Dr. Steve Parkes, from STAR-Dundee, for kindly providing information about the design of their cable used in our experiments; Petri Keski-Opas and Ilkka Kelander, from Huawei Technologies Oy, for providing the opportunity to work on challenging industrial problems.

Also, many thanks for the kind hospitality of Prof. Madhavan Swaminathan for receiving me in the Center for Co-design of Chip, Package, System during my visit at the Georgia Institute of Technology, and all the other colleagues there that welcomed me, sharing their experiences and making the days in Atlanta more enjoyable. This was a very stimulating visit and it was a unique opportunity to learn and enriching for both this work and my personal life.

Finally, I need to thank also my parents Ilso and Noeli, who have to endure the distance between us; my brother, friends and everybody who supported this journey inside the university or outside of it, making the days of these years fun and enjoyable; and my colleagues from the time at UFRGS, in Porto Alegre, who were a key part of the start of all this journey.

“All models are wrong,
but some are useful.”

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Modern electronic systems are composed by many modules that achieve different goals, for example, sensors, displays and processing or data storage units. The data generated by such modules has to be transmitted across them as electric signals. Figure 1.1 exemplifies this transmission. When a signal is transmitted from one device to another, through an long interconnect, it is distorted by the channel and by interference from external electromagnetic noise, while also producing electromagnetic fields that may pose problems to other parts of the system. Therefore, in a good design, the signal and power integrity (SPI) distortions introduced by those effects to the link should be quantified and mitigated, because signal or power integrity issues may lead to unreliable systems, or even worse, systems that fail to operate.

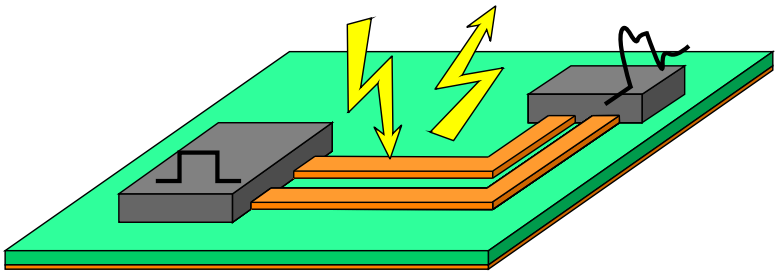


Figure 1.1: Illustrative diagram of the interconnection between two circuital components.

In the simplest way to connect two devices, electric signals travel via physical interconnects. Long electrical interconnects are responsible for a considerable part of signal degradation in high-speed channels [1]. In these long interconnects, signals are subject to propagation effects such as attenuation, ringing, delay, dispersion, reflections and crosstalk, arriving at the receiving end of the channel distorted. However, the length of the interconnect does not refers to its size, but to its electrical length, which is related to the frequency at which it operates. If the physical length is longer than 10% of the wavelength at the highest frequency of interest, this should be treated as a long

interconnect [1]. This effect is highlighted in the case of high-frequency signals. As the frequency and transmission rate of signals increase, their wavelength become shorter, making them behave as electrically long channels even if their physical length is small.

A thorough electromagnetic compatibility (EMC) mitigation and SPI plan will significantly improve the probability of a successful electronic design [2]. Such plan should take into account signal integrity and electromagnetic interference (EMI) along the interconnects. The existence of models that allow those analysis to be carried out through simulations is essential in this regard. With those models, important improvements can be made through optimizations in the design and/or stochastic analysis, all without the requirement of expensive prototypes. But in order to achieve these goals, special attention should be paid to the numerical models used in the analysis. Computer-aided design (CAD) models are logical and accurate, and they can be parametrized by physical quantities of the design, but their simulation requires the numerical solution of electromagnetic (EM) equations, which is very computational intensive, and additionally, they might require a high level of detail which is not always available for intricate structures. On the other hand, data-based black box models seek efficient models that are easily integrated with simulation programs with integrated circuit emphasis (SPICE), but their parametrization is a challenge and their complexity becomes high in the case of long interconnects. Therefore, novel approaches for the modeling of interconnects might present new ways to overcome those issues and should be welcome.

1.1 Modeling of long interconnects

The modeling of long interconnects is an area which has been studied since several decades. The basic structure that models them is the transmission line (TL) [3]. Basically, this structure is governed by the so-called Telegrapher's equations, which are a pair of coupled differential equations that, in the multiconductor case, relate the array of voltages $\mathbf{V}$ and currents $\mathbf{I}$ of the line at any given point x along its length and at any given time instant t as

$$\begin{aligned} \frac{\partial}{\partial x}\mathbf{V}(x,t) &= -\mathbf{L}\frac{\partial}{\partial t}\mathbf{I}(x,t) - \mathbf{R}\,\mathbf{I}(x,t), \\ \frac{\partial}{\partial x}\mathbf{I}(x,t) &= -\mathbf{C}\frac{\partial}{\partial t}\mathbf{V}(x,t) - \mathbf{G}\,\mathbf{V}(x,t). \end{aligned} \tag{1.1}$$

In this structure, the per-unit-length (PUL) resistance $\mathbf{R}$, capacitance $\mathbf{C}$, inductance $\mathbf{L}$ and conductance $\mathbf{G}$ matrices of the multiconductor transmission line (MTL) can be used together with the Telegrapher's equation to obtain an exact model for the known structure, assuming that its cross section is much smaller than its wavelength. Such matrices can be estimated analytically or from numerical simulations accounting for the geometry and material properties of the TL [3].

Circuits composed by structures governed by the equations in (1.1) require their solution in order to have their response analyzed. These equations can be solved more

easily in the frequency domain. In such domain, mode transformations allow their quick solution and incorporation of boundary conditions from the remaining parts of the circuit [3] at any given frequency or set of frequencies. However, most circuits containing these interconnects are comprised by linear components, i.e., the interconnects, capacitances, resistances and inductances, together with non-linear transceivers, drivers, transistors, etc. [4], as shown in Fig. 1.2. Such non-linear components are better simulated in the time domain. This time-domain analysis is achieved via the transient simulation of the system [5]. Therefore, in order to simulate it, the interconnect should be modeled via techniques which allow a mixed time- and frequency-domain representation, since the effects of high-frequency interconnect models are better described in the frequency-domain, but the non-linear terminations of such interconnects (e.g., the link transceivers) can only be accounted via time-domain simulations.

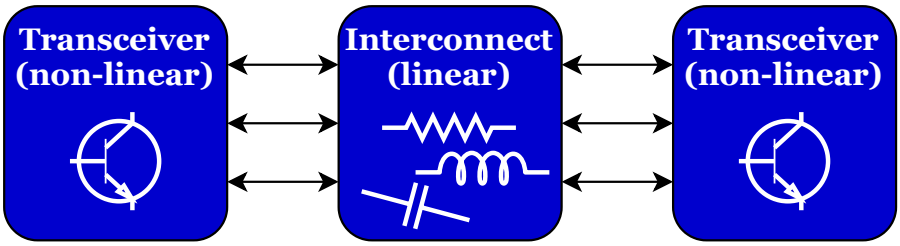


Figure 1.2: Simulation scenario of a system connecting non linear transceivers via a linear interconnect.

Several techniques have been proposed for the transient simulations of circuits with TLs [1], [6]. A straightforward way is by performing the discrete-time convolution [4], where the TL model is appropriately selected so that it can be represented by a causal inverse Fourier series. Such model is an strict discrete-time function for which the convolution equations can be established and integrated with the rest of the system. In a more efficient way, the circuit might be simulated via the waveform relaxation (WR) method [7]. The WR method partitions the structure into smaller subsystems and use waveform iterations to represent the coupling between neighboring subsystems. It allows the solution of each subsystems separately for the entire time-interval, each one with solution parameters optimized for its characteristics. It is possible to use transverse partition to decouple the interconnect channels, and longitudinal partition to separate the channels from the terminations [7]. The longitudinal partitioning can also split the system along the signal path, dividing the TL into smaller parts [8].

The simulations must always include the interconnect model. The simplest approach approximates the TL as a cascade of lumped-circuit [9] components blocks, as shown in Fig. 1.3. The TL is divided in such a way that every block is electrically short at the frequency of interest, usually in π or T-shaped blocks combining the PUL parameters corresponding to a TL section with no more than 10% of the signal wavelength in length [1]. But as the maximum signal frequency increases, the wavelength decreases,

and therefore more and more blocks are necessary in order to accurately representing the simulated line.

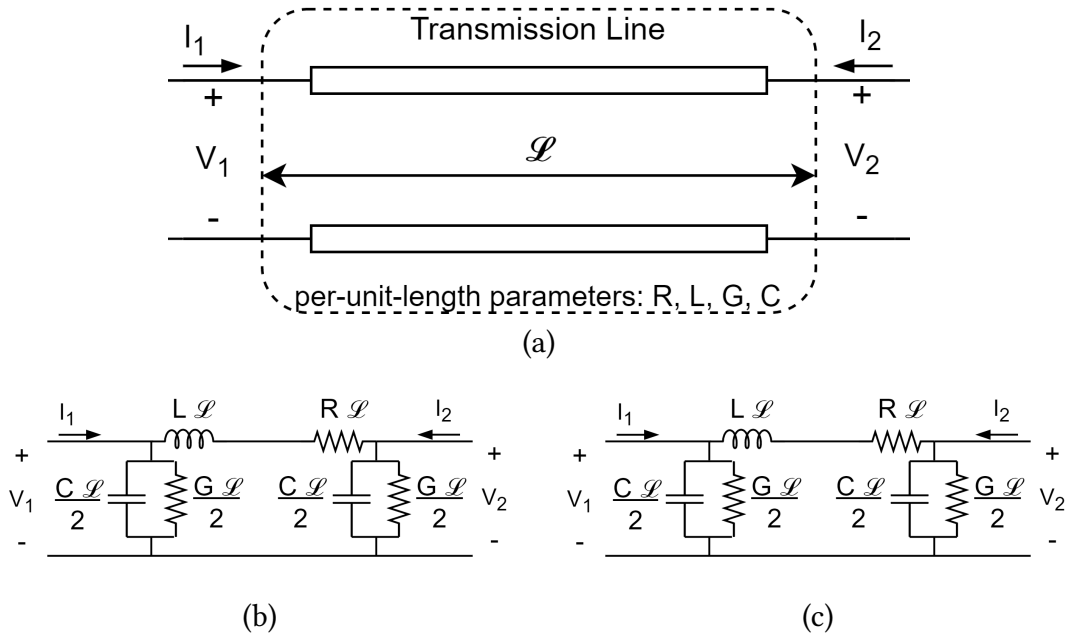


Figure 1.3: Diagram of a transmission line (a) and its lumped segmentation equivalents: π-shaped approximation (b); and T-shaped approximation (c).

An alternative based on the frequency-domain transmission line equations is the generalized method of characteristics (MOC) [10], [11]. It uses mode transformations in order to diagonalize the MTL equations and implement the propagation function for each of the modes. However, this method often assumes that the PUL matrices are constant with frequency [3], which is unrealistic, as the resistance of interconnect increases due to the skin effect, and capacitances and inductances might change due to the use of materials with frequency-dependent properties. The frequency variation of the PUL matrices leads to frequency dependent mode transformations, which are undesirable in time-domain simulations and thus hinders its direct utilization [3], [12].

A model that deserves attention is the asymptotic waveform evaluation (AWE) [13]. This technique uses Padé approximations to match the moments (i.e., the derivatives of the frequency domain transfer function) of TL model matrices. The Padé approximation is defined as the best approximation provided by a rational function of a given order. It is possible to directly match $2q-1$ moments and initial condidions of the TL moments via a rational model with q poles [14]. The computational efficiency of this technique can be improved by using recursive formulas to compute the employed moment matrices and by expanding the TL matrices as a Taylor series [15]. The same AWE-based approach

can be employed to the MOC formulation to extend it to model complicated frequency-dependent PUL matrices in a simple way [16]. The Padé approximations above can also be used to approximate expressions of exponential matrices describing the solution of Telegrapher's equations [17]. The coefficients can be computed analytically from the TL solution, preserving the passivity of the interconnect link, for constant PUL parameters. In the case of frequency dependent PUL parameters, they can be modeled via positive and real functions [18].

The most advanced circuit simulation tools provide the W-element model [19] for TL simulations. This model uses a frequency-domain approximation method based on the interpolation of the PUL matrices and a matrix delay separation. For such approximation, numerical integration formulas are available, simplifying the simulation procedure. It avoids the use of frequency-dependent modal transformations by computing the matrix delay separately from the propagation functions [19], and is successfully employed for the simulation of TLs. A test case in which the W-element model is employed is presented in chapter 2. However, as it will be presented, the simulated structures might be complicated, a situation which requires both the knowledge of the simulated structure and attention to model even apparently insignificant details. Additionally, most real systems cannot be approximated by these TL solutions, as they consist on complex structures that cannot be exactly modeled via homogeneous and uniform TLs.

Black-box models avoid these issues. They are obtained from data representing the impulse response of the interconnect, obtained via full-wave simulations, if possible, or via measurements of the structure. As long as the data is available, the knowledge of the internal components of the structure is not necessary. A successful approach models data from a frequency response by using rational approximations [20], which can then be translated into an equivalent circuit comprised by resistors, capacitors and inductors [12], [21], [22]. The most successful among these techniques is the so-called vector fitting (VF) [12], [23], which via an iterative approach estimates an appropriate set of poles to be used in the approximation, in a very robust way. Based on that, cables can be modeled as a multiport admittance matrix, with data obtained separately for high frequency components and DC components [24]. The VF technique can be used to interpolate the frequency response, and after the passivity of the model is enforced, a SPICE equivalent circuit is extracted and used to predict EMI of the cable link. The complexity of the model can be reduced by using a complex-valued VF in order to approximate baseband models [25]. This technique preserves the advantages of VF, while being able to reduce in half the complexity of the estimated models. With those models, simulators should support complex valued signals and matrices, or in the case this support is not available, it is possible to process the obtained models and achieve real valued models compatible with SPICE [25].

Nonetheless, when the propagation delay between ports of the system becomes large, its frequency response presents a phase that continuously changes and thus it requires a rational model with many terms in order to achieve a satisfactory accuracy.

In the case of these long interconnects, a more efficient approach is the use of delayed rational models (DRMs) [26]–[28]. Such model explicitly represents the delays involved in the signal propagation in the distributed structures, simplifying the rational part of the model that then only need to model the signal attenuation. The equivalent circuit of this model includes also ideal and lossless TLs, which are simple to simulate, and allow a reduction in the number of required capacitor and inductors, i.e., a reduction in the number of dynamical elements. Overall, the use of DRMs for the simulation of long interconnects should lead to faster simulation times and a smaller memory requirement.

The estimation of the time-delays embedded in such models is critical. In interconnects which can be represented via a RC network, the approximate propagation delays can be estimated analytically from the combination of the network parameters plus the terminations. The obtained delay is often called Elmore delay, and it can also extended to the case of interconnects represented via RLC networks [29], [30]. The MTL-based techniques described above (e.g., the MOC) also have delays embedded into their propagation functions. Such models might be extended to the case where the interconnect consists of multiple cascaded TLs and discontinuities between them [31]. To arrive at such model, a DRM is obtained via a matrix pencil technique in which the solution of a generalized eigenvalue problem is used to estimate the system poles.

In a more generic structure, but for an interconnect with a single time-delay, its value in the DRM can be directly optimized given that a reasonable search interval is defined [32]. If we consider a transmission line, this interval can come from the lossless time-delay in the propagation equation [32], [33] or it can be estimated via the magnitude of the frequency response, by considering that when its delay is removed, it should be modeled via a minimum phase shift function [34]. In this way, the phase angle of the frequency response can be calculated directly from its magnitude [35]. The Hilbert transform provides a systematic way to find the dominant propagation delay in a frequency response based on its causality property [36]–[38]. This method is described in section 4.3.1. A similar causality-based approach for delay estimation is based on singular value decomposition (SVD) of causal Fourier continuations [39], where a linear-phase due to the time-delay is introduced into the system of equations which define the Fourier coefficients, from which the delay is extracted. Such delay can be employed to generate a simple, compact and guaranteed causal DRM adequate for the simulation of the structure that generated such frequency response.

However, a generic interconnect with unmatched terminations, as is often the case, will observe multiple delays between their ports due to reflections at the terminations. One way to account for multiple delays is by considering a single propagation delay, but inserting also a positive feedback loop which produce multiple periodic reflections in the system [40], [41]. The multitude of delays present in an interconnect is amplified in the case where internal discontinuities are present along the interconnect, and therefore, additional non-periodic reflections and delays are produced also along the transmission path. In this case, the estimation of multiple delays from frequency domain data is a difficult task with the available techniques. Time-domain approaches can

analyze the system response to an impulse or band limited pulse and identify suitable partition points corresponding to the system delays [42], but the continuity among the regions must be carefully checked [43]. In the case of frequency-domain data, the most solid approach is reliant on time-frequency decompositions based on wavelet transforms such as the Gabor transform [28]. This technique is described in section 4.3.2. However, the different characteristics of the signal attenuation for each delay term, and which is naturally dependent on the delay value, are not captured by this transform. Large delays should observe attenuation at lower frequencies when compared to small delays, as a longer delay means that the signal traveled for a longer distance within the interconnect, and thus suffered more attenuation. Another difficulty lies on the definition of the number of delays that should be included in the DRM [41], which will affect its accuracy and the convergence of the model estimation techniques.

Once the time-delays are identified, the rational parts of the DRM are easily estimated if the original transfer function can be separated into the sum of delayed transfer functions. In this ideal scenario, each part can be fitted individually after removing its individual delay, with any fitting method suitable for lumped structures: the vector fitting [28], the Loewner matrix framework [44], [45], time-domain vector fitting [43] and Prony's method [27]. A previous determination of a minimum phase response of the separated transfer functions using the Hilbert's transform method can be applied before the rational approximation is performed [26], [37], in order to estimate the correct delay that should be used for the respective part of the transfer function.

Green's functions can be used for the analysis of a MTL, achieving a DRM for its multiport impedance representation [46]. In this model, the delays are explicitly extracted, and the computation of poles and residues is done separately for each delay mode. The above technique can be extended to evaluate EMI arising from the coupling of a plane-wave to the MTL [47]. In a structure more complex than a MTL, EMI can be predicted via a DRM estimated using the delayed vector fitting (DVF) algorithm [48], [49]. This algorithm considers a common set of poles for each delay, and through iterations estimate such poles in a similar way as the VF algorithm.

The DRMs above can be extended to the case where some design parameters are accounted in the model, producing a parametric model [50]. Additional interconnect effects such as dispersion and overlapping echoes can be accounted by adding delays to the denominator of the DRM [51], [52]. When inserted into circuit simulations, the DRMs will give rise to time-delay systems. Model order reduction (MOR) approaches can be used to reduce the model complexity and further speed up such simulations [53]. A possible MOR technique suitable for time-delay systems relies on the construction of a Krylov space and the projection of the system on this space [54], and by approximation of the time-delays via a recursive relation of Laguerre polynomials [55]. Alternatively, the time-delay MOR can be constructed by selecting interpolation points with a iterative greedy algorithm in order to control the target error threshold.

A long list of techniques suitable for the modeling and simulation of long interconnects was presented above, each of them with its pros and cons. While TL-based

techniques usually work well, they are not suitable for systems which cannot be easily represented by a TL structure. On the other hand, black-box techniques can handle any kind of system, but they might produce complex systems which are slow to simulate, if the do not account for the propagation delay of the structure. When such delay (or delays) are accounted, the model estimation still poses some challenges, requiring high precision delay values to produce accurate models. A rather unexplored area is the use of data-driven machine learning (ML) algorithms for the modeling of electric interconnects. Such techniques have experienced a significant growth in several areas, e.g., image recognition and signal processing, with great success. However, their application to the modeling of interconnects deserve further investigation, and in order to do that, the available techniques should be studied.

1.2 Machine Learning Regression Techniques

Machine learning refers to a large group of data-driven modeling techniques which can be used to obtain surrogate models for black-box systems. Some of the main ones are briefly described below: the artificial neural networks (ANN), the support vector machine (SVM) regression, the least-squares support vector machine (LSSVM) regression and the Gaussian process regression (GPR).

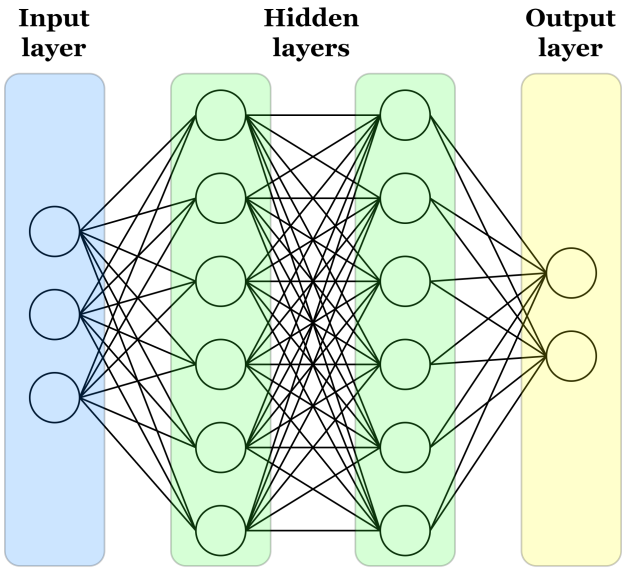


Figure 1.4: Illustrative diagram of the architecture of a two-hidden-layer artificial neural network.

Artificial neural networks are a model inspired by the human brain, in which simple elements called neurons operate in parallel, arranged in a layered structure. Each neuron applies an activation function to its input. The output of the model is then given by

the connections between the neurons. The training of an ANN is given by the estimation of the connection weights between the elements [56]. Figure 1.4 shows an example of the structure of an ANN. It has an input layer, which receives the input variables of the system, a number of hidden layers, and an output layer which takes the outputs of the last hidden layer and produces one or more output variables. The neurons making up each hidden layer are defined by non-linear activation functions, while the activation function of the output layer is generally linear. Typically, the training of the ANN is performed by optimizing the connection weights between the neurons using standard gradient-based back-propagation techniques. The description of the long list of "flavors" available for the ANN topology is out of the scope of this thesis. The reader is referred to [57] for further details.

The SVM and LSSVM regressions provide a simpler input-output relation than the ANN. Instead of using a deep cascade of activation functions and connection weights as the ANN, such regressions are formulated as a simple linear combination. However, instead of performing the regression in the input parameter space, where the linear input-output relation is not certain, it projects the data into a possibly higher dimensional feature space, where it can become closer to linear [58]. Figure 1.5 tries to exemplify such transformation. Additionally, one of the main advantages of the SVM is that this projection can be performed implicitly, via the use of a kernel function, for which many successful options are already tested with great success [59].

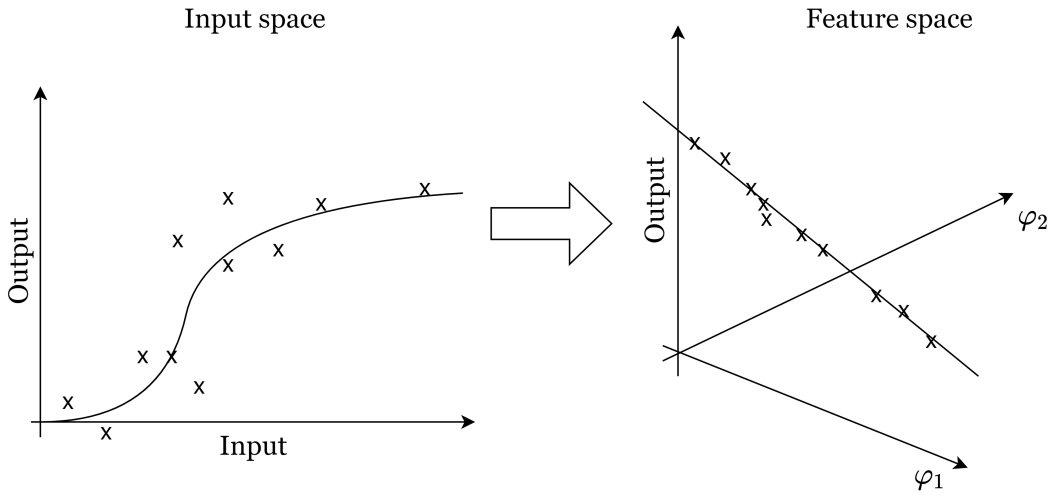


Figure 1.5: Diagram showing the transformation provided by the kernel function from a low-dimensional input space to a higher-dimensional feature space.

While the ANN can adjust its objective function to minimize any performance metric, the SVM is formulated with a specific loss function in mind. It uses a so-called ε-intensive zone, shown in green in Fig. 1.6. All the points predicted by the model inside this zone with width equal to 2ε do not add any penalty to the loss function. Only

the terms outside of this zone, which are called support vectors, add a penalty to the loss function equal to the square of its distance to the ε-intensive zone. This loss function plus a regularization term used to assign good generalization properties to the model are minimized via a convex optimization problem. The approximate model is then written as a function of such support vectors. The LSSVM can be seen as a simplification of the SVM, in which the ε-intensive zone is equal to zero [60]. Such simplification leads to a model which can be estimated via the solution of a linear system, instead of an optimization problem, and in which all training points are also support vectors. On one hand, this increases the power of the method in predicting highly non-linear outputs, but on the other hand, its prediction time might be slower, as the number of support vectors is larger. Nonetheless, the prediction time of a SVM model is usually negligible, and the LSSVM is successfully applied in many kinds of regression problems [61].

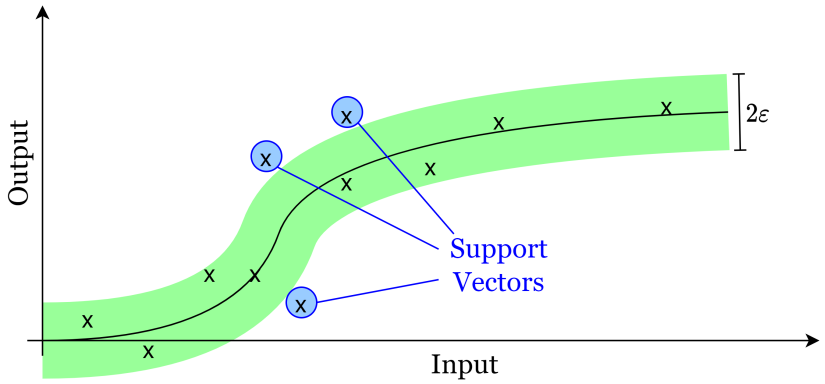


Figure 1.6: Illustration of support vector machine regression, with the indication of the ε-intensive region and the support vectors.

All the techniques above provide as a result a deterministic model, i.e., a surrogate model for which a deterministic function is able to predict the output variable of interest as a function of the input parameters. They are limited in the sense that they do not provide information about the reliability of their prediction. The model error is known for the available samples, but it is unpredictable for any other generic point in the input space. The Gaussian process regression (or Kriging model), on the other hand, provides a probabilistic model [62], which together with a prediction for the desired output, estimates also useful statistics regarding that prediction. In order to do this, it is defined via a mean function, providing its expected output, but also a correlation function that provides the correlation between the output at different points in the input space. Both of those functions should be specified when training the GPR model.

We can summarize the main points of the techniques above by saying that usually the ANN provides the biggest potential in modeling a generic function, as additional layers with distinct shape and functions can be added to its structure. However, it also has thousands of coefficients that should be estimated, and for such, they require a

large amount of data reserved for its training. The SVM and LSSVM regressions, on the other hand, should provide good models with less training data, by implicitly using a high dimensional feature space with possibly an infinite number of dimensions. The LSSVM regression is specially useful by having the simpler formulation and solution. The GPR, on the other hand, has the main advantage of providing a probabilistic model output. Additional machine learning are available, and the reader is referred to [57] for additional information about them.

1.3 Goal of this thesis

Considering the literature review from section 1.1 and the machine learning techniques in section 1.2, this thesis aims at contributing with alternative ways for the modeling of generic long interconnects. When taking into account all the electromagnetic effects such as propagation delay, losses, reflections, crosstalk, etc. which are present in those structures, standard models are hard to obtain and/or might have a large complexity which slows down the simulations that must be carried out with them. Such models can be achieved in two different ways:

(i) Via physical-based models: the level of detail required for an accurate model is not well known. High detailed models are slow to be solved via 3D full-wave simulations, and simplifications might lead less accurate models, specially when dealing with high-frequency applications. Furthermore, sometimes the detailed design of a component of a high speed link is intellectual property of some companies, and therefore not readily available for engineers which use those parts in their design;

(ii) Via data-driven modeling approach: in this approach, the use of models without the explicit representation of the delays present in the interconnect structure might lead to models which are excessively complex, requiring a very large number of terms in order to capture the phase variations of systems with delay. On the other hand, if the delay characteristics of the interconnect are explicitly represented, the accurate estimation of the delay values that should be used within the model is very difficult.

This thesis tackles the corresponding issues in two different ways:

(i) By modeling the interconnecting channel piece-by-piece based on its physical dimensions and material properties, the necessary details of the channel are added and the complete model is experimentally validated by comparing its response to experimental data;

(ii) By building a data-driven metamodel which implicitly takes into account the delay characteristics which are intrinsic of long interconnects, along with the other effects present in passive and linear electric structures.

The data-driven metamodel above is based on the least-squares support vector machine (LSSVM) regression. Specifically, such technique allows the construction of non-parametric models which embed all their complexity within its kernel function specified during the training phase. The appropriate selection of this kernel function leads to accurate models which scale well for predictions in high dimensional parameter spaces [61], [63]. The technique presented in this thesis builds a DRM without the need of specifying a finite number of delays, by estimating a model that contains any delay value within a certain interval, together with other characteristics present in a generic interconnect, e.g., losses and attenuation along the transmission channel. This is done via adjusting the kernel function so that it provides all those specified characteristics. These models can be translated into fast and efficient simulation blocks for large system simulations, as shown in Fig. 1.2, and/or they could be used to include additional design parameters into the model, as can be done with CAD models. Above all, the analysis of the final model provides an insight about the system that originated the data, and via this analysis this thesis also brings a new way to identify key parameters of the model, e.g., the multiple delays present in a frequency-domain response of an interconnect. Furthermore, despite being based only on data, the kernel-based techniques have a dual representation that preserves the knowledge about its model structure. This is an advantage in contrast to other powerful machine learning (ML) techniques such as the artificial neural networks [64], for which the final model might become highly complex, hindering its understanding.

1.4 Organization of the text

The organization of this work goes from this lengthy textual introduction in chapter 1 to a conclusion in chapter 5. Those final remarks summarize what was done and outline future steps that should to be performed for further improving the presented techniques, along to new possibilities that might achieve good results using ideas similar to this work. Along the way, in chapter 2, some of the physical-based alternatives combined with a black-box model demonstrate the issues of a traditional modeling workflow for the simulation and experimental validation of a realistic cable link structure. Then, in chapter 3, a kernel-based black-box model is presented and adapted to be used with complex-valued data, in order to be used with the target data of this thesis. The core of this work comes in chapter 4. There, a specific kernel is developed in order to be used to estimate a surrogate model based on the frequency response of long interconnects. Such kernel replicates a DRM with an infinite number of delays. The analysis of the estimated model leads to the identification of the most dominant propagation delays in that interconnect. Results and comparisons of the proposed techniques are included along all chapters.

Chapter 2

Physical-Based Modeling

The first way to deal with the modeling of long interconnects is based on their physical design. Any proposed structure can be simulated in a EM CAD tool such as CST Studio Suite, Ansys HFSS, Altair Feko, etc., and integrated with the rest of the circuit in order to provide the complete system response. However, despite the advancement in electromagnetic simulation techniques, a full-wave simulation does not have the efficiency needed for a design space exploration and optimization over a large number of parameters. Sometimes, simplifications are possible in order to accelerate the procedure, such as the separation of individual components and the consideration of TL interconnects as having an uniform cross-section, but as the example below will show, these simplifications are not always straightforward in the case of complex interconnect structures.

2.1 The SpaceWire Cable Link

SpaceWire (SPW) technology is a full-duplex data link between two on-board devices (e.g., an instrument and a router or processor) used in spatial applications, that can operate up to 400 Mbits/s [65]. The SPW standard [66] is a document that sets requirements that shall be accomplished with on-board data handling applications in space projects. It allows high speed links up to 10 m in length to be used in the harsh spatial conditions, while also creating standard data interfaces that increase the compatibility among the system components by standardizing the components and the applications of the physical, encoding, data and network layers. The physical layer of a generic SPW link is shown in Fig. 2.1, and consists in the interconnection of low-voltage differential signaling (LVDS) drivers and receivers with a multi-section channel, made of printed circuit boards (PCB), cable assemblies and connectors. The standard specifications [66], meant to guarantee a reliable communication over the SPW channel, concern characteristic impedances, maximum jitter and skew between the differential signals, eye-diagrams opening, common-mode voltage range, insertion and return loss, operating temperature, weight, etc. Thus, particular attention should be payed to the

differential mode (DM) and common mode (CM) [67] characteristics of this link.

Among the different elements of the link, the SPW cable represents a complex structure, consisting of 4 twisted pairs, 5 shields and two micro D-type female connectors. The geometry of the cable cross-section and the material of the cable are not completely specified by the standard [66] and can vary from one cable producer to another, thus introducing additional uncertainties in the model of such structure. However, the validation through measurements of the SPW cable model is challenging, since its terminations are not compatible with standard connectors (e.g., SMA) used by the conventional measuring instruments, and an ad-hoc setup needs to be adopted, thus posing calibration problems. This work presents a physical-based modeling strategy in order to establish a model for the passive part of the physical layer of the SPW link, i.e., excluding the transceivers, and the validation of such model. Such approach defines CAD models for the components of the link, one-by-one, and then cascade them in order to achieve a model for the full link. The description of individual models starts by the shortest component, the connectors, described in the section below.

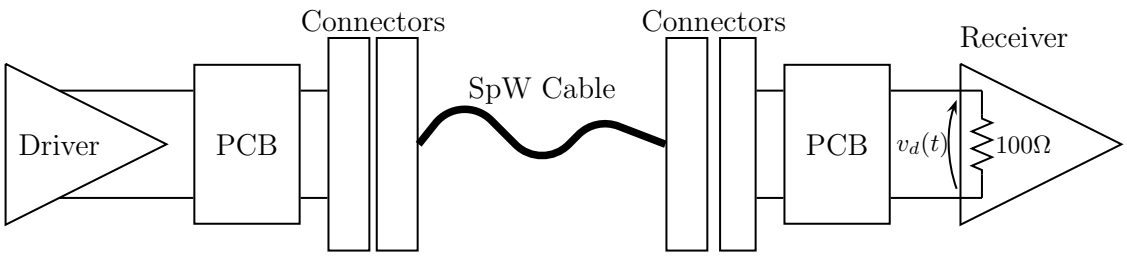


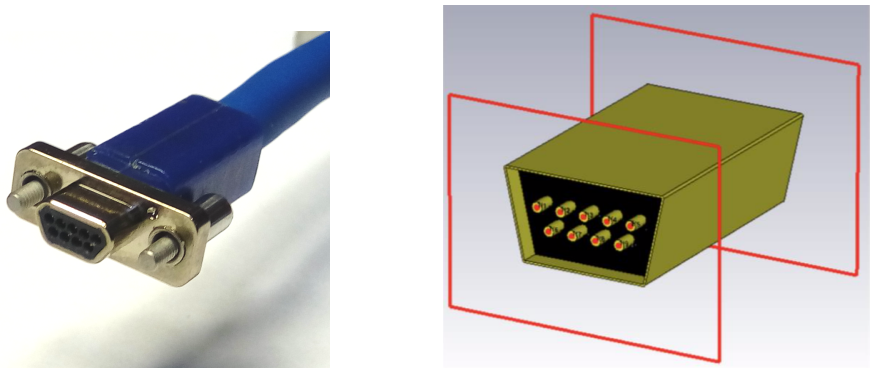
Figure 2.1: Schematic of a SPW link according to [66].

2.2 Micro D-Type Connector

The appropriate connector to be used with the SPW data link has its physical dimensions and the geometry completely specified [68]. The connector type is named micro-D, a miniature connector which is shown in Fig. 2.2(a). It consists on 9 pins with an external metallic shell which provides continuity to the cable shield, thus establishing a common reference on both ends of the cable.

In the absence of a complete CAD model, the proposed modeling approach relies on the simplified 3D implementation of the connector in the full-wave EM solver CST Microwave Studio shown in Fig. 2.2(b). The 3D model is used to accurately characterize the electromagnetic behavior of each pin of the connector, including their coupling and their interaction with the connector shell in the frequency-domain up to 1 GHz.

For the sake of simplicity only the male micro D-type connector is simulated and have its response extracted. This means that, implicitly, any possible differences between the male and the female version of the connector are neglected. The above simplification is motivated by the fact that the only difference between the male and the



(a) (b)

Figure 2.2: Micro-miniature D-type female connector (a) and its 3D implementation in CST MICROWAVE STUDIO (b).

female version in our simplified 3D reproduction derived by the geometry specified by the standard is that the pins in the female version would be 0.33 mm longer with respect to the male version. The above difference can be considered negligible in the bandwidth of use of the connector.

The frequency response obtained from the linear CAD models above usually consists on a finite number K of tabulated frequency-domain samples defined at the complex frequency points $\{s_1, \dots, s_K\} = \{j\omega_1, \dots, j\omega_K\}$. In order to be included in SPICE simulations, the data should be modeled in a way compatible with such simulations, i.e., in terms of standard linear circuit components or, if allowed by the simulation solver, in terms of a differential-algebraic system of equations (DAE) in the same way as the modified nodal analysis (MNA) DAE obtained from standard circuit components [12]. The usual way to extract this CAD model for SPICE simulations is via a rational (or VF) model [23], written as

$$\mathbf{F}(s_k) \approx \tilde{H}(s_k) = \mathbf{R}_0 + \sum_{j=1}^{n_p} \frac{\mathbf{R}_j}{s_k - p_j}. \tag{2.1}$$

The above expansion writes the multiport frequency-response $\mathbf{F}(s_k)$ in terms of poles p_j and residues $\mathbf{R}_j$. The latter, which appear linearly in the model, can be easily estimated via a linear regression [69]. However, the poles, which have a non-linear relation to the model, cannot be estimated as easily. Nonetheless, the well proven VF algorithm, described more in depth in section 4.1, has been successfully employed for this task [23]. It uses a sequence of linear problems to find an appropriate set of poles through iterations. It has important properties such as high computational efficiency, good accuracy which can be further improved by adding additional poles, and a simple formulation [12]. The obtained pole-residue representation can be extracted as a space-state system of equations, or converted to circuital components [21], [22].

Regarding the connector described in section 2.2, it has its response extracted via this algorithm using $n_p = 10$ poles. This extracted model is validated by its simulation in HSPICE and posterior comparison of the simulated scattering parameters with their corresponding curves from the full-wave simulations. The results are shown in Fig. 2.3, showing a very good agreement between them in the selected bandwidth. The resulting netlist can then by employed by most commercial SPICE-like simulators. It is ought to remark that the circuit equivalent, which has been generated by fitting the results of a 3D full-wave solver, already accounts for the frequency-dependence of materials and for all interactions among the parts of the connector. After all these steps, the connector model is set to be used with the other components of the link, e.g., the SPW cable.

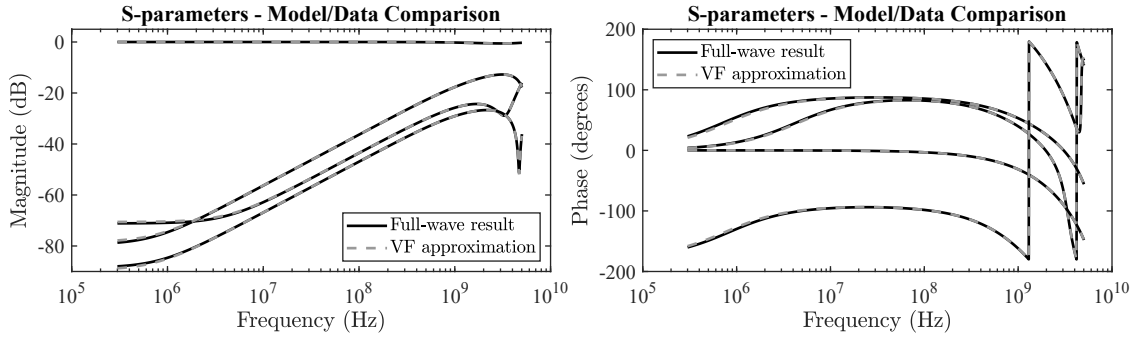
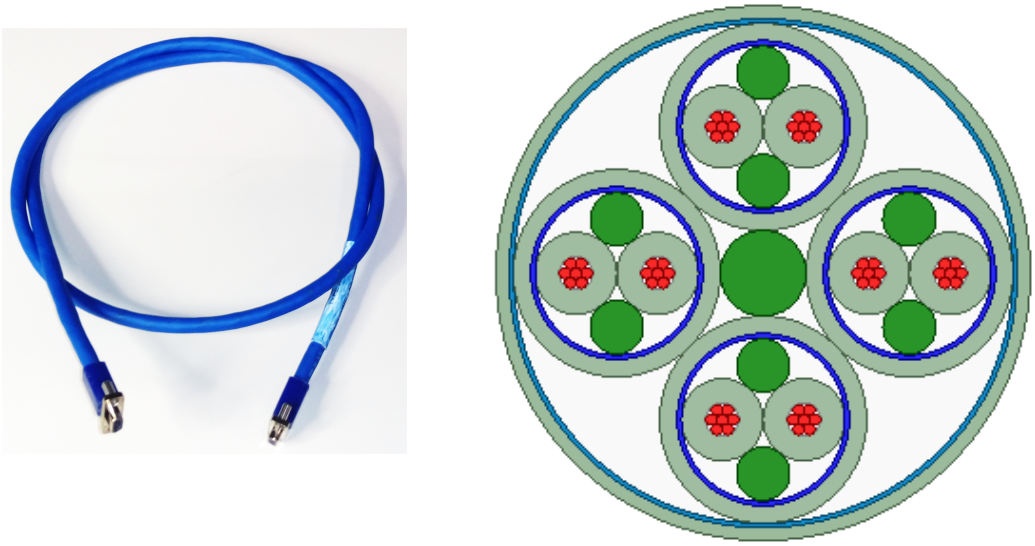


Figure 2.3: Comparison between for S-parameters from the full-wave simulation and the model used in SPICE simulations: magnitude (a); and phase (b).

2.3 SPW cable model

The longest component of the SPW link is the SPW cable shown in Fig. 2.4-(a). As such, it has the largest potential to suffer external interference, and thus it is our main device under test (DUT). Its specification is aimed at a cable well built to support the harsh EM ambient found in space, with the cross section of the SPW cable specified in the standard [66] shown in Fig. 2.4-(b). It consists of 13 conductors: 4 twisted pairs (red circles), each one responsible for transmitting a differential signal via this channel with differential characteristic impedance of $(100 \pm 6)\,\Omega$; each pair is wrapped by a shield (blue circles), and an outer shield (external light blue circle) provides additional electromagnetic protection for the whole bundle of conductors. The twist period (i.e., the length of an entire twist) is specified by the standard to be between 12 and 16 times the outside diameter of two shielded and jacketed wires.

The correspondence between the cable wires and the terminal connector pins is depicted in Fig. 2.5. In the modeling and measurement bandwidth, which extends up to 1 GHz, the cable behaves as a complex distributed structure, since the wavelength at the



(a) (b)

Figure 2.4: SPW cable used for the measurements (a) and cross-section of the SPW cable Variant 01 (b).

maximum frequency (i.e., $\lambda_{MAX} = 15$ cm) is two orders of magnitude smaller than the maximum cable length admitted by the standard (i.e., 10m). As such, it is modeled via transmission line structures, which can account for different levels of details. Here, it is proposed three different models: a mostly rough simplified model, called Model Level 0, an intermediate model, called Model Level 1, and a more detailed structure called Model level 2. They are described in the subsections below.

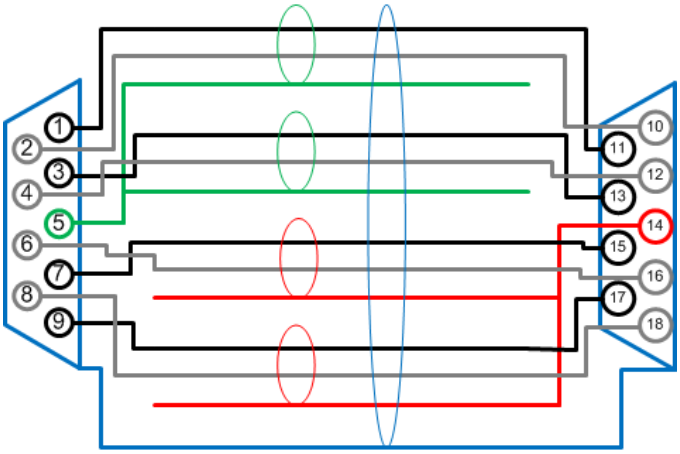


Figure 2.5: Schematic of the SPW cable structure with the corresponding port numbers.

2.3.1 Model Level 0

Some approximations are introduced to simplify the modeling procedure. First of all, an untwisted version of each wire pair is considered: the per-unit-length (PUL) parameters are estimated as a function of frequency by means of a 2D solver based on the cable cross section of Fig. 2.4-(b), and are used to build a W-element of the line in HSPICE [19]. Additionally, all shields of the cable are considered as solid shields. The described cross section with solid shields is extended uniformly along all length of the the cable as shown in Fig. 2.6.

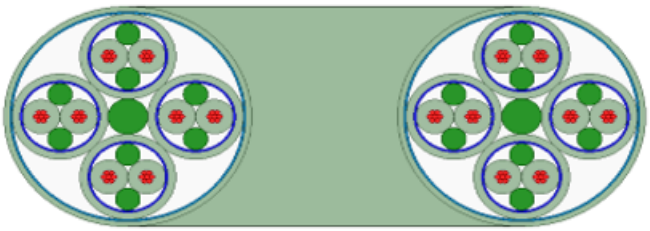


Figure 2.6: Simulated structure of the SPW cable Model Level 0.

Despite the previous structural simplifications, the resulting model accounts for the possible frequency dependence of the materials and for phenomena like proximity effect, skin effect and shield transfer impedance, since the frequency-dependent p.u.l. matrices are computed via a 2D electromagnetic solution.

2.3.2 Model Level 1

The previous model can be improved by accounting for the effect of the twists along each differential pair of wires. To this aim, the cable has been modeled by using the cascade of 8 cells per twist period; each 8 cells have then an approximate length of 2.5 cm, resulting in 40 twist/m, and each pair of wires inside the cell is still parallel but its position is shifted 45° around its center with respect to the previous cell. Such structure is depicted in Fig. 2.7.

Figure 2.8 provides a comparison between the DM and CM reflection loss at a port of the system for the twisted and untwisted model of the SPW cable, and also the CM to DM conversion along the cable. The results show that the considered scattering parameters are similar for both versions of the cable, making this complexity in the model unnecessary. However, it could be an important aspect if the model would be employed in simulations where the effect of an external source of EMI to twisted pair of wires is analyzed.

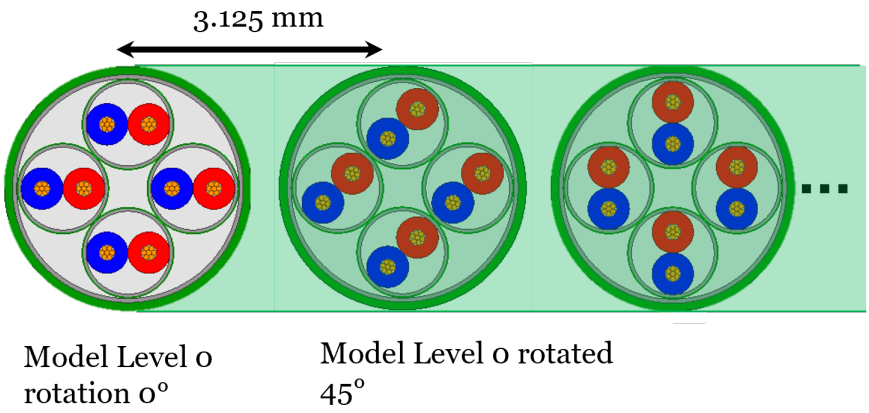


Figure 2.7: Simulated structure of the SPW cable Model Level 1.

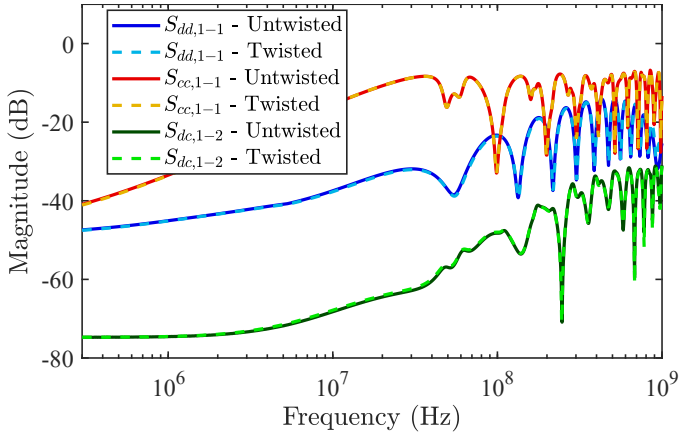


Figure 2.8: Comparison of some mixed-mode scattering parameters generated by the models of twisted and untwisted SPW cables.

2.3.3 Model Level 2

The concept of a direct and abrupt transition from the uniform SPW cable to the connector is unrealistic. As can be seen in Fig. 2.9, the real transition happens in a more smooth and less defined way [70]. The differential pairs of wires are separated from its pair and connected to the appropriate pin in the connector, changing its position along the way.

Therefore, the SPW cable model can be enhanced by considering the use of two intermediate blocks, with the aim of recreating the transition between the cable and the connector cross-sections, as shown in Fig. 2.10. The transition is estimated visually to have a total length of 25 mm. A simplified model of this transition is schematically illustrated in Fig. 2.11. It is assumed to consist of the cascade of two blocks with constant cross sections. The transition block no. 1 is 23 mm long and is intended to represent the

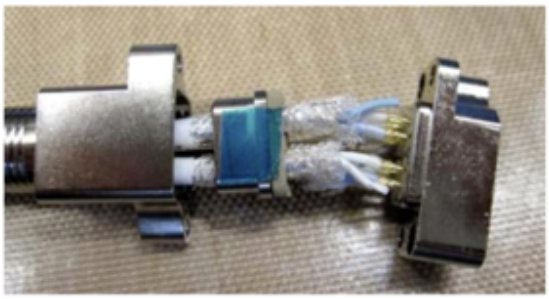


Figure 2.9: Internal picture of a Micro D-Type connector attached to the SPW cable [70].

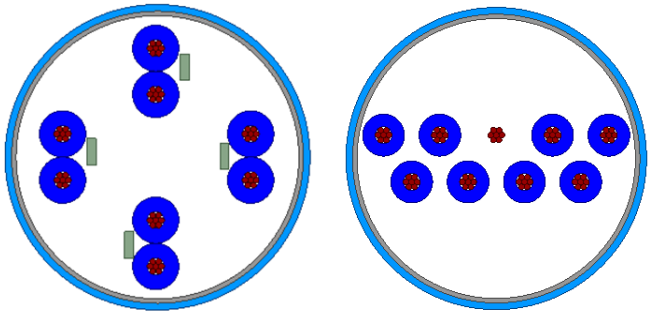


Figure 2.10: At the cable ends, a transition is necessary to fit the cable cross-section (Left panel) to the D-type connector cross-section (Right panel). In the left panel, the gray rectangles next to wire represent the shield of each twisted pair reduced to a pigtail; in the right panel, the smaller dot indicates the position of the two pigtails continuing through the connector, while the other two pigtails are left in open circuit.

part of the transition closer to the connector, where the wires stick out of the connector pins; the cross-section is illustrated in the right panel of Fig. 2.10 and is assumed to remain constant throughout the block length. The transition block no. 2 is 2 mm in length and is intended to represent the section where the wires are still grouped together and stick out of the cable; the inner shields are reduced to pigtails and run parallel to the other wires (left panel of Fig. 2.10).

The two sections corresponding to the cable transitions have been characterized through their PUL parameters computed by means of a 2D field solver. The resulting two sets of frequency-dependent PUL matrices are implemented in HSPICE via two cascaded W-elements in each of the cable extremities, and the cable model level 0 in between them. The next section aims at validating these models, showing their adequacy for predicting the measurement of the frequency response of the actual SPW cable link.

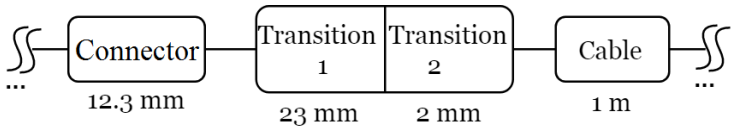


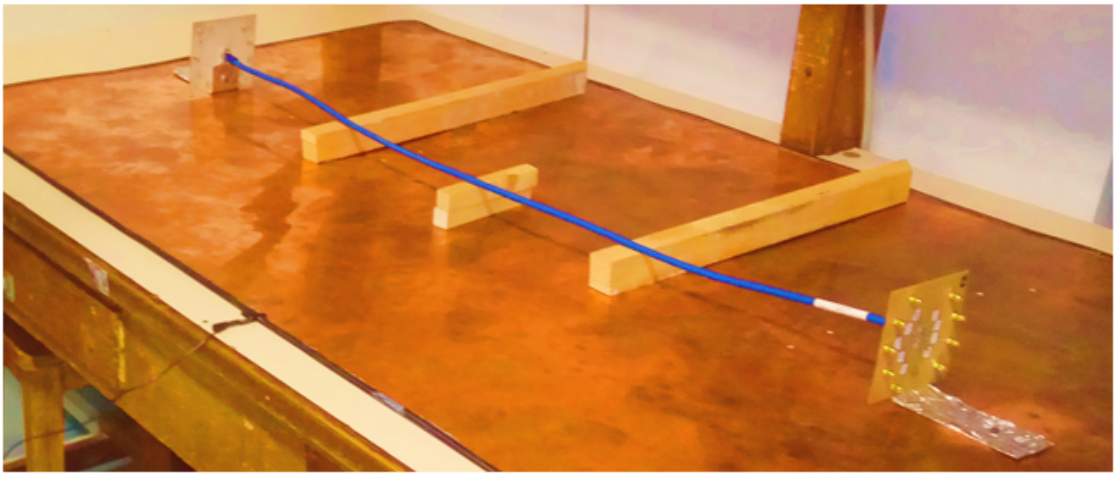
Figure 2.11: Schematic of the cable-connector transition adopted in the simulations.

2.4 Measurement setup

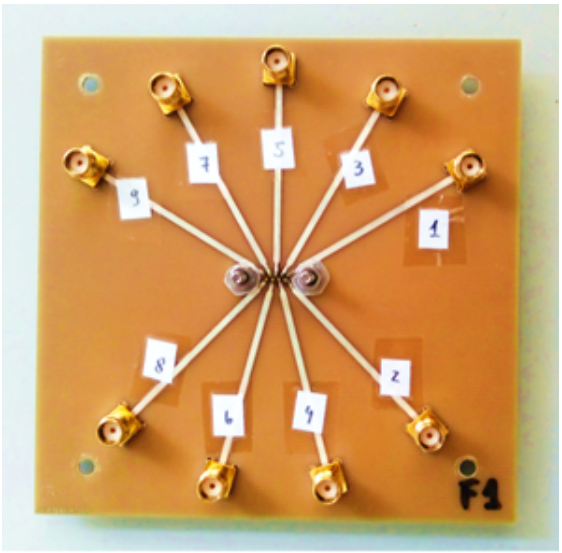
The models presented above have a solid theoretical foundation. However, they can only be validated once they are compared with experimental data. The measurement setup used for the experimental validation of the derived models is shown in Fig. 2.12-(a) [71]. It consists of a 1 m long SPW cable connected on both ends to the test fixture shown in Fig. 2.12-(b), built for compatibility with a Vector Network Analyzer (VNA), and which correspond to a possible implementation of the PCBs from the setup in the diagram of Fig. 2.1.

The test fixtures at both ends of the line are equal in order to have a similar effect at each port of the system. Each fixture consists of a 9-pin male micro D-type connector soldered on a PCB; the connector shell is referred to the PCB ground plane and each of the 9 pins are connected to individual traces leading to the VNA cables via SMA adapters. The test fixture board, built on a FR4 substrate, consists of 9 copper traces of length 42.6 mm designed in order to have a 50 Ω characteristic impedance in the main part, closer to where the RF connectors are soldered, and narrower near the micro D-type connector in order to reach the connector pins with a sufficient clearance between the paths, as shown in Fig. 2.12-(c). As the distance between the traces is not constant, the structure cannot be considered as a uniform transmission line structure. Therefore, the electrical behavior of the PCB has been derived by a 3D electromagnetic simulation, followed by a rational approximation based on (2.1), producing an equivalent HSPICE netlist.

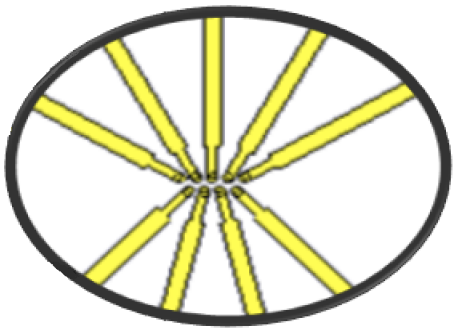
The construction of this test fixture directly compatible with the VNA via the SMA connectors allows the validation of its extracted models via measurements. In order to do this, the S-parameters of the PCB are measured and compared with the extracted models, in two different ways: (i), when only the PCB and its SMA connectors are assembled, without the Micro D connector; and (ii), when the Micro D-type female connector has also been soldered to the PCB fixture. The first measurement validates the PCB model, while the second also provides an additional validation of the connector model described in Sec. 2.2 [71]. The results comparing some of these measurements to simulations of their models are shown in Fig. 2.13. It is interesting to notice that, as the ends of the connector could not be connected to defined loads, they remained in open circuit and thus, with the open test fixture most of the signals are reflected at the connector end, which can be clearly seen in the plot of the $S_{4,4}$ parameter in Figs. 2.13(b) and 2.13(d). Figs. 2.13(a) and 2.13(c) compare, for each structure, the measured



(a)



(b)



(c)

Figure 2.12: Panel (a): measurement setup used to validate the proposed physical-based models of the SPW cable and of the D-type connector; Panel (b): test fixture (i.e., PCB board) used to connect the SPW cable to the VNA; Panel (c): detail of the PCB center.

amplitude and phase of the measured coupling between SMA ports no. 1 and 2 with the model prediction. The good agreement between the measured and the simulated results validate the proposed physical-based models of the fixtures used to interface the cable.

The full measurement setup turns out to be a 18-port system, which contains all the linear structures and components of a SPW channel. Even if most of the requirements specified by the SPW standard [66] are related to the differential mode (DM) and

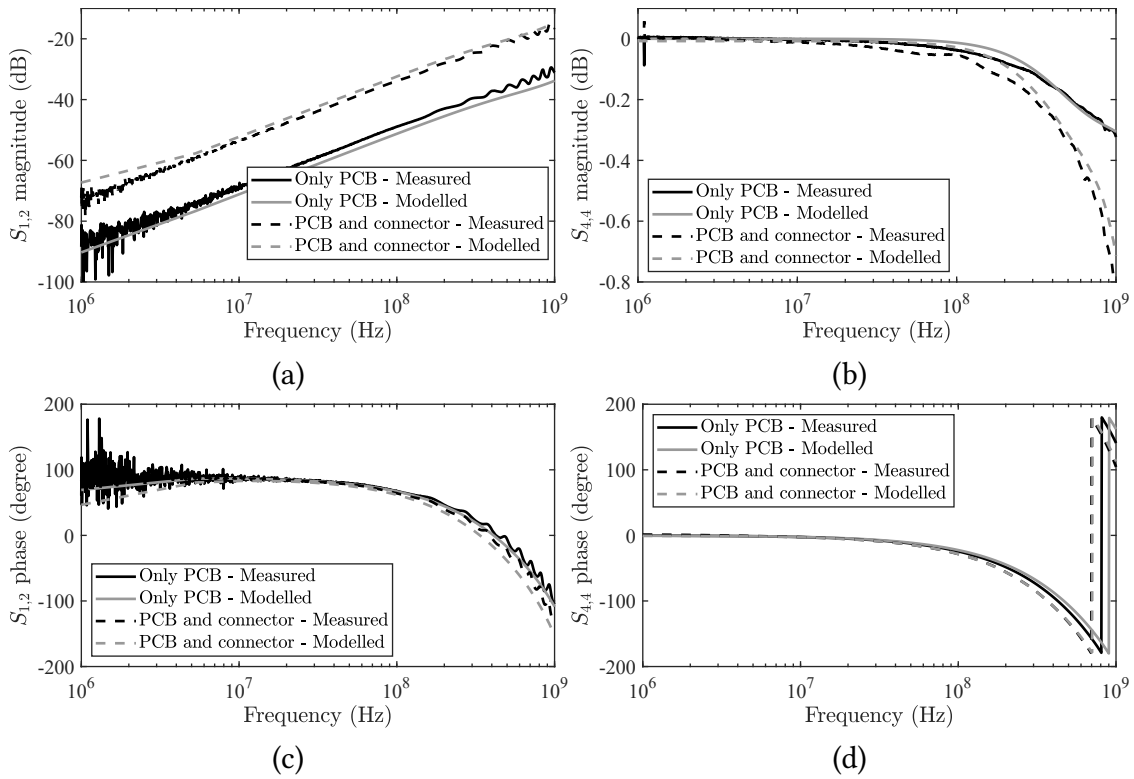


Figure 2.13: Comparison of measurements and models of two fixtures made by the PCB alone and the PCB with a soldered connector: reflection coefficient of port no. 4 (panel a) and coupling between the two adjacent lines no. 1 and 2 (panel b).

common mode (CM) signaling along the link, the frequency-domain instruments able to fully characterize the link are mostly available with single-ended interfaces. Baluns can be used to make a differential measurement [72], but they do not provide information about the CM and also require an additional component, influencing the measured response. Therefore, single-ended measurements are mandatory if information about the CM propagation is required. Since it is also possible to convert the single-ended scattering matrix $\mathbf{S^{(d)}}$ of a differential pair from the ports $1-1'$ to $2-2'$ to the mixed-mode [67] scattering parameters matrix $\mathbf{S}^{(mm)}$, the DM and CM characterization of the communication channel can be readily derived from the single-ended data via [73]:

$$\mathbf{S}^{(mm)} = \mathbf{M}\mathbf{S}^{(d)}\mathbf{M}^{-1}, \tag{2.2}$$

where

$$\mathbf{S}^{(d)} = \begin{bmatrix} S_{11} & S_{11'} & S_{12} & S_{12'} \\ S_{1'1} & S_{1'1'} & S_{1'2} & S_{1'2'} \\ S_{21} & S_{21'} & S_{22} & S_{22'} \\ S_{2'1} & S_{2'1'} & S_{2'2} & S_{2'2'} \end{bmatrix}, \tag{2.3}$$

$$\mathbf{M} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \tag{2.4}$$

and

$$\mathbf{S}^{(mm)} = \begin{bmatrix} S_{11}^{dd} & S_{12}^{dd} & S_{11}^{dc} & S_{12}^{dc} \\ S_{21}^{dd} & S_{22}^{dd} & S_{21}^{dc} & S_{22}^{dc} \\ S_{11}^{cd} & S_{12}^{cd} & S_{11}^{cc} & S_{12}^{cc} \\ S_{21}^{cd} & S_{22}^{cd} & S_{21}^{cc} & S_{22}^{cc} \end{bmatrix}, \tag{2.5}$$

in which c refers to the CM parameters, d to the DM parameters, 1 to the port $1 - 1'$ and 2 to the port $2 - 2'$ (e.g., S_{12}^{cd} represents conversion from the differential mode in port to into the common mode in port 1, and so on).

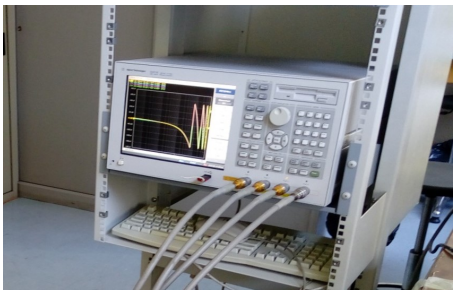


Figure 2.14: Vector Network Analyzer used in the validation of the SPW link.

The port numbering of the 18 ports is shown in Fig. 2.5. The full characterization of a structure with a larger number of ports via a measurement instrument with a smaller number of available ports should be performed sequentially, in a way that all the parameters outside the main diagonal of the scattering matrix are obtained at least once [67], [74]. By means of this four-port VNA depicted in Fig. 2.14, the 18×18 scattering matrix is obtained by performing 30 sets of measurements, even if the theoretical minimum would be 27 sets of measurements. The procedure measures all the off-diagonal elements of the scattering matrix at least once [74], [75], as shown in Fig. 2.15. The extra measurements are used to check the repeatability and the reliability of the measurement results. At the same time, the ports not connected to the VNA are terminated with a matched impedance of 50 Ω. The full frequency-domain characterization of the 18 single-ended ports established in this setup is achieved with this four-port VNA, in the bandwidth from 300 kHz to 1 GHz.

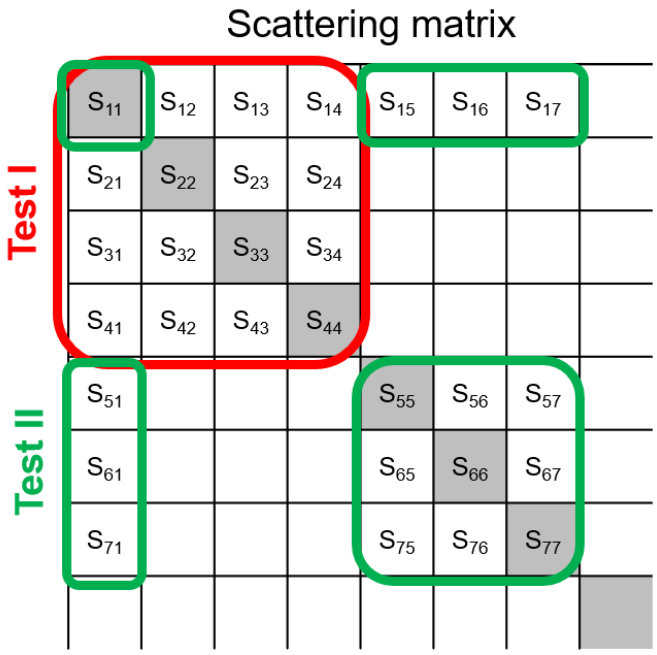
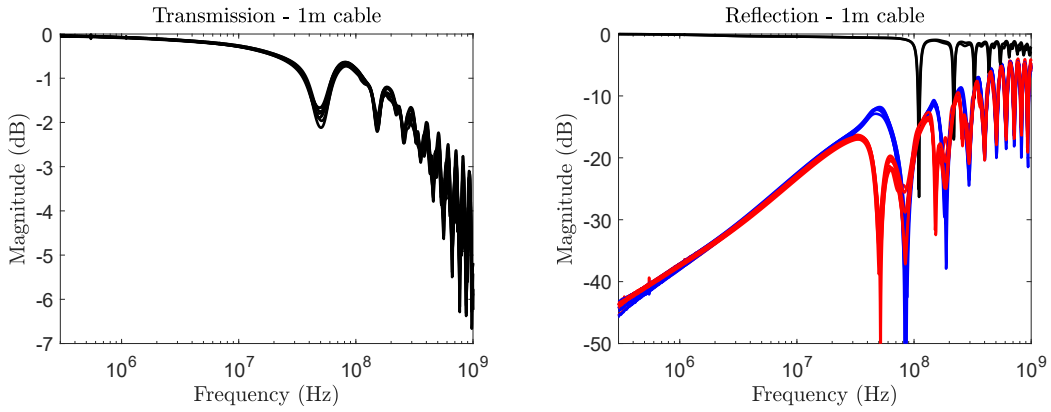


Figure 2.15: Procedure for measurement of the 18×18 scattering matrix of the SPW link via 4-port VNA. The tests depicted in the picture are added until the 18×18 matrix is completely filled after 30 tests.



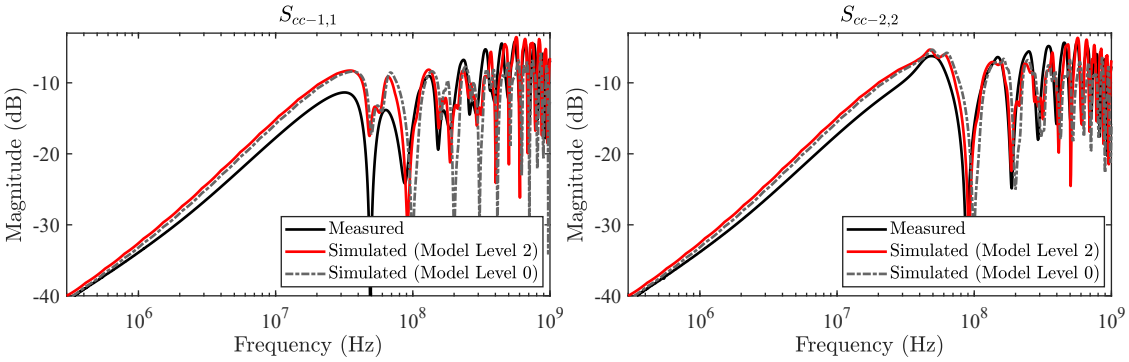
(a) (b)

Figure 2.16: Measured single-ended insertion loss between any two ports of the setup and reflection loss on the 18 ports. Panel (a): transmission from one end to the other end of the conducting wires; Panel (b): reflection observed on all the measured ports.

2.5 Validation

Some of the single-ended measurements obtained via the procedure described in the previous section are illustrated in Fig. 2.16. The transmission pattern, plotted in Fig. 2.16-(a), is similar for all 8 conductors of the 4 twisted pairs of the cable. However, reflection plots displayed in Fig. 2.16-(b) show three different patterns: the first one, in black, indicates that most of the signal is reflected and refers to ports connected to the inner shields, which always have one end in open circuit; the other two patterns refer to signal wires and their difference is caused by the asymmetrical shield connection: the blue curves refer to pairs whose shields are left open in the section under test, while the red curves refer to pairs whose shields are left open far from the section under test (this case presents additional low-frequency resonances determined by longer signal reflections).

The obtained results are then converted into the corresponding mixed-mode representation for each differential pair present in the SPW link [73], which is more relevant to the differential signaling employed in such channel. This conversion transforms the single-ended scattering parameters to the corresponding DM and CM scattering parameters S_{dd} and S_{cc} respectively, and to the mode-conversion parameters: S_{cd}, that describes the DM-to-CM conversion, and S_{dc}, that describes the CM-to-DM conversion.



(a) (b)

Figure 2.17: Comparison of CM reflection coefficients: measurements vs. HSPICE simulations of models with increasing degree of details. Panel (a): CM port defined between pins 1-2 of Fig. 2.5. Panel (b): CM port defined between pins 10-11 of the same Fig. 2.5.

For the mixed-mode validation, the measurement setup presented in Fig. 2.12 has been implemented in HSPICE by replacing each component (i.e., the test-fixtures, the connectors and the SPW cable) with its corresponding equivalent circuit netlist, which are already validated. The scattering parameters are then calculated from the voltage and current waveforms resulting from the frequency-domain HSPICE simulation. Both the measurements and the HSPICE model simulations are converted into mixed-mode scattering parameters. In addition, two different models of the SPW cable are considered in the HSPICE simulations: *Model Level 0* (i.e., with a direct connection between

the cable and the connector) and *Model Level 2* (i.e., with a *smoother* transition between the cable and the connector). Model Level 1 is dropped due to its similarity to Model Level 1.

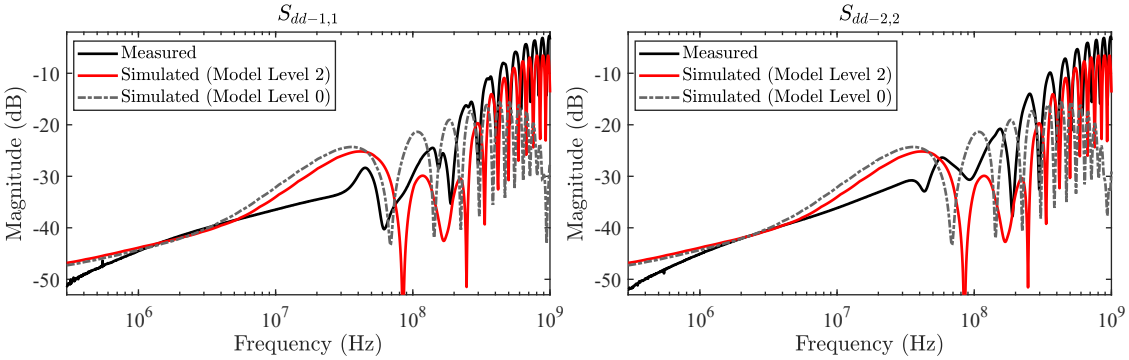


Figure 2.18: Comparison of DM reflection coefficients: measurements vs. HSPICE simulations of models with increasing degree of details. Panel (a): DM port defined between pins 1-2 of Fig. 2.5. Panel (b): DM port defined between pins 10-11 of the same Fig. 2.5.

The frequency response of the CM reflection scattering parameter at the ports of a twisted pair of conductors is shown in Fig. 2.17. The same two patterns that happen in the single-ended parameters in Fig. 2.16 appear here, meaning that in Fig. 2.17(a) is shown the reflection loss at a port where the inner shield of the cable is connected to the central pin of the connector, and in Fig. 2.17(b) is shown the reflection loss at a port with the inner shield in open-circuit. It should be noticed that the peaks on the frequency response achieve a value close to -5 dB, slightly increasing as the frequency increases. The Model Level 2, introducing a transition between the cable and the connector significantly improves the accuracy of the model prediction at higher frequencies, even though the adopted transition is drastically simplified with respect to reality.

Figure 2.18 shows the frequency response of the DM reflection scattering parameter at both ends of a twisted pair. Here, the terminations of individual shields (either in short circuit or in open circuit) do not have a significant impact, as the responses at both ports are similar. On the other hand, the importance of an adequate model of the transitions between the cable and the connector is evidenced by the fact that Model Level 2 better accounts reason for the growing trend of the peaks as the frequency increases.

The frequency behaviour of other mixed scattering parameters is shown in Fig. 2.19, where it is confirmed that the Model Level 2, including the transitions from the cable to the connector, helps better explaining the high-frequency portion of the undesired mode conversions. It is ought to be noted that the apparent deviation between measured and simulated values at low frequencies is due to the limitations imposed by the noise floor of the measurement equipment, while simulation computations are performed

with a higher numerical accuracy.

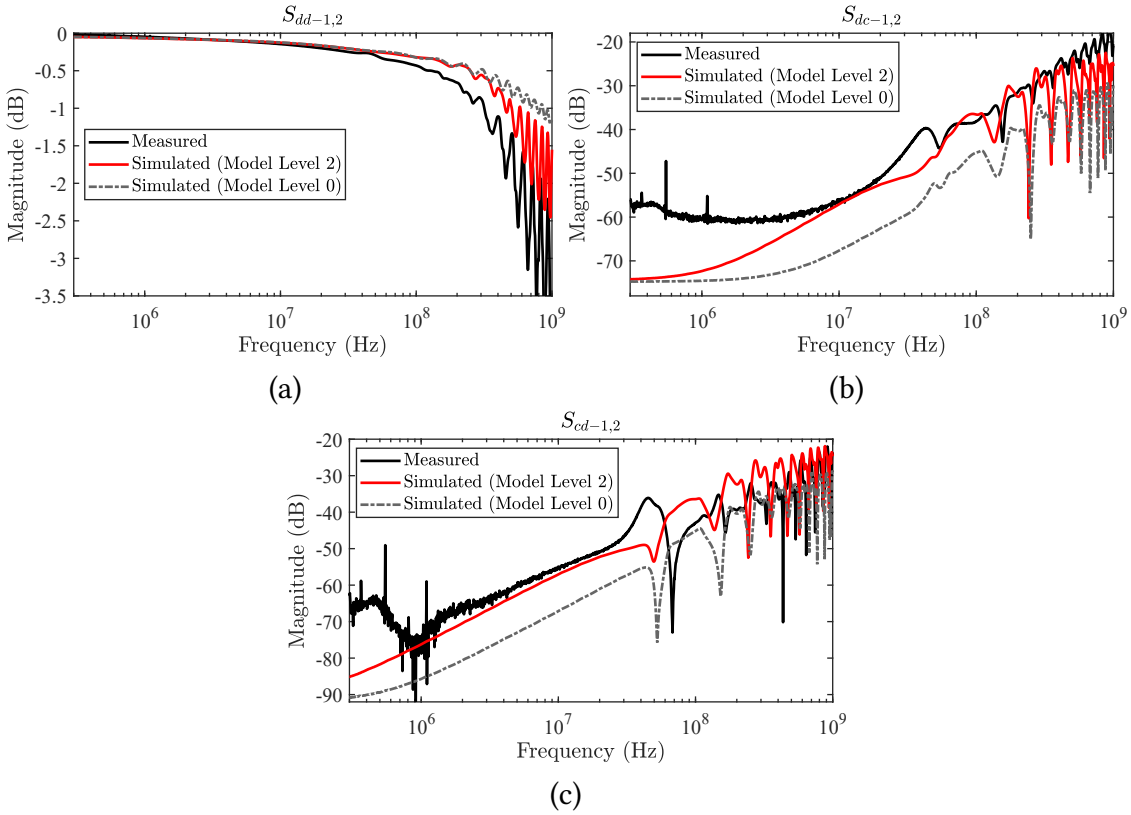


Figure 2.19: Comparison of some mixed-mode scattering coefficients: measurements vs. HSPICE simulations of models with increasing degree of details. DM-mode transmission (panel (a)) and mode conversion (panel (b) and (c)) from port 1-2 to port 10-11 of Fig. 2.5.

2.6 Remarks

A physical-based modeling of each of the components of the SPW physical layer was presented here and validated via experimental measurements. The simulation setup assembled with the individual models of the parts that constitute the SPW link was able to replicate its actual behaviour. However, we saw that many details have to be considered, e.g. a small cascade of blocks used to represent the transition between the cable and the connector (instead of using an abrupt transition) achieve a model with better accuracy, specially at higher frequencies. It was seen that the Model Level 2 developed and presented can be used for the reliable prediction of digital signals along the SPW links, with reliable CM and DM representations of the communication differential pair, and good results for the mode conversion along the channel. However, such results

were only obtained after the construction of a measurement setup to characterize the frequency-domain response of the SPW link and an incremental improvement of the cable model design. In the next chapter, we start to present data-based black-box models which do not require knowledge of the structure of the involved components, and is based only on behavioral data of the DUT obtained from reliable simulations or even directly from performed measurements. With such approach, much of the uncertainty on the design of the physical model be avoided, as the models are directly based on data that is assumed to be reliable and the focus can be placed on the accuracy and target characteristics of such model.

Chapter 3

Data-Driven Metamodels via the Least-Squares Support Vector Machine Regression

An alternative of the previously described physical-based models is the use of data-driven metamodels. With such approach, the individual modeling of the system components is unnecessary, as a metamodel is estimated only based on data from the system ports. The data can be obtained via measurements or simulations, and then the most convenient data acquisition approach can be chosen. A black-box data-driven model is a system defined in terms of its inputs and outputs characteristics, regardless of the physics and internal mechanisms that achieve such behavior. Black-box modeling techniques aim to construct a surrogate model which is able to mimic the behavior of a real system via a black-box model by inferring the unknown model from data representing the system's behavior to a certain stimulus. They perform a deterministic procedure, such as the computation of an analytical formula, which provides an approximate relation $\tilde{y} = \tilde{\mathscr{M}}(\mathbf{x}) \approx y$ from a set of input parameters $\mathbf{x} \in \mathbb{R}^d$ to an output $y \in \mathbb{R}$. Sometimes, they are also called metamodels, or surrogate models, because they act as a surrogate for the true system $\mathscr{M}(\mathbf{x})$. In this sense, they are indispensable for the exploration of a large design space with different goals, e.g., design optimization [76], uncertainty quantification [59], [61], [62], [77]–[80] or sensitivity analysis [81]. Their fast and readily available mathematical relation provides an easy way to obtain the large number of data required by the tasks above.

The classical way of estimating a metamodel is via inference of empirical data originating from the target model, which statistically infers the dependency of the output with respect to the input [82]. In this regard, metamodels can be split into two groups:

1. *Parametric models*, which assume a finite set of parameters and basis functions, and perform an inference particular to these specific parameters in the models. An effective model is structured over intuition and deep knowledge about the

problem context [69] or physical laws [82]. The parameters capture all the behavior of the data, i.e., the model predictions are independent of the data. Additionally, its complexity is fixed, no matter the amount of data available;

2. *Non-parametric models*, do not explicitly assume a set of parameters for the model, and instead perform a generic inference from the available data. The approximate function is inferred solely from the empirical data set. The parameters in the model can be seen as a function of the data, and the amount of information which they capture grows as the amount of data is increased. This inference does not require any a priori information about the statistical law underlying the problem or function [82].

A metric to evaluate the accuracy of such models is via the the use of the mean squared error (MSE):

$$\text{MSE} = \frac{1}{K}\sum_{k=1}^{K} \|\mathrm{y}_k - \tilde{\mathrm{y}}_k\|^2. \tag{3.1}$$

The minimization of the metric above for a linear parametric model leads to the least-squares (LS) regression, one of the simplest modeling implementations. This method interpolates the observed outputs $\{\mathrm{y}_1, \dots, \mathrm{y}_K\}$ using a set of basis functions (or *features*) $\{\varphi_0(\mathbf{x}), \dots, \varphi_N(\mathbf{x})\}$. In other words, for the available data, the approximate model is equivalent to the following linear system [56]

$$\begin{bmatrix} \tilde{\mathrm{y}}_1 \\ \vdots \\ \tilde{\mathrm{y}}_K \end{bmatrix} = \begin{bmatrix} \varphi_0(\mathbf{x}_1) & & \varphi_N(\mathbf{x}_1) \\ & \ddots & \\ \varphi_0(\mathbf{x}_K) & & \varphi_N(\mathbf{x}_K) \end{bmatrix} \begin{bmatrix} \mathrm{w}_0 \\ \vdots \\ \mathrm{w}_N \end{bmatrix}, \tag{3.2}$$

or equivalently, in terms of vectors and matrices:

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w}. \tag{3.3}$$

As an example, for a regression via a quadratic polynomial function, the basis functions are $\{1, x, x^2\}$. The LS model estimation consists on finding the best vector of parameter weights $\mathbf{w} = [\mathrm{w}_0, \dots, \mathrm{w}_N]^T$ such that the MSE is minimized. This is obtained via

$$\mathbf{w} = \left(\mathbf{\Phi}^T\mathbf{\Phi}\right)^{-1}\mathbf{\Phi}^T\mathbf{y}. \tag{3.4}$$

The resulting approximate model is then written according to

$$\mathrm{y}(\mathbf{x}) \approx \tilde{\mathrm{y}}(\mathbf{x}) = \sum_{i=0}^{N} \mathrm{w}_i\varphi_i(\mathbf{x}). \tag{3.5}$$

When this method is applied for high dimensional problems, i.e., when d is large, it suffers from the *curse of dimensionality*, which prevents it from capturing singularities

in the problem without a prohibitive number of training samples [82]. Additionally, more and more features can be added to match the training data, but this may lead to a larger issue called *overfitting*. Overfitting happens when some metamodeling technique learns every sample in a set of data from which it is estimated, but so well that it learns also the unique noise present in this small set of samples [69]. An example showing this phenomenon is shown in Fig. 3.1. A set of noisy data samples is extracted from a smooth system. Then, two wrong models are estimated: a simple linear regression in blue, which is too is simple to capture the higher order variations of the data, and an polynomial model with maximum order equal to 8 in red, which uses more features than necessary, and thus overfits the data, and is specially bad when the model is extrapolated outside of the available samples range. In a one dimension space, it is clear to visualize this phenomenon. However, most modeling problems are in higher dimensional spaces, where this visualization is not clear, and so more robust techniques are necessary.

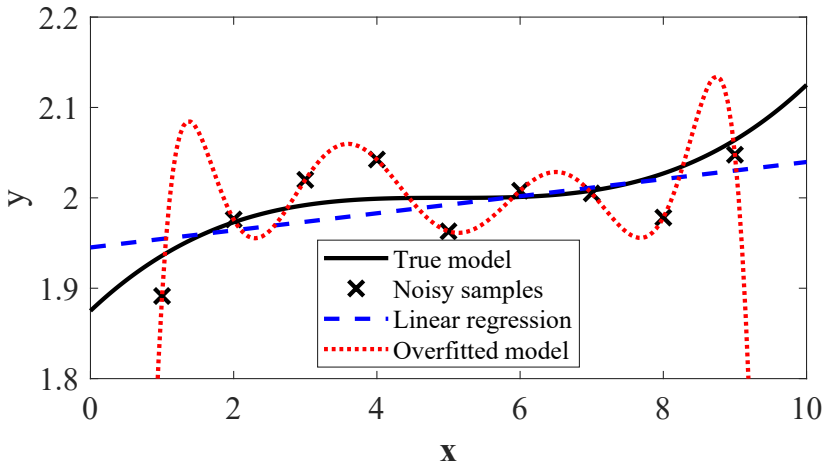


Figure 3.1: Example of a one-dimensional metamodel showing the difficulty in selecting the adequate level of complexity for the model.

Other common reason why a metamodel fails, besides the above ones, might include an inadequate pre-processing of the training data, lack of validation and extrapolation of the results [69]. Adequate attention should be payed to model selection, as non-informative features add uncertainty to the model. Compact representations, i.e., representations with a minimal number of features, often also have good generalization properties [58].

A solution for many of those issues is the use of *regularization*. While it solves the issue of obtaining a unique solution for the system in (3.2) in the cases where $N \geq K$, it also inserts a bias-variance trade-off on the choice of $\mathbf{w}$ which might be used to prevent overfitting and for feature selection, pursuing less complicated models [56]. Among the machine learning techniques presented in section 1.2, the least-squares support vector machine (LSSVM) is a powerful alternative to the modeling of electric interconnects.

While it does not oversimplifies the resulting model, it also uses the positive features of regularization in order to obtain a metamodel which generalizes well for data not used in its training. Therefore, the next sections study the LSSVM in depth in order to apply it to improve electric interconnect models.

3.1 The Least-Squares Support Vector Machine

The least-squares support vector machine is a black box machine learning technique that can be applied in classification or regression tasks. The standard LSSVM regression searches to approximate a set of training data pairs $\{(\mathbf{x}_k, \mathrm{y}_k)\}$ for $k = 1 \dots K$, with a non-linear regression $\tilde{\mathrm{y}} = \tilde{\mathscr{M}}(\mathbf{x})$ of the input $\mathbf{x} = \left[x_1, \dots, x_d\right]^T \in \mathbb{R}^d$. Let us start considering the following *primal space* formulation of the LSSVM regression which maps y as a function of $\mathbf{x}$ according to:

$$\tilde{\mathrm{y}}(\mathbf{x}) = \langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{x}) \rangle + b = \sum_{i=1}^{N} \mathrm{w}_i \varphi_i(\mathbf{x}) + b, \tag{3.6}$$

which is a parametric model with feature space $\boldsymbol{\varphi}(\cdot) : \mathbb{R}^d \to \mathbb{R}^N$. In this model, $\varphi_0(\mathbf{x}) = 1$ from (3.2) is explicitly represented via the constant bias term b. The operator $\langle \cdot, \cdot \rangle$ represents the inner product, and the model output is based in the inner product of the weights and the basis functions. The LSSVM regression is derived by solving the following optimization problem [60]:

$$\begin{aligned} \min_{\mathbf{w}, e_k} \quad & J_p(\mathbf{w}, e) = \frac{1}{2} \langle \mathbf{w}, \mathbf{w} \rangle + \frac{\gamma}{2} \sum_{k=1}^{K} |e_k|^2 \\ s.t. \quad & \langle \mathbf{w}, \boldsymbol{\varphi}(\mathbf{x}_k) \rangle + b + e_k = \mathrm{y}_k \end{aligned} \tag{3.7}$$

for $k = [1, \dots, K]$. The model error $e_k = \mathrm{y}_k - \tilde{\mathrm{y}}(\mathbf{x}_k)$ is computed as the difference true data and approximate model. The optimization problem minimizes two terms: the loss function, which for the LSSVM regression corresponds to the sum of squared error (SSE); and the ℓ_2-norm regularization term $\langle \mathbf{w}, \mathbf{w} \rangle = \|\mathbf{w}\|^2$. The proper selection of γ makes the model both accurate and flat, providing a way to regulate the focus on those two characteristics: a large value for γ will lead to a model with smaller error on the training set, but that may have difficulties in the generalization of its predictions for $\mathbf{x}$ values located far from the training samples. In order to avoid such overfitting, γ should be regulated so that the model becomes flat and smooth, generalizing better across all the design space, while also keeping a small error. To find the minimum of the optimization problem in (3.7), we can use its Lagrangian, which for the problem in (3.7) writes:

$$\mathscr{L}(\mathbf{w}, b, \mathbf{e}; \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \gamma \sum_{k=1}^{K} e_k^2 - \sum_{k=1}^{K} \alpha_k \left(\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_k) + b + e_k - \mathrm{y}_k \right), \tag{3.8}$$

where the coefficients α_k are composed by the Lagrange multipliers belonging to the optimization problem, with $\alpha_k \in \mathbb{R}$. Computing the derivatives of the Lagrangian and making them equal to zero results in the definitions below:

$$\mathbf{w} = \sum_{k=1}^{K} \alpha_k \boldsymbol{\varphi}(\mathbf{x}_k), \tag{3.9}$$

$$\sum_{k=1}^{K} \alpha_k = 0, \tag{3.10}$$

$$e_k = \frac{\alpha_k}{\gamma}, \tag{3.11}$$

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_k) + b + e_k - \mathrm{y}_k = 0. \tag{3.12}$$

By combining the equations in (3.12) and (3.10), and substituting in (3.12) the value for $\mathbf{w}$ and e_k in (3.9) and (3.11), we can build the following linear system of equations:

$$\begin{bmatrix} \boldsymbol{\Omega} + \mathbf{I}_K/\gamma & \mathbf{1}_K \\ \mathbf{1}_K^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix}, \tag{3.13}$$

where $\mathbf{y} = [\mathrm{y}_1, \ldots, \mathrm{y}_K]^T$, $\mathbf{1}_K = [1, \ldots, 1]^T$ is a vector containing K equal elements, $\mathbf{I}_K$ is an identity matrix with size $(K \times K)$, and the $(K \times K)$ kernel matrix $\boldsymbol{\Omega}$ depends on the kernel function $\mathrm{k}(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ computed over the available training samples. This kernel function is a key element for the LSSVM and for this work, and thus, we will dedicate section 3.3 for its detailed explanation. The element $\Omega_{i,j}$ in the i-th row and j-th column of the kernel matrix is defined by applying the kernel function as follows:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_j) \right\rangle. \tag{3.14}$$

The unknowns $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]^T$ and b are computed by solving (3.13). After determining those parameters, the so called *dual space* model representation is written as

$$\tilde{\mathrm{y}}(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k \mathrm{k}(\mathbf{x}, \mathbf{x}_k) + b. \tag{3.15}$$

This model is equivalent to the primal space model, with the main difference being their dimension: the primal space model has N terms, which is the number of basis of the primal space, i.e., the dimension of the feature space $\boldsymbol{\varphi}(\mathbf{x})$. Its determination require the computation of N unknowns, i.e., the N terms in $\mathbf{w}$. On the other hand, the dual space model is a nonparametric regression, i.e., it has a number of terms equal to the number of training samples, K, and its estimation require the computation of the K terms in $\boldsymbol{\alpha}$. Both models also have the same bias term b.

3.2 Extension of the Least-Squares Support Vector Machine for Complex Models

In the LSSVM regression framework described above, both the input and the output consist on real valued functions, and cannot be directly applied for models involving complex-valued quantities. The literature present some ways to do it, either by treating individually the each part of the complex quantity [63], or by adapting the technique for complex-valued models [83]. The former technique might achieve its goal, however it loses the intrinsic relation among the real and imaginary parts which their target quantities might have. Therefore, for the latter category, the literature provide some references that extend the kernel-based methods to perform regression on complex-valued functions, such as the support vector machine (SVM) [84], the least-mean-square (LMS) algorithm [85]–[87] and mixture models [88]. The LSSVM regression can also be readily extended to a complex framework [83], [89]–[91], where its advantages obtained from the regularization and its dual formulation are maintained.

In order to arrive at its complex-valued formulation, the LSSVM regression from Sec. 3.1 is modified to have complex-valued inputs $\mathbf{x} \in \mathbb{C}^d$ and output $\mathrm{y} \in \mathbb{C}$. With these definitions in mind, the primal space formulation of the LSSVM regression can be rewritten as

$$\tilde{\mathrm{y}}(\mathbf{x}) = \langle \mathbf{w}, \boldsymbol{\varphi}^*(\mathbf{x}) \rangle + b, \tag{3.16}$$

where the regression coefficients $\mathbf{w} \in \mathbb{C}^N$, basis function $\boldsymbol{\varphi}(\cdot) : \mathbb{C}^d \to \mathbb{C}^N$ and bias term $b \in \mathbb{C}$ are also defined as complex quantities.

The LSSVM regression model is derived by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w},e} \quad & J_p(\mathbf{w}, \boldsymbol{e}) = \frac{1}{2}\mathbf{w}^H\mathbf{w} + \frac{\gamma}{2}\sum_{k=1}^{K} e_k e_k^* \\ s.t. \quad & \Re\{\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_k) + b + e_k\} = \mathrm{y}_{k,r} \\ & \Im\{\mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_k) + b + e_k\} = \mathrm{y}_{k,i} \end{aligned} \tag{3.17}$$

for $k = 1, \ldots, K$. In the above equation, the vector of model error $\boldsymbol{e} = \{e_k\}_{k=1}^K$ which is observed by the LSSVM model is equal to the true value y_k minus the model output of (3.16) computed at $\mathbf{x}_k$, for all training points. In the same way as the traditional LSSVM regression, problem (3.17) minimizes the ℓ_2-norm of the primal space model coefficients $\mathbf{w}$ plus the sum of squared errors, weighted by the regularization parameter γ. The optimal solution for this model is found by means of its Lagrangian, which writes:

$$\begin{aligned}\mathscr{L}(\mathbf{w}, b, \mathbf{e}; \boldsymbol{\alpha}) = \frac{1}{2}\left(\mathbf{w}_r^T\mathbf{w}_r + \mathbf{w}_i^T\mathbf{w}_i\right) + \gamma \sum_{k=1}^{K}\left(e_{k,r}^2 + e_{k,i}^2\right) \\ - \sum_{k=1}^{K}\alpha_{k,r}\left(\mathbf{w}_r^T\boldsymbol{\varphi}_r(\mathbf{x}_k) - \mathbf{w}_i^T\boldsymbol{\varphi}_i(\mathbf{x}_k) + b_r + e_{k,r} - \mathrm{y}_{k,r}\right) \\ - \sum_{k=1}^{K}\alpha_{k,i}\left(\mathbf{w}_r^T\boldsymbol{\varphi}_i(\mathbf{x}_k) + \mathbf{w}_i^T\boldsymbol{\varphi}_r(\mathbf{x}_k) + b_i + e_{k,i} - \mathrm{y}_{k,i}\right),\end{aligned} \tag{3.18}$$

where most quantities from Sec. 3.1 were split into their real and imaginary parts in order to facilitate the computations, e.g., $\alpha_k = \alpha_{k,r} + j\,\alpha_{k,i}$, $\mathbf{w} = \mathbf{w}_r + j\,\mathbf{w}_i$, $e_k = e_{k,r} + j\,e_{k,i}$, $\mathrm{y}_k = \mathrm{y}_{k,r} + j\,\mathrm{y}_{k,i}$ and $\boldsymbol{\varphi}(\mathbf{x}_k) = \boldsymbol{\varphi}_r(\mathbf{x}_k) + j\,\boldsymbol{\varphi}_i(\mathbf{x}_k)$. The derivatives of this Lagrangian are equal to:

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}_r} = \frac{1}{2}(2\mathbf{w}_r) - \sum_{k=1}^{K}\alpha_{k,r}\boldsymbol{\varphi}_r(\mathbf{x}_k) - \sum_{k=1}^{K}\alpha_{k,i}\boldsymbol{\varphi}_i(\mathbf{x}_k), \tag{3.19a}$$

$$\frac{\partial\mathscr{L}}{\partial\mathbf{w}_i} = \frac{1}{2}(2\mathbf{w}_i) + \sum_{k=1}^{K}\alpha_{k,r}\boldsymbol{\varphi}_i(\mathbf{x}_k) - \sum_{k=1}^{K}\alpha_{k,i}\boldsymbol{\varphi}_r(\mathbf{x}_k), \tag{3.19b}$$

$$\frac{\partial\mathscr{L}}{\partial b_r} = -\sum_{k=1}^{K}\alpha_{k,r}, \tag{3.19c}$$

$$\frac{\partial\mathscr{L}}{\partial b_i} = -\sum_{k=1}^{K}\alpha_{k,i}, \tag{3.19d}$$

$$\frac{\partial\mathscr{L}}{\partial e_{k,r}} = -\alpha_{k,r} + 2\frac{\gamma}{2}e_{k,r}, \tag{3.19e}$$

$$\frac{\partial\mathscr{L}}{\partial e_{k,i}} = -\alpha_{k,i} + 2\frac{\gamma}{2}e_{k,i}, \tag{3.19f}$$

$$\frac{\partial\mathscr{L}}{\partial\alpha_{k,r}} = -(\mathbf{w}_r^T\boldsymbol{\varphi}_r(\mathbf{x}_k) - \mathbf{w}_i^T\boldsymbol{\varphi}_i(\mathbf{x}_k) + b_r + e_{k,r} - \mathrm{y}_{k,r}), \tag{3.19g}$$

$$\frac{\partial\mathscr{L}}{\partial\alpha_{k,i}} = -(\mathbf{w}_r^T\boldsymbol{\varphi}_i(\mathbf{x}_k) + \mathbf{w}_i^T\boldsymbol{\varphi}_r(\mathbf{x}_k) + b_i + e_{k,i} - \mathrm{y}_{k,i}). \tag{3.19h}$$

Making these derivatives equal to zero results in the definitions below:

$$\mathbf{w} = \sum_{k=1}^{K}\alpha_k\boldsymbol{\varphi}^*(\mathbf{x}_k), \tag{3.20}$$

$$\sum_{k=1}^{K}\alpha_k = 0, \tag{3.21}$$

$$e_k = \frac{\alpha_k}{\gamma}, \tag{3.22}$$

$$\mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}_k) + b + e_k - \mathrm{y}_k = 0. \tag{3.23}$$

Similarly to the previous section, by combining the equations in (3.23) and (3.21), separating its real and imaginary parts, and substituting in (3.23) the value for $\mathbf{w}$ and e_k in (3.20) and (3.22), one can build the following linear system of equations:

$$\begin{bmatrix} \boldsymbol{\Omega} + \mathbf{I}_{2K}/\gamma & \begin{matrix} \mathbf{1}_K & \mathbf{0}_K \\ \mathbf{0}_K & \mathbf{1}_K \end{matrix} \\ \begin{matrix} \mathbf{1}_K^T & \mathbf{0}_K^T \\ \mathbf{0}_K^T & \mathbf{1}_K^T \end{matrix} & \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_r \\ \boldsymbol{\alpha}_i \\ b_r \\ b_i \end{bmatrix} = \begin{bmatrix} \mathbf{y}_r \\ \mathbf{y}_i \\ 0 \\ 0 \end{bmatrix}. \tag{3.24}$$

The $2K \times 2K$ kernel matrix $\boldsymbol{\Omega}$ is defined as

$$\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{\Omega}^{(1,1)} & \boldsymbol{\Omega}^{(1,2)} \\ \boldsymbol{\Omega}^{(2,1)} & \boldsymbol{\Omega}^{(2,2)} \end{bmatrix}. \tag{3.25}$$

In the $K \times K$ square submatrices that compose $\boldsymbol{\Omega}$, an element $\Omega_{i,j}^{(.,.)}$ in the i-th row and j-th column is defined as follows:

$$\Omega_{i,j}^{(1,1)} = \mathfrak{R}\{\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)\}, \tag{3.26a}$$

$$\Omega_{i,j}^{(1,2)} = -\mathfrak{I}\{\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)\}, \tag{3.26b}$$

$$\Omega_{i,j}^{(2,1)} = \mathfrak{I}\{\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)\}, \tag{3.26c}$$

$$\Omega_{i,j}^{(2,2)} = \mathfrak{R}\{\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)\}, \tag{3.26d}$$

and complex Gram matrix

$$\boldsymbol{\Omega}_{\mathbb{C}} = \boldsymbol{\Omega}^{(1,1)} + j\,\boldsymbol{\Omega}^{(2,1)}, \tag{3.27}$$

for $i, j = 1, \ldots, K$.

The kernel function now is a complex function $\mathrm{k}(\cdot,\cdot) : \mathbb{C}^d \times \mathbb{C}^d \to \mathbb{C}$, defined as:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_j) \right\rangle. \tag{3.28}$$

The complex-valued kernel is even more essential for the model estimation than its real-valued counterpart, as it will carry the relation between real and imaginary parts of the data. This kernel trick that replaces the feature space by its corresponding reproducing kernel Hilbert space (RKHS) allows the computation of the LSSVM based on the dual space formulation, without the specific knowledge about its feature space. The unknowns $\boldsymbol{\alpha} = \boldsymbol{\alpha}_r + j\,\boldsymbol{\alpha}_i = [\alpha_{1,r} + j\,\alpha_{1,i}, \ldots, \alpha_{K,r} + j\,\alpha_{K,i}]^T$ and $b = b_r + j\,b_i$ are computed by

solving (3.24). Similarly to the real-valued LSSVM regression, the determination of the previous parameters leads to the dual space representation of the regression, written as

$$\tilde{y}(\mathbf{x}) = \sum_{k=1}^{K} \alpha_k k(\mathbf{x}, \mathbf{x}_k) + b. \tag{3.29}$$

Once the dual space model is determined for known basis, any of its primal space coefficients can be obtained from (3.20), and thus also its equivalent primal space model. For a single coefficient w_i that weights the basis $\varphi_i(\mathbf{x})$ in the primal space model in (3.16), this computation writes:

$$w_i = \sum_{k=1}^{K} \alpha_k \varphi_i^*(\mathbf{x}_k), \tag{3.30}$$

and thus

$$\mathbf{w} = \sum_{k=1}^{K} \alpha_k \boldsymbol{\varphi}^*(\mathbf{x}_k). \tag{3.31}$$

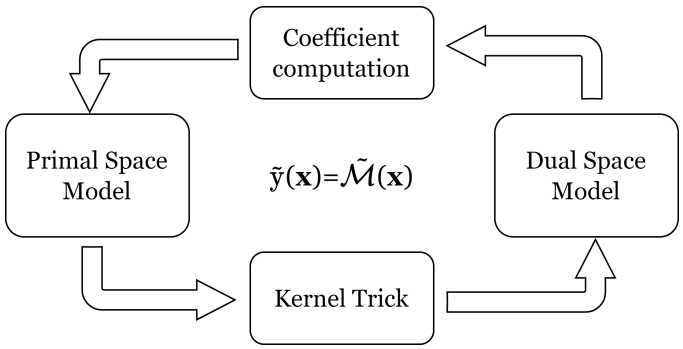


Figure 3.2: Diagram showing the duality of the LSSVM regression model.

With this coefficient computation and the kernel trick, the duality of the LSSVM can be optimally exploited. A LSSVM regression model based on a feature space with a very large (or even infinte) number of basis N can be computed in its dual space formulation in (3.29), via a predefined or ad-hoc kernel estimated via the kernel trick in (3.28), by estimating the K parameters $\boldsymbol{\alpha}$. This nonparametric dual space formulation is equivalent to the parametric model in (3.16), which can be computed via the knowledge of the feature space and the trick in (3.30). Figure 3.2 shows a diagram exemplifying this duality.

3.3 The Kernel Function

The previous sections developed a LSSVM regression model, where two representations were available: the primal and the dual space models. The primal space formulation

in (3.16) expressed the model based on a feature space $\boldsymbol{\varphi}(\mathbf{x})$, while the dual space formulation is a model that relies on the kernel function $\mathrm{k}(\mathbf{x}, \mathbf{x}_i)$. The name *kernel* comes from the field of integral operator theory, which supports much of the theory about the relation between kernels and their feature spaces. Indeed, this kernel-based representation projects data into a high dimensional space, increasing the computational power of linear regression models [58]. By using an adequate kernel, it is possible to make implicitly a non-linear map into a high dimensional feature space.

The parametric learning procedure selects appropriate features based on empirical knowledge to represent data and estimate a linear model. Feature selection is an essential part of that procedure, and one kind of feature selection is the principal component analysis (PCA), which computes features as a linear combination of original attributes and sort them by the amount of variance that the data exhibits in each direction. Kernel-based techniques are an alternative feature selection procedure, by performing a non-linear map that is implicitly based on a set of features. With the use of these kernel techniques, the non-parametric learning algorithm and theory can be decoupled from the specific application. The details of the application are encoded in the choice or design of an appropriate kernel function [58].

The kernel function directly computes the inner product of features as a function of a pair of input points, as stated in (3.28). The dimension of the feature space does not need to affect the computational time, and thus, the kernel function may be used as a tool to avoid the computational issues inherent to the evaluation of the feature map. The only information needed by kernel-based learning algorithms is the kernel matrix $\boldsymbol{\Omega}$, also called Gram matrix [58].

It is not required to know the feature space that corresponds to a certain kernel function, as long as we know that it has a reproducing kernel Hilbert space (RKHS) of functions for which it corresponds to the kernel. The next subsection describes some techniques and characteristics required to ensure that a predefined function is indeed a kernel function.

3.3.1 Using Predefined Kernels

The use of predefined kernels avoids the definition of the feature space. This can be seen as a more natural way to deal with the model, as the focus can be shifted to the learning algorithm. Indeed, if a computationally efficient kernel function is defined, the learning algorithm will be more easily extended to larger data sets, as a large part of the computational cost consists on the computation of the kernel matrix.

The characteristics that a kernel should possess can start to be derived from the Cauchy-Schwarz inequality [58]:

$$\left\langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_j) \right\rangle^2 \leq \left\langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_i) \right\rangle \left\langle \boldsymbol{\varphi}(\mathbf{x}_j), \boldsymbol{\varphi}(\mathbf{x}_j) \right\rangle . \tag{3.32}$$

valid for a real-valued feature space. In the case of complex-valued functions, the magnitude of this inner product should be considered instead:

$$\left|\left\langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_j)\right\rangle\right|^2 \leq \left\langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_i)\right\rangle \left\langle \boldsymbol{\varphi}(\mathbf{x}_j), \boldsymbol{\varphi}(\mathbf{x}_j)\right\rangle . \tag{3.33}$$

From the above inequality, an upper bound for the magnitude is defined:

$$|\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)|^2 \leq \mathrm{k}(\mathbf{x}_i, \mathbf{x}_i)\, \mathrm{k}(\mathbf{x}_j, \mathbf{x}_j). \tag{3.34}$$

It is possible also to easily define a symmetry that should be observed by the kernel function [92]:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_j)\right\rangle = \left\langle \boldsymbol{\varphi}(\mathbf{x}_j), \boldsymbol{\varphi}(\mathbf{x}_i)\right\rangle^* = \mathrm{k}(\mathbf{x}_j, \mathbf{x}_i)^*, \tag{3.35}$$

and a positive value for the kernel computed for a given input pair $\{\mathbf{x}_i, \mathbf{x}_i\}$:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_i) = \langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_i)\rangle = \|\boldsymbol{\varphi}(\mathbf{x}_i)\|^2 \geq 0. \tag{3.36}$$

The conditions above are useful, but not sufficient to guarantee that a given function has a corresponding feature space. A stronger test comes from Mercer's condition, which writes [60], [93]:

$$\iint_{\mathcal{X}\times\mathcal{X}} \mathrm{k}(\mathbf{x}, \mathbf{x}')g(\mathbf{x})g(\mathbf{x}')^* d\mathbf{x} d\mathbf{x}' \geq 0, \tag{3.37}$$

computed over the input domain $\mathcal{X}$ for any square integrable function $g(\mathbf{x})$. In practical terms, this means that the kernel function should have a Gram matrix which is positive semi-definite, i.e., $\boldsymbol{\Omega}$, for the real-valued case, or its Hermitian part $\boldsymbol{\Omega}_H = (\boldsymbol{\Omega}_{\mathbb{C}} + \boldsymbol{\Omega}_{\mathbb{C}}^H)/2$, for the complex-valued case, have all eigenvalues non-negative. When computed over the training samples and for any vector $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_K]^T \in \mathbb{R}^K$, Mercer's condition is given by:

$$\boldsymbol{\lambda}^T \boldsymbol{\Omega}_H \boldsymbol{\lambda} \geq 0. \tag{3.38}$$

When the kernel satisfies (3.37), it is called a *Mercer kernel*. For every Mercer kernel $\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)$, there is a RKHS of functions defined over the domain of $\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)$ for which $\mathrm{k}(\cdot, \cdot)$ is the reproducing kernel [58].

The above condition presents a large similarity to the correlation functions used in the Gaussian process regression (GPR) [62] and similar stochastic models. Indeed, correlation functions are also Mercer kernels. The covariance matrices are positive definite for all finite sets of input points, which is precisely the defining property of the Mercer condition. The definition of the GPR with a given correlation function avoids defining explicitly the function class and prior distribution over those functions [58].

3.3.2 Making a kernel

The converse of the conclusions above is also true: for any Hilbert space of functions in which the evaluated functionals are bounded and linear, there exists a reproducing kernel function [58]. Therefore, another approach in choosing the appropriate kernel for the kernel-based model is to derive said kernel from the inner product of the desired feature space $\boldsymbol{\varphi}(\mathbf{x}) = [\varphi_1(\mathbf{x}), \dots, \varphi_N(\mathbf{x})]^T$. This is a powerful approach in special situations, e.g., for working in non-euclidean spaces such as sets of finite strings. When this route is taken, there is no need to verify Mercer's condition, as the feature space corresponding to the kernel exists and is known.

Given the above set of desirable features, their RKHS can be simply computed by expanding the inner product [60]

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{n=1}^{N} \lambda_n \varphi_n(\mathbf{x}_i) \varphi_n^*(\mathbf{x}_j), \tag{3.39}$$

in which the positive constants $\{\lambda_n\}_{n=1}^N$ were added for weighting individual features. This kernel is valid also for $N \to \infty$, if the resulting $\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)$ is bounded.

This reasoning can be expanded to the case where $\boldsymbol{\varphi}(\mathbf{x})$ is defined as the set of functions $\varphi(\mathbf{x}; \vartheta)$ with $\vartheta \in \Theta$. If Θ is defined as a continuous space, its RHKS is given by [82]

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \int_{\Theta} \varphi(\mathbf{x}_i; \vartheta) \varphi^*(\mathbf{x}_j; \vartheta) d\vartheta. \tag{3.40}$$

A simpler way to make a kernel is by combining other known kernels. Assuming that $k_a(\mathbf{x}_i, \mathbf{x}_j)$ and $k_b(\mathbf{x}_i, \mathbf{x}_j)$ are kernels, the following properties hold true in making a new kernel $k'(\mathbf{x}_i, \mathbf{x}_j)$ [60]:

$$k'(\mathbf{x}_i, \mathbf{x}_j) = a k_a(\mathbf{x}_i, \mathbf{x}_j), \tag{3.41a}$$

$$k'(\mathbf{x}_i, \mathbf{x}_j) = k_a(\mathbf{x}_i, \mathbf{x}_j) + b, \tag{3.41b}$$

$$k'(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{P} \mathbf{x}_j^*, \tag{3.41c}$$

$$k'(\mathbf{x}_i, \mathbf{x}_j) = k_a(\mathbf{x}_i, \mathbf{x}_j) + k_b(\mathbf{x}_i, \mathbf{x}_j), \tag{3.41d}$$

$$k'(\mathbf{x}_i, \mathbf{x}_j) = k_a(\mathbf{x}_i, \mathbf{x}_j) k_b(\mathbf{x}_i, \mathbf{x}_j), \tag{3.41e}$$

$$k'(\mathbf{x}_i, \mathbf{x}_j) = k_a(\boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_j)), \tag{3.41f}$$

$$k'(\mathbf{x}_i, \mathbf{x}_j) = p_+(k_a(\mathbf{x}_i, \mathbf{x}_j)), \tag{3.41g}$$

$$k'(\mathbf{x}_i, \mathbf{x}_j) = \exp(k_a(\mathbf{x}_i, \mathbf{x}_j)), \tag{3.41h}$$

where $a, b > 0$, $P = P^H > 0$, and $p_+(\cdot)$ is a polynomial with positive coefficients.

Another operation that can be performed on kernels are their normalization:

$$\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathrm{k}_a(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{\mathrm{k}_a(\mathbf{x}_i, \mathbf{x}_i)}\sqrt{\mathrm{k}_a(\mathbf{x}_j, \mathbf{x}_j)}}. \tag{3.42}$$

The above normalization is equivalent to computing the cosine of the angle between $\boldsymbol{\varphi}(\mathbf{x}_i)$ and $\boldsymbol{\varphi}(\mathbf{x}_j)$ in the feature space. It can be seen as a special case of the conformal transformation of the kernel with factor $c(\mathbf{x})$ [60]:

$$\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j) = c(\mathbf{x}_i)\mathrm{k}_a(\mathbf{x}_i, \mathbf{x}_j)c(\mathbf{x}_j). \tag{3.43}$$

By combining all the above properties with known predefined kernels, one should be able to find a suitable kernel for the desired application.

3.3.3 Commonly used kernel functions

Since the introduction and popularization of the SVM framework [82], several kernel functions have been proposed. This section lists some of them below. Each kernel function depends on a pair of inputs $\left(\mathbf{x}_i, \mathbf{x}_j\right)$, and possibly on other additional parameters used to adjust them. Those are called hyperparameters, which will be described better in the next section.

The three most common kernel functions are the linear, polynomial and Gaussian, or radial basis functions (RBF) kernels. Standard LSSVM implementations will use one of these three functions [60]. They are written by:

Linear

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_j^T \mathbf{x}_i \tag{3.44}$$

Polynomial

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \left(c + \mathbf{x}_j^T \mathbf{x}_i\right)^d \tag{3.45}$$

Gaussian

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{3.46}$$

Nonetheless, many other Mercer kernels can be used. Some other alternatives presented in the literature [94] are listed below:

Exponential:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{2\sigma^2}\right) \tag{3.47}$$

ANOVA

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{n=1}^{N} \exp\left(-\sigma\left(\mathbf{x}_i^n - \mathbf{x}_j^n\right)^2\right)^d \tag{3.48}$$

Rational Quadratic

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\|\mathbf{x}_i - \mathbf{x}_j\|^2 + c} \tag{3.49}$$

Inverse Multiquadric

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|^2 + c^2}} \tag{3.50}$$

Bessel

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \frac{J_{\nu+1}(\sigma\|\mathbf{x}_i - \mathbf{x}_j\|)}{\|\mathbf{x}_i - \mathbf{x}_j\|^{-n(\nu+1)}} \tag{3.51}$$

Cauchy

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{1 + \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}} \tag{3.52}$$

Many of the above kernels depend only on the distance $(\mathbf{x}_i - \mathbf{x}_j)$. Such kernels are said to be *stationary*, and particularly, isomorphic stationary if they depend only on the magnitude of this distance. Many fields assume stationary problems, and thus, might benefit from this sort of kernel [95].

Another class of kernels are the *conditionally positive definite* kernels. Such kernels are positive definite only if the set constants λ in (3.38) is chosen such that $\sum_{i=1}^{K} \lambda_i = 0$. They are of particular interest because a Mercer kernel $\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j)$ is associated to any conditionally positive definite kernel $\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j)$ via [96]

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}\left[\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j) - \mathrm{k}'(\mathbf{x}_i, \mathbf{x}_0) - \mathrm{k}'(\mathbf{x}_j, \mathbf{x}_0) + \mathrm{k}'(\mathbf{x}_0, \mathbf{x}_0)\right], \tag{3.53}$$

for any $\mathbf{x}_0$ in the input space. Given the property in (3.53), most kernel-based modeling techniques can be applied also with conditionally positive kernels. Some examples of this class of kernel are given by [94]:

Sigmoid

$$\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\alpha \mathbf{x}_i^T \mathbf{x}_j + c\right) \tag{3.54}$$

Multiquadratic

$$\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\|\mathbf{x}_i - \mathbf{x}_j\|^2 + c^2} \tag{3.55}$$

Wave

$$\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j) = \frac{\theta}{\|\mathbf{x}_i - \mathbf{x}_j\|} \sin\left(\frac{\theta}{\|\mathbf{x}_i - \mathbf{x}_j\|}\right) \tag{3.56}$$

Power

$$\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j) = -\|\mathbf{x}_i - \mathbf{x}_j\|^d \tag{3.57}$$

Log

$$\mathrm{k}'(\mathbf{x}_i, \mathbf{x}_j) = -\log\left(\|\mathbf{x}_i - \mathbf{x}_j\|^d + 1\right) \tag{3.58}$$

A less explored field concerns the category of *complex-valued kernels*. The most simple kernel of this category is the basic complex kernel [92], [97]:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \cos\left(\mathbf{c}^T\left(\mathbf{x}_i - \mathbf{x}_j\right)\right) + j\sin\left(\mathbf{c}^T\left(\mathbf{x}_i - \mathbf{x}_j\right)\right), \tag{3.59}$$

in which the vector of hyperparameters $\mathbf{c}$ can be simplified to $\mathbf{c} = \omega_0 \mathbf{1}_d$ to reduce the quantity of hyperparameters. Such simplistic kernel can be expanded to a more elaborate structure by the use of the properties from Section 3.3.2, specially (3.41e). Another way to build a complex-valued kernel from a real-valued one $\mathrm{k}_R(\mathbf{x}'_i, \mathbf{x}'_j)$ is by separating the real and imaginary parts of the input as $\mathbf{x}_k = \mathbf{x}'_k + j\mathbf{x}''_k \; \forall \, k$ and building an independent kernel from them [86]:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{k}_R(\mathbf{x}'_i, \mathbf{x}'_j) + \mathrm{k}_R(\mathbf{x}''_i, \mathbf{x}''_j) + j\left(\mathrm{k}_R(\mathbf{x}'_i, \mathbf{x}''_j) - \mathrm{k}_R(\mathbf{x}''_i, \mathbf{x}'_j)\right). \tag{3.60}$$

Additionally, a complex kernel can be built from a certain feature space using the inner product expansions in (3.39) and (3.40). Nonetheless, the go-to Gaussian kernel is also available in a complex-valued form [84]:

$$\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\left(\mathbf{x}_i - \mathbf{x}_j^*\right)^T\left(\mathbf{x}_i - \mathbf{x}_j^*\right)/\sigma^2\right). \tag{3.61}$$

Some other classes of kernels might be useful for their specific applications. Two of them are briefly discussed below [95]:

- When dealing with large data sets, *compactly supported kernels* offer a good advantage, as their output is zero when $\|\mathbf{x}_i - \mathbf{x}_j\| > \Delta_x$, i.e., the distance between the input points is larger than a threshold Δ_x, and thus they produce a sparse Gram matrix.

- Also useful for computations over large data sets are *separable nonstationary kernels*, in which the kernel function can be separate into a product of functions which depend on only one input variable as $\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{k}_a(\mathbf{x}_i)\mathrm{k}_b(\mathbf{x}_j)$. For such kernels, the Gram matrix can be computed through a Kronecker product, reducing the computational time.

3.4 Tuning of the Model Hyperparametters

Most of the kernel functions described above have one or more parameters that tune their shape and which cannot be directly estimated by the regression problem, also called *hyperparameters*. The Gaussian kernel, for example, has the parameter σ which controls the width of its Gaussian: when σ is large, the kernel produces a significant output at the coordinates $\mathbf{x}$ distant from $\mathbf{x}_k$, while if σ is small, the kernel output in negligible far from the training point $\mathbf{x}_k$. This kind of balance provided by the kernel function must be tuned, together with other parameters from the kernel-based method in use, e.g., the regularizer γ in the LSSVM regression from Section 3.1. The parameters that need to be tuned can be grouped in a generic vector of hyperparameters $\boldsymbol{\lambda}$. The accuracy of the modeling scheme unavoidably depends on the tuning of $\boldsymbol{\lambda}$, which is given by $\boldsymbol{\lambda} = [\gamma, \sigma]^T$ in the case of a LSSVM model with a Gaussian kernel.

Above all, it is desirable to perform the tuning of those hyperparameters in an automatic procedure, in a way applicable for any model. A procedure for this tuning can be described by the diagram in Fig. 3.3, and it starts by defining a candidate value for the hyperparameters $\boldsymbol{\lambda}_c$. With this value, a ML model is estimated and its performance is assessed via a predefined performance metric. This is done in a cyclic and algorithmic way, updating the value of the candidate parameters at each step of the cycle, with the goal of finding the optimal tuning parameters $\boldsymbol{\lambda}^\star$. Proven approaches used for this are grid and simplex searches [98], genetic algorithms [69] or Bayesian optimization [99].

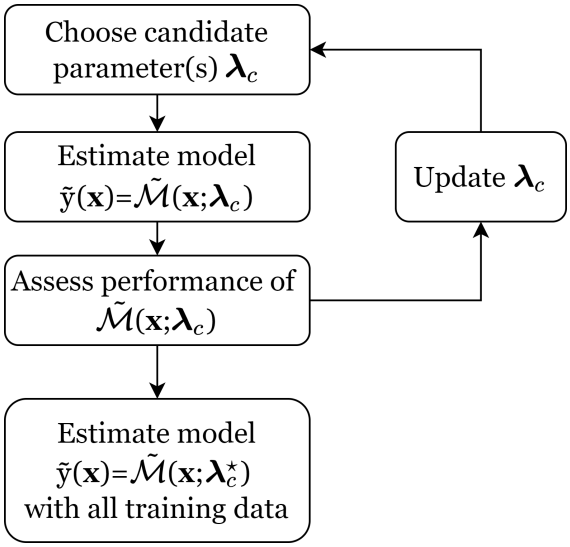


Figure 3.3: Cycle for the optimization of hyperparameters and definition of model.

The use of Bayesian optimization in order to find $\boldsymbol{\lambda}^\star$ which produces the best model performance is widely used with ML techniques. This algorithm specifically selects the next candidate configuration of the hyperparameters $\boldsymbol{\lambda}_c$ in order to maximize the reward towards finding the global optimum of a non-convex function [100]. By using

this sampling scheme, the search converges to the optimum solution in fewer iterations than with standard search schemes (e.g., grid search), while keeping the capacity to avoid local minima. An open-source implementation of this optimization scheme is available in the scikit-optimize Python library [98].

Nonetheless, the optimality of the estimated model relies heavily on the performance estimation technique which is applied. The estimation via the aparent model error computed over the training samples is often too optimistic [69], i.e., they might provide a model error estimation smaller than it should be due to the model overfitting the training data. A better approach is to test the model and assess its performance on samples not used for training it, by dividing the available data into a *training* and a *validation* data set. Then, the training data is used to estimate the model, and the validation data assess its performance in order to tune $\boldsymbol{\lambda}$ and arrive at the optimized value $\boldsymbol{\lambda}^{\star}$. However, the size of this validation set might need to be large for a precise performance estimation, which is not appropriate, since samples might be costly and needed for model building. A technique known as k-fold CV [101] can be seen as promising candidates to overcome the above issues.

In the k-fold cross-validation (CV), the training set is split into k smaller sets, called *folds*. Then, for each of the k folds, the model is trained using $k-1$ folds as training data and by using a given configuration of the hyperparameters $\boldsymbol{\lambda}$, while the remaining fold is kept as a validation set used to evaluate the model accuracy on data which were not used during the training. The above scheme is iterated for all the k-folds. Then, for each analyzed combination of the hyperparameters $\boldsymbol{\lambda}$, the overall model performance is assessed by the k-fold CV error $CV_{\text{error}}^{(\boldsymbol{\lambda})}$, which is the average of the values computed during the k iterations, according to

$$CV_{\text{error}}^{(\boldsymbol{\lambda})} = \frac{1}{k}\sum_{n=1}^{k} CV_{\text{error},n}^{(\boldsymbol{\lambda})}, \tag{3.62}$$

where $CV_{\text{error},n}^{(\boldsymbol{\lambda})}$ is the MSE of the model on the n-th test fold, or any other metric appropriate for the studied problem. The common choice for k is usually 5 or 10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance. When $k = 1$ the k-fold validation is called leave-one-out cross-validation. An illustration of the k-fold CV for the case with $K = 21$ training samples and 3 folds (i.e., $k = 3$) is shown in Fig. 3.4.

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

$$\boldsymbol{\lambda}^{\star} = \arg\min_{\boldsymbol{\lambda}} CV_{\text{error}}^{(\boldsymbol{\lambda})}. \tag{3.63}$$

After that, as indicate in the bottom block of Fig. 3.3, the optimal hyperparameters are used to train a model where all training data is used as training set, with no validation

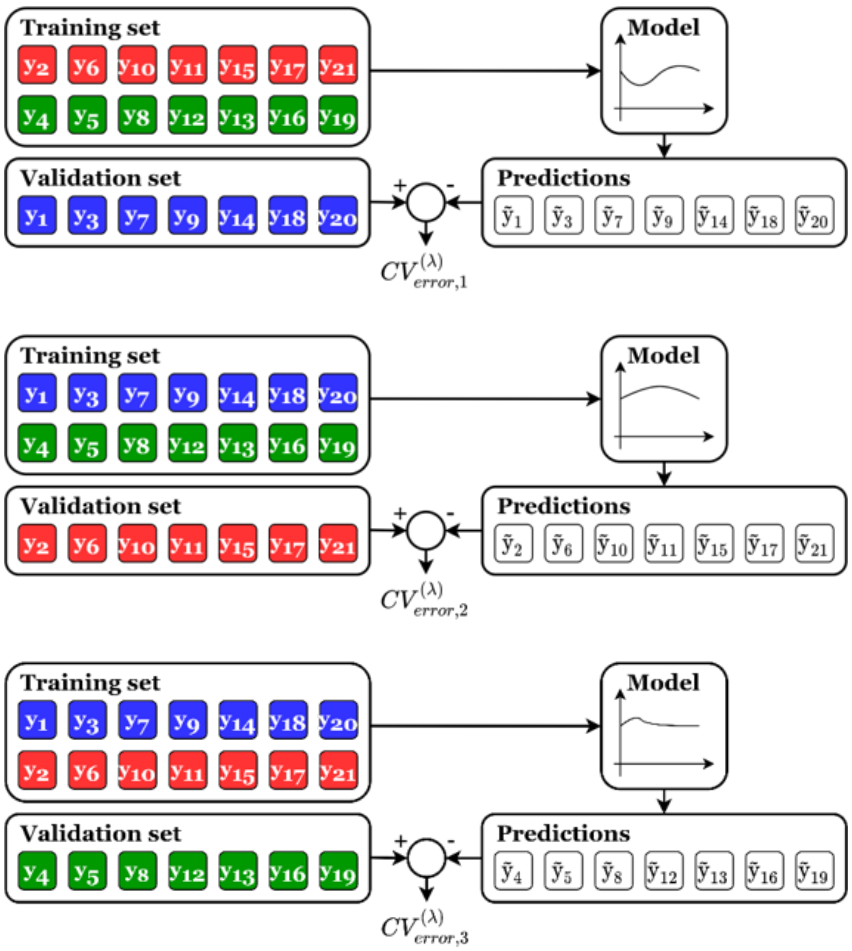


Figure 3.4: Example: a k-fold cross-validation with $k = 3$ and $K = 21$ training samples.

set. However, a small external test data set might still be used to better prove the accuracy of the model. When these steps are finished, this model can be used robustly in its target application.

3.5 Modeling Examples

The kernel-based techniques presented above are widely used by the scientific community, and examples can be found in the literature [61], [91], [102]. On the other hand, the use of a custom kernel and the duality from Fig. 3.2 is less exploited. This trick is the main tool for the development of the next chapter, and therefore, in order to introduce said technique, two examples on a familiar framework are presented in this section, namely, the fast Fourier transform (FFT) and the discrete-time Fourier Transform (DTFT). Kernel-based models which are equivalent to those transform are

built and used to compute their coefficients in an alternative way to their standard formulation. These are just toy examples, without a solid advantage over their well-tested alternatives, but they provide the foundation of the techniques used in the next chapter.

3.5.1 Kernel-Based Fast Fourier Transform

The discrete Fourier transform, also known as FFT due to the most famous algorithm employed for its computation, is a transform that computes the frequency spectrum of a periodic and discrete time signal. In this transform, the coefficients $Y(k)$ at discrete frequency k are computed from the N_s samples of the signal $y(n)$ according to [103]:

$$Y(k) = \sum_{n=0}^{N_s-1} y(n) e^{-j2\pi kn/N_s}. \tag{3.64}$$

Its inverse writes the original signal at the discrete time values n as a function of the frequency spectrum as:

$$y(n) = \frac{1}{N_s} \sum_{k=0}^{N_s-1} Y(k) e^{j2\pi kn/N_s}. \tag{3.65}$$

By analyzing (3.65), it is evident that $y(n)$ is a linear expansion of complex-exponential basis functions given by:

$$\varphi_k(n) = e^{j2\pi kn/N_s}, \tag{3.66}$$

for $k = [0, \dots, N_s - 1]$. The use of such bases allow (3.65) to be rewritten as:

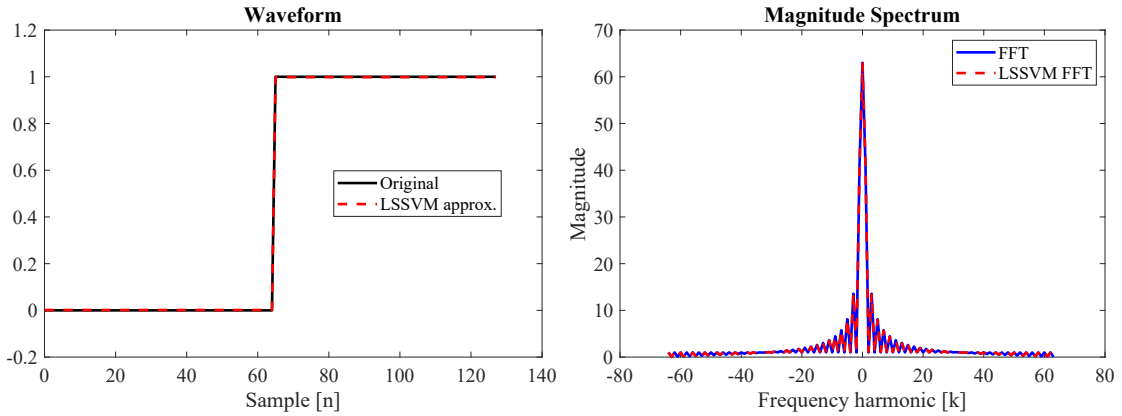
$$y(n) = \sum_{k=0}^{N_s-1} \mathrm{w}_k \varphi_k(n) = \sum_{k=0}^{N_s-1} \frac{Y(k)}{N_s} e^{j2\pi kn/N_s}. \tag{3.67}$$

A parallel can be made to the primal formulation of the LSSVM regression in (3.16): if the vector of basis functions is given by the basis defined in (3.66), the weights w_k are also equal to $Y(k)/N_s$ (the coefficient b acts as a constant basis which is also present in $\boldsymbol{\varphi}$). However, the LSSVM model has an equivalent dual space model given by (3.29), which is estimated via the kernel function from the features $\boldsymbol{\varphi}(n) = \left[\varphi_0(n), \dots, \varphi_{N_s-1}(n)\right]^T$ according to (3.39):

$$k(n_i, n_j) = \boldsymbol{\varphi}(n_i)^T \boldsymbol{\varphi}^*(n_j) \tag{3.68}$$

$$= \sum_{k=0}^{N_s-1} \left(e^{j2\pi kn_i/N_s}\right) \left(e^{-j2\pi kn_j/N_s}\right) \tag{3.69}$$

$$= \sum_{k=0}^{N_s-1} e^{j2\pi k(n_i-n_j)/N_s}. \tag{3.70}$$



(a) (b)

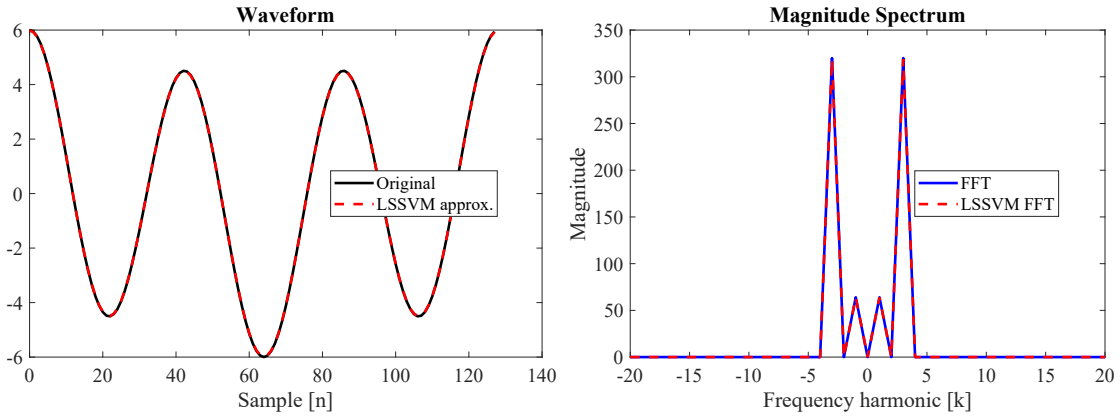
Figure 3.5: Comparison between the time-domain waveform and its LSSVM approximation (a), and comparison of the spectrum compute via the FFT and its LSSVM-based alternative (b), for a square wave.

With the above kernel, it is possible to estimate a non-parametric dual-space LSSVM regression model by using (3.13). The comparison between this estimated model and the original data for one period of a square wave is shown in Fig. 3.5(a). Additionally, the model can be used to compute the weights w_k corresponding to its parametric primal space formulation via (3.20), which by definition, would be proportional to the FFT frequency domain coefficients $Y(k)$. It is possible to see how this holds by comparing them. Figure 3.5(b) shows the comparison of the magnitude spectrum produced by analyzing the primal space model of the LSSVM regression and by directly applying the FFT to the waveform, showing a remarkable similarity.

An additional example can be produced by applying the same methodology to the sum of a sinusoidal wave with two combined frequencies defined as $y(n) = \cos(2\pi n/N_s) + 5\cos(6\pi n/N_s)$. The corresponding spectrum should have peaks at the discrete frequencies $k = 1,3$, with different amplitudes. And this is exactly what is observed in Fig. 3.6, confirming both the accuracy of the model and the spectrum estimated via the LSSVM duality.

3.5.2 Kernel-Based Discrete-Time Fourier Transform

The FFT above computes the spectrum of a periodic waveform. Its equivalent when dealing with non-periodic waveforms is the DTFT, which, by considering that the period of the waveform tends to infinity, produces a continuum spectrum valid for any angular frequency ω. Indeed, like for the FFT, this spectrum is periodic due to the waveform being discrete. From this periodic spectrum $Y(\omega)$, the waveform $y(n)$ can be



(a) (b)

Figure 3.6: Comparison between the time-domain waveform and its LSSVM approximation (a), and comparison of the spectrum compute via the FFT and its LSSVM-based alternative (b), for a two-tone sinusoidal waveform.

written as [103]:

$$y(n) = \frac{1}{2\pi} \int_{2\pi} Y(\omega) e^{j\omega n} d\omega, \tag{3.71}$$

which is a linear expansion w.r.t. the basis functions

$$\varphi(n;\omega) = e^{j\omega n}. \tag{3.72}$$

Similar to the FFT, we could model $y(n)$ via a dual space kernel-based LSSVM regression model in (3.29). In order to do this, the appropriate kernel function for the features in (3.72) should be computed through (3.40). Such kernel function writes:

$$k(n_i, n_j) = \int_0^{2\pi} \varphi(n_i;\omega)\varphi^*(n_j;\omega) d\omega \tag{3.73}$$

$$= \int_0^{2\pi} e^{j\omega n_i} e^{-j\omega n_j} d\omega \tag{3.74}$$

$$= \int_0^{2\pi} e^{j\omega(n_i - n_j)} d\omega, \tag{3.75}$$

which is evaluated to

$$k(n_i, n_j) = 2\pi \,\mathrm{sinc}(n\pi). \tag{3.76}$$

The non-parametric dual-space LSSVM model is obtained by solving (3.13). The model primal space weights in (3.16) can be computed for the continuous interval $\omega \in [0,2\pi]$ used when making the kernel function. In doing this estimation, a weight w

would be computed for each value of ω, i.e., the function $\mathrm{w}(\omega)$ would be computed. By comparing the model (3.16) and the DTFT formulation in (3.71), it is evident that $\mathrm{w}(\omega) = Y(\omega)/2\pi$, and thus proportional to the spectrum of the the modeled waveform.

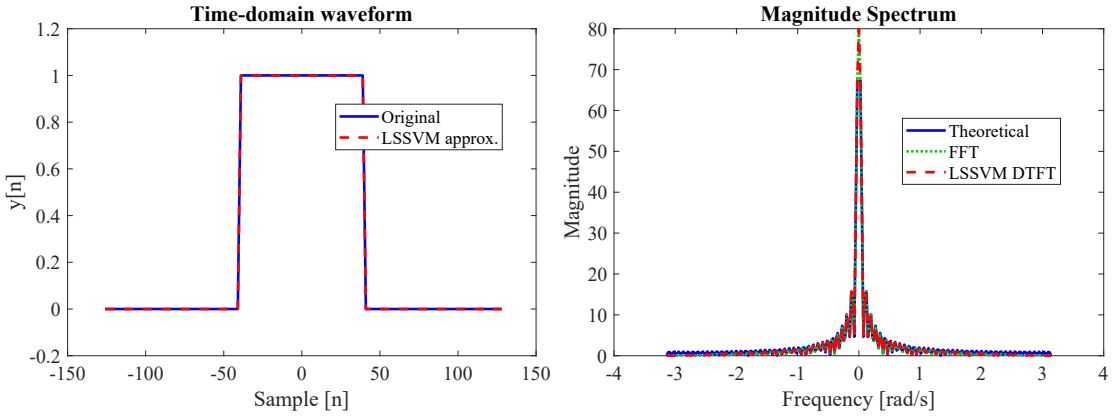


Figure 3.7: Comparison between the time-domain waveform and its LSSVM approximation (a), and comparison of the DTFT spectrum computed analytically, via the LSSVM-based alternative and via the FFT approximation (b), for a square pulse.

The described methodology is demonstrated in three examples: first, at a time limited pulse function defined as:

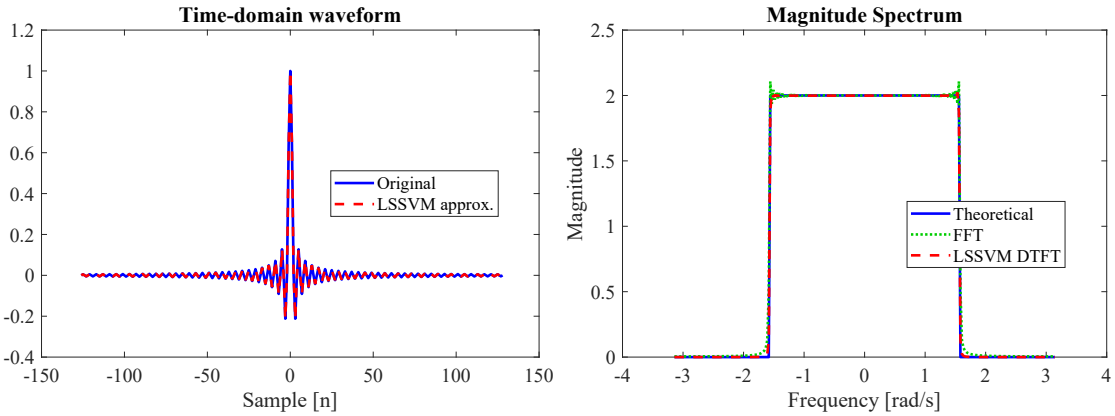
$$y(n) = \Pi\left(\frac{2n}{M}\right) = \begin{cases} 1, & |n| < M/2, \\ 1/2, & |n| = M/2, , \\ 0, & |n| > M/2. \end{cases} \tag{3.77}$$

with width $M = 80$. The above waveform is shown in Fig. 3.7(a). The spectrum of this pulse can be computed analytically and is given by:

$$Y(\omega) = \frac{\sin(M\omega/2)}{\sin(\omega/2)}. \tag{3.78}$$

The curve in Fig. 3.7-(a) shows that the LSSVM approximate model is accurate, and the magnitude spectrum in Fig. 3.7-(b), computed by multiplying $\mathrm{w}(\omega)$ by 2π, is accurate in comparison to the expected analytical spectrum and to the output of the FFT on the waveform (which is inaccurate due to considering the waveform periodic, but appropriate as a rough approximation).

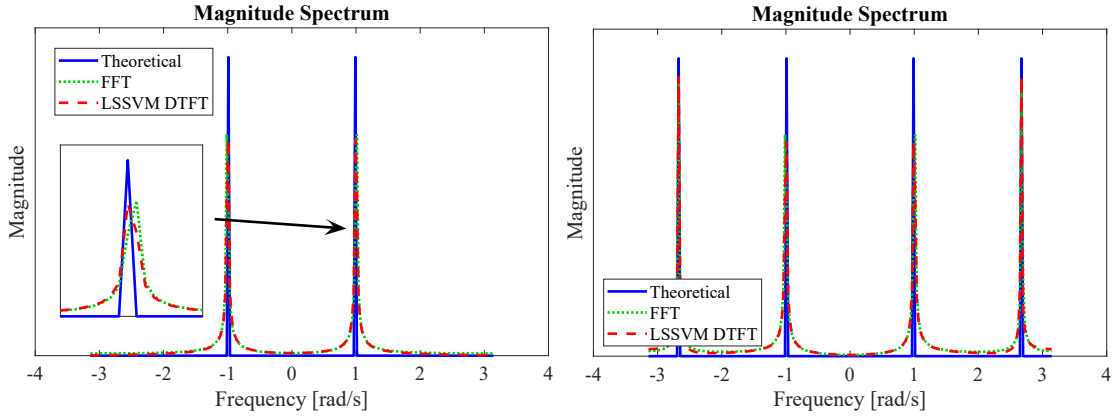
Then, the procedure is repeated for a time domain waveform given by $y(n) = \mathrm{sinc}(M\,n)$. The above waveform is shown in Fig. 3.7(a). This waveform tends to zero, but differently from the previous example, is not compactly supported, and its analytical DTFT is equal to a square pulse. In this example, a certain advantage is seen in the



(a) (b)

Figure 3.8: Comparison between the time-domain waveform and its LSSVM approximation (a), and comparison of the DTFT spectrum computed analytically, via the LSSVM-based alternative and via the FFT approximation (b), for a sinc($M\,n$) signal.

LSSVM DTFT method plotted in Fig. 3.7(b): while the FFT presents significant oscillations near the edges due to the Gibbs phenomenon [103], the LSSVM DTFT produces a spectrum very similar to the analytical solution.



(a) (b)

Figure 3.9: Comparison between the DTFT spectrum computed analytically, via the LSSVM-based alternative and via the FFT approximation, for a sinusoidal signal with one frequency (a), and two frequency components (b).

As a final example, the same methodology is applied to pure sinusoidal waveforms. Their corresponding spectrum is equal to the pair of Dirac's delta functions $\delta(\omega-\omega_0)+\delta(\omega+\omega_0)$ with infinite amplitude at the frequency $\omega = \omega_0$ and $\omega = -\omega_0$ for each sinusoidal component of frequency ω_0 present in the waveform. Despite the LSSVM-based DTFT being unable to produce such sharp peak at the desired frequency, Fig.

3.9(a) shows that it produces peaks at the desired frequencies, with a small accuracy gain over the FFT. This holds true also when the waveform has more than one sinusoidal component, e.g., in Fig. 3.9(b), where the DTFT of a two-tone sinusoidal signal is plotted. The LSSVM-based DTFT was also able to match the sharp impulses in the spectrum of a waveform with sinusoidal components at two different frequencies.

The examples above demonstrate the power of the duality in extracting useful information (the spectrum) from a LSSVM regression model. The next chapter takes this idea further and uses the same concepts in order to identify the time-delays that produced a certain frequency-response.

Chapter 4

Time-delay identification in frequency responses

Given the challenges in modeling long electrical interconnects described in chapters 1 and 2, and the LSSVM regression technique presented in chapter 3, this chapter merges them by applying the LSSVM regression to the modeling of a long electrical interconnect. Along with the resulting LSSVM model, an alternative scheme for the estimation of the dominant propagation delays within a delayed rational model (DRM) based on the duality of the LSSVM regression is obtained. The LSSVM regression with an ad-hoc kernel is adopted for these models. The proposed kernel has been built by considering an infinite dimensional feature space formed by an infinite number of delay-rational basis, which accounts for a continuous set of delay values within a certain interval. Thanks to the above features, the proposed method is able to identify the dominant propagation delays of the original system.

4.1 Rational models

The frequency response obtained from the linear CAD models, such as the models presented in section 2.2, usually consists on a finite number K of tabulated frequency-domain samples defined at the complex frequency points $\{s_1, \dots, s_K\} = \{j\omega_1, \dots, j\omega_K\}$. In order to be included on SPICE simulations, the data should be modeled in a way compatible with such simulations, i.e., in terms of standard linear circuit components or, if allowed by the simulation solver, in terms of a differential-algebraic system of equations (DAE) in the same way as the modified nodal analysis (MNA) DAE obtained from standard circuit components [12]. The usual way to extract this CAD model for SPICE simulations is via a rational (or VF) model [23], written as

$$H(s_k) \approx \tilde{H}(s_k) = r_0 + \sum_{j=1}^{n_p} \frac{r_j}{s_k - p_j}, \tag{4.1}$$

where the approximate model $\tilde{H}(s_k)$ approximates the tabular samples $H(s_k)$ via a linear expansion of rational basis $\varphi_p(s;p)$:

$$\tilde{H}(s_k) = r_0 + \sum_{j=1}^{n_p} r_j \varphi_p(s;p_j), \tag{4.2}$$

with the basis functions given by

$$\varphi_p(s;p) = \frac{1}{s-p}. \tag{4.3}$$

The rational basis above has a predefined parameter p. Each of the n_p basis in the expansion of (4.1) should have its pole p_j determined, along with its corresponding residue r_j and constant term r_0. The later, which have a linear appearance in the model, can be easily estimated via a linear regression [69]. However, the poles, which have a non-linear relation to the model, cannot be estimated as easily. Nonetheless, the well proven VF algorithm has been successfully employed for this task [23]. It uses a sequence of linear problems to find an appropriate set of poles through iterations. It has important properties such as high computational efficiency, good accuracy which can be further improved by adding additional poles, and a simple formulation [12]. The obtained pole-residue representation can be extracted as a space-state system of equations, or converted to circuital components [21], [22].

4.2 Delayed-Rational Models

In a DRM, the same model structure from the rational models is present, with the addition of an exponential term that explicitly represents the effects of time-delays in a frequency response of a system. Given a set of samples of the transfer function $H(s_k) \in \mathbb{C}$ with $k = 1 \ldots K$ of a long distributed structure at the complex frequencies $s_k \in \mathbb{C}$, the DRM has the following form:

$$H(s_k) \approx \tilde{H}(s_k) = r_0 + \sum_{i=1}^{n_\tau} \tilde{H}_i(s_k) e^{-s\tau_i}, \tag{4.4}$$

with,

$$\tilde{H}_i(s) = \sum_{j=1}^{n_{p,i}} \frac{r_{ij}}{s - p_{ij}}. \tag{4.5}$$

where each of the n_τ terms $\tilde{H}_i(s)$ corresponds to a rational model as the one in (4.1) with $n_{p,i}$ poles $p_{ij} = p'_{ij} + jp''_{ij} \in \mathbb{C}$, residues $r_{ij} = r'_{ij} + jr''_{ij} \in \mathbb{C}$ and a constant term $r_0 \in \mathbb{R}$. Each of the rational terms is associated to its corresponding time-delay term $\tau_i \in \mathbb{R}^+$, with only positive delay values considered in order to guarantee the causality of the modeled system.

The DRM requires the estimation of three categories of parameters: poles and residues for its rational terms, and the time-delays. The direct estimation of all those parameters at the same time is impractical, as suitable values are needed for the time-delays in the exponential term and for residues and poles in the numerator and denominator of the rational term. The identification of the dominant propagation delays from frequency response samples is an essential aspect for the achievement of a low-order DRM, since suitable approaches for the estimation of the rational part are available [44], [48]. Several of the methods available in the literature must also perform an optimization on the first identified set of delays in order to achieve the optimal model [33], [34], [42], [44]. If we are considering multiple delays, the optimization should be carried out in a high-dimensional space subject to local minima, making it a difficult task. Therefore, a precise estimation of the involved delays can provide an easier way to achieve a compact DRM.

After the time-delays are extracted, the rational parts of the DRM are easily estimated if the target transfer function can be split into the sum of delayed transfer functions. However, the separation of the transfer function into the parts associated to each delay is a challenging task, often based on the Gabor transform [26], [28], [44] or in time-domain segmentation [27], [43]. This complicate step can be avoided by assuming that each delay term is associated to the same set of poles. With this assumption, the DRM model $\tilde{H}(s)$ of (4.4) can be simplified as follows

$$\tilde{H}(s;\mathbf{r},\mathbf{p},\boldsymbol{\tau}) = r_0 + \sum_{i=1}^{n_\tau}\sum_{j=1}^{n_p}\frac{r_{ij}}{s-p_j}e^{-s\tau_i}. \tag{4.6}$$

The model parameters to be estimated by this new regression problem (i.e., to fit the data) are: the residues $\mathbf{r} = [r_0, r_{11}, \dots, r_{n_\tau n_p}]^T$, the poles $\mathbf{p} = [p_1, \dots, p_{n_p}]^T$ and the delays $\boldsymbol{\tau} = [\tau_1, \dots, \tau_{n_\tau}]^T$, for an overall number of $n_p n_\tau + n_p + n_\tau + 1$ unknowns.

The DRM is linear with respect to the residuals $\mathbf{r}$, but it is nonlinear with respect to the poles $\mathbf{p}$ and the delays $\boldsymbol{\tau}$. Unfortunately, it is still unfeasible to estimate all the parameters (i.e., the poles, the delays and the residuals) at the same time. A wise strategy is to rely on a two-step identification algorithm, where the delays are identified independently and afterwards the poles are estimated by considering such delays. Two methods that assume know delays and perform the estimation of poles are described below. The passivity of the resulting models is not guaranteed by the methods. However, it can be checked and enforced afterwards by using methods available in the literature [42].

4.2.1 Delayed Vector Fitting

A natural method to estimate a DRM is the so called delayed vector fitting (DVF) [42], [48], [49]. This is a modified version of the VF algorithm that uses delayed basis functions and through iterations identify a common set of poles for all the delays. Starting

from (4.4), the number of time-delays should be truncated to a small number n_τ, and a common rational approximation is considered for all coefficients $\tilde{H}_i(s)$ [48], i.e., this rational function is the same for all time-delay τ_i as elicited in (4.6). However, such rational function is rewritten in an alternative way expressed by:

$$\tilde{H}(s) = \sum_{i=1}^{n_\tau} \frac{\sum_{j=0}^{n_p} R_{ij}\phi_j(s)}{\sum_{j=0}^{n_p} r_j\phi_j(s)} e^{-s\tau_i}, \tag{4.7}$$

where each basis

$$\phi_j(s) = \begin{cases} 1, & j = 0; \\ \frac{1}{s-a_j}, & j = 1, \dots, n_p. \end{cases} \tag{4.8}$$

is associated to its own predefined "basis pole" a_j, which is different from the actual "model poles" p_j in (4.6).

Considering the above model, the approximation error at a single frequency point is expressed by

$$e_k = H(s_k) - \frac{\sum_{j=0}^{n_p} R_{ij}\phi_j(s_k)}{\sum_{j=0}^{n_p} r_j\phi_j(s_k)} e^{-s_k\tau_i}. \tag{4.9}$$

The computation of the SSE or other objective metric from the above expression leads to a complex nonlinear function of the model coefficients R_{ij} and r_j, mostly due to the presence of the terms r_j in the denominator of the expression. In order to overcome this issue, an iterative weighting is applied to (4.9). This iterative process, known as Sanathanan–Koerner (SK) iteration, the error is weighted by multiplying (4.9) by the frequency-dependent weight $\mathscr{W}_k^{(m)}$:

$$\mathscr{W}_k^{(m)} = \frac{\sum_{j=0}^{n_p} r_j^{(m)}\phi_j(s_k)}{\sum_{j=0}^{n_p} r_j^{(m-1)}\phi_j(s_k)}. \tag{4.10}$$

The weighted error $\varepsilon_k^{(m)} = \mathscr{W}_k^{(m)} e_k$ is thus given by

$$\varepsilon_k^{(m)} = \frac{H(s_k)\sum_{j=0}^{n_p} r_j^{(m)}\phi_j(s_k) - \sum_{i=1}^{n_\tau}\sum_{j=0}^{n_p} R_{ij}^{(m)}\phi_j(s_k)e^{-s_k\tau_i}}{\sum_{j=0}^{n_p} r_j^{(m-1)}\phi_j(s_k)}. \tag{4.11}$$

At the beginning of this iteration, with $m = 0$, the coefficients are set such that $r_j^{(0)} = 1$ for $j = 0$ and 0 otherwise. Then, with known parameters $r_j^{(m-1)}$, the weighted error becomes linear with respect to $R_{ij}^{(m)}$ and $r_j^{(m)}$, which can be computed via the addition of a non-triviality constraint and the solution of a standard least-squares system with the goal of minimizing such error [48]. This process is repeated for some iterations, in the

same way as the VF algorithm [23], by updating the coefficients $r_j^{(m)}$ and the basis poles a_j in (4.8) until they converge to a fixed value in the iteration $\bar{m}$. At that point, the final coefficients $R_{ij}^{(\bar{m})}$ and $r_j^{(\bar{m})}$ can be used to estimate the final model. The DVF method provide excellent results when an accurate estimate for the dominant time-delays is provided and the assumption of common poles is valid. Otherwise, an optimization over a small interval can be performed to tune the initial set of time-delay values [48] and improve the model accuracy.

4.2.2 Incremental Delayed Rational Fitting

The limitation of a common set of poles can be relaxed by the empirical algorithm that progressively builds the DRM in (4.4), by estimating the rational parts corresponding to the delay terms one at a time, and adding them to the overall model [83]. The DRM model $\check{H}$ is built in n_τ steps, being n_τ the number of identified delays, by considering a different set of poles for each delay term, i.e.,

$$H(s_k) \approx \tilde{H}(s_k) = r_0 + \sum_{i=1}^{n_\tau} \tilde{H}_i(s_k) e^{-s_k \tau_i}, \tag{4.12}$$

for $k = 1, \dots, K$, where, similar to (4.4), $\tilde{H}_i$ indicates the approximated rational terms associated to the delay τ_i.

The proposed procedure progressively identified the terms $\tilde{H}_i$ of the DRM in (4.12) by adding a new rational term at each iteration. Specifically, at a generic step l of the procedure, a standard rational fitting algorithm [23] is applied to get an approximated version $\tilde{H}_l(s_k)$ of the rational term associated to the l-th delay τ_l.

The target approximation of each rational term $\check{H}_l(s_k)$ is computed as the difference between the target transfer function H and the DRM approximation at the current iteration $\tilde{H}^{(l)}$, compensated for the time-delay associated to the current step τ_l, and writes:

$$\check{H}_l(s_k) = \left[H(s_k) - \tilde{H}^{(l-1)}(s_k) \right] e^{s_k \tau_l} \tag{4.13}$$

for $l = 1, \dots, n_\tau$. At each step l, a rational approximation $\tilde{H}_l(s_k)$ for $\check{H}_l(s_k)$ is estimated via the VF algorithm described in section 4.1.

Whilst, the overall DRM approximation $\tilde{H}^{(l)}$ at this generic iteration l writes

$$\tilde{H}^{(l)}(s_k) = \sum_{i=1}^{l} \tilde{H}_i(s_k) \exp(-s_k \tau_i). \tag{4.14}$$

At the fist step, the procedure is initialized as $\tilde{H}^{(0)}(s_k) = 0$. After n_τ steps, the algorithm ends and provides as a results the DRM $\tilde{H}(s_k) = \tilde{H}^{(n_\tau)}(s_k)$.

The rational fitting adopted at each iteration considers a different number of poles for each of rational term $\tilde{H}_i$ of the DRM. For the rational term, the number of poles is

selected by looking at the marginal improvement of the model error (i.e., we check if the model MSE decreases significantly when the number of poles is increased by one). The above procedure is summarized in Alg. 1.

The procedure does not lead to the optimal selection of the poles and residuals for the DRM, but it is simple and effective in providing a DRM with a small number of poles, which are different for each delay term. The DVF presented in the previous section is a more robust and automated method for a delayed rational fitting using the same poles for all delay terms [48].

Algorithm 1 Incremental rational fitting algorithm

1: Group the target frequency response sample pairs in the sets $\{(s_1, H(s_1)), \ldots, (s_K, H(s_K))\}$ and initialize the algorithm with the time-delay vector $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_{n_\tau}]$, and the approximate model $\tilde{H}^{(0)}(j\boldsymbol{s}) = 0$;
2: **for** $l = 1, \ldots, n_\tau$ **do**
3: Compute $\check{H}_l(s)$ by using (4.13);
4: Use VF as described in section 4.1 to get an approximate rational model $\tilde{H}_l(s)$ for $\check{H}_l(s)$ with a maximum number of poles from 1 to a small $n_{p,\max}$;
5: Select the appropriate number of poles as explained in 4.2.2;
6: Use (4.14) to update the partial DRM $\tilde{H}^{(l)}(s)$;
7: **end for**
8: $\tilde{H}(s) = \tilde{H}^{(n_\tau)}(s)$ is the final model containing all delays.

4.3 Time-delay estimation from Frequency-domain impulse response

The techniques presented above assumed that the time-delays were already known, and estimate the poles required for the rational part of the model. Naturally, once the delays τ_i and the poles p_j of the models are known, the residues can be easily estimated by solving a simple linear regression. However, how can we find an accurate estimation of the delays knowing only the samples $H(s_k)$? Two available schemes are briefly described below, based on the Hilbert transform and the Gabor transform, respectively.

4.3.1 Hilbert Transform Method

The physical systems which represent electric interconnects posses three important properties: they should stable, causal and passive [104]. Passivity was briefly commented in section 4.2, and will not be further discussed within the thesis, as it can be enforced after the models are obtained. Stability in a DRM is guaranteed by having all poles with a negative real part. However, such transfer function $H(s) = U(s) + jV(s)$

is causal only if its real part $U(j\omega)$ and imaginary part $V(j\omega)$ computed at the angular frequency $s = j\omega$ respect the Kramers–Kronig relations [35], [104], written as:

$$U(j\omega) = \frac{1}{\pi} \fint_{-\infty}^{\infty} \frac{V(j\omega')}{\omega' - \omega} d\omega', \tag{4.15}$$

and

$$V(j\omega) = -\frac{1}{\pi} \fint_{-\infty}^{\infty} \frac{U(j\omega')}{\omega' - \omega} d\omega', \tag{4.16}$$

where $\fint$ represents the computation of the Cauchy principal value of the improper integral, which has a singularity at $\omega = \omega'$.

However, any system can be decomposed into the product of a stable minimum-phase component $H_{min}(j\omega)$ and an all-pass function that accounts for the propagation delay τ [103]:

$$H(j\omega) = H_{min}(j\omega)e^{-j\omega\tau}. \tag{4.17}$$

The Hilbert transform method for delay identification finds the dominant propagation delay of the causal transfer function at $j\omega = j\omega_k$ by computing which delay τ can be removed from the transfer function which would allow it to remain causal. It is based on the Hilbert transform, which establishes an unique relation between the magnitude and phase of minimum-phase systems such that [36]:

$$\arg\left(H_{min}(j\omega)\right) = -\frac{1}{2\pi} \fint_{-\pi}^{\pi} \log|H_{min}(j\theta)| \cot\left(\frac{\omega - \theta}{2}\right) d\theta. \tag{4.18}$$

From (4.17), it is evident that the magnitude of $H_{min}(j\omega)$ can be easily determined as:

$$|H_{min}(j\omega)| = |H(j\omega)|. \tag{4.19}$$

The relation in (4.18) can therefore be applied to estimate the phase of $H_{min}(j\omega)$. After obtaining the complex value of $H_{min}(j\omega)$, the delay τ is estimated as [36], [37]

$$\tau = -\frac{\arg\left(e^{-j\omega\tau}\right)}{\omega} = -\frac{\arg\left(H(j\omega)/H_{min}(j\omega)\right)}{\omega}, \tag{4.20}$$

computed as the average over all available frequency points. The Hilbert transform method has good performance for systems with a single delay in the frequency response [37]. When multiple delays are present, the system delays can be identified by using the Gabor transform.

4.3.2 Gabor Transform Method

The short-time Fourier transform (STFT) is a technique to perform the time-frequency analysis of a signal, which can provide information on the time instant when certain frequencies occur, or the contrary, can indicate the frequencies present at a certain time

interval of such signal [105]. The tool used for this analysis is the spectrogram, a plot in a time-frequency axis where the energy decomposition of the signal is depicted [28].

The coefficients used to build the spectrogram are obtained by performing a windowed Fourier transform on the signal, with the window shifted in the time-axis by τ. When the window used in this transform is the Gabor wavelet, it is known as Gabor transform. The Gabor wavelet is the window which minimizes the uncertainty of this transform in both the time and frequency domain. It is defined as the Gaussian modulated by a complex exponential below:

$$f(t) = e^{(t-t_0)^2/a^2} e^{-j\,\omega_0(t-t_0)}. \tag{4.21}$$

Interestingly, its Fourier transform is also a Gabor wavelet:

$$F(\omega) = e^{(\omega-\omega_0)^2 a^2} e^{-j\,t_0(\omega-\omega_0)}. \tag{4.22}$$

Based on this and on the STFT, the normalized frequency domain Gabor transform is defined as follows [28]:

$$\mathcal{G}(\omega, \tau) = \int_{-\infty}^{+\infty} H(\xi) W^*_{\omega,\tau}(\xi) d\xi, \tag{4.23}$$

where,

$$W_{\omega,\tau}(\xi) = W(\xi - \omega) e^{-j\xi\tau}, \tag{4.24}$$

is the frequency-shifted window function modulated by the complex exponential $e^{-j\xi\tau}$, with the number of oscillations proportional to τ. The Gabor transform employed here uses a normalized Gaussian window [105], defined as:

$$W(\xi) = (a^2\pi)^{-1/4} e^{-\frac{\xi^2}{2a^2}}, \tag{4.25}$$

such that $\|W\|_2 = 1$.

The shape of the window $W(\xi)$ depends on the parameter a. Such parameter provides a trade-off between the time and frequency domain resolution of the transform. A small value for a produces a narrow window, for which the resolution of transform $\mathcal{G}(\omega, \tau)$ is high in frequency-domain (i.e., for the variable ω) and poor in time-domain (i.e., for the variable τ). Contrarily, with a large a, the transform resolution is poor in frequency-domain and high in time-domain.

The spectrogram is computed as $|\mathcal{G}(\omega, \tau)|^2$ and represents the time-frequency energy decomposition of $H(\xi)$ [28]. The propagation delays τ_i can be obtained from this energy distribution by averaging the energy decomposition over ω via:

$$G(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |\mathcal{G}(\omega, \tau)|^2 d\omega. \tag{4.26}$$

Once computed, the maxima of $G(\tau)$ provide a good estimate for the system delays τ_i [28], [48]. If $G(\tau)$ has multiple local maxima, this method is able to approximate multiple time-delays, thus overcoming the limitations of the Hilbert transform method. However, the time-frequency trade-off given by the window width still exists, and so the necessity of tuning such width.

4.4 Kernel-Based Delay Rational Model

Delay rational models are a natural choice to represent distributed systems [42], [48]. However, the previous sections illustrated some of the issues and limitations in their estimation. The kernel-based techniques presented in chapter 3 are a powerful tool that can be employed to sidestep some of the issues. But in order to introduce them smoothly, let us start by thinking on the simplest way that a DRM can be estimated.

4.4.1 Simple Grid Search

A grid search approach can be envisaged as the simplest way to estimate the delays and the poles of the DRM in (4.6). Such straightforward approach represents the key element of the generic algorithm for the delay identification presented in section 4.5.

Without loss of generality, the following synthetic transfer function with a single real pole at -3 rad/s and single delay at 3 s, is considered:

$$H(s) = \frac{1}{s+3} e^{-3s}, \tag{4.27}$$

in the bandwidth $s = j\omega$ with $\omega \in [0,5]$ rad/s. The above function will be used hereafter to explain step-by-step the underlying idea behind grid search and to highlight the pros and cons of such approach.

Contrary to the optimization approach, the grid search a priori assumes the value of poles and delays by exploration of a mutidimensional grid. In the considered example, a reasonable guess for poles and delays in a 2D grid is defined. The poles are restricted to real values to avoid using a 3D grid, which hinders the desired visualization, but the generalization to complex poles follows easily from the presented ideas. We set the number of poles n_p and of delay terms n_τ along with their range of variation $p_j \in [p_m, p_M]$ and $\tau_i \in [\tau_m, \tau_M]$ in terms of their minimum and maximum values, such that the poles and delays in (4.6) take the following values:

$$p_j = p_m + (j-1)\Delta_p, \tag{4.28}$$

and

$$\tau_i = \tau_m + (i-1)\Delta_\tau, \tag{4.29}$$

where

$$\Delta_p = (p_M - p_m)/n_p, \tag{4.30}$$

and

$$\Delta_\tau = (\tau_M - \tau_m)/n_\tau. \tag{4.31}$$

By knowing the values of the poles and delays, the non-linear regression problem in (4.4), can be recasted in terms of the following linear combination of basis functions:

$$H(j\omega) \approx \sum_{i=1,j=1}^{n_p,n_\tau} r_{ij}\hat{\varphi}_{ij}(\omega; p_j, \tau_i) + r_0 \tag{4.32}$$

where

$$\hat{\varphi}_{ij}(\omega; p_j, \tau_i) = \frac{1}{j\omega - p_j} e^{-j\omega\tau_i} \tag{4.33}$$

and r_{ij} are the unknowns of the linear regression problem representing the residues of the DRM, which can be readily estimated after the grid is established by solving a linear least squares regression.

The above scheme is now applied to the synthetic transfer function of (4.27) by using a 3×3 (i.e., $n_p = 3$ and $n_\tau = 3$) discretization grid for the pole and delay. The three scenarios depicted in Fig. 4.1 have been considered in order to stress the performances and drawbacks of the proposed approach. Specifically, the scenarios can be divided as such:

- Case a: the discretization is such that the pole and delay values of the original system in (4.27) happen to exactly coincide with a grid node (Fig. 4.1, left panel);
- Case b: the discretization is such that the original delay value happen to lay on a discretization trace, but the pole is off any grid traces (Fig. 4.1, right panel);
- Case c: the discretization is such that the original pole value happen to lay on a discretization trace, but the delay is off any grid traces (Fig. 4.1, bottom panel).

The above three discretization scenarios have been used to construct three versions of the DRM model of (4.32), in which the basis functions $\hat{\varphi}_{ij}$ are built using the blue, red and green dots of the 2D grid in Fig. 4.1, respectively. The corresponding residues are finally calculated via a LS minimization.

Fig. 4.2 shows how well the three models fit the original transfer function. It is ought to remark that the models corresponding to the discretized space for case a and b turn out to be very accurate. On the other hand, the model corresponding to case c, in which the value of the delay falls in between the grid points, is far from being acceptable. In

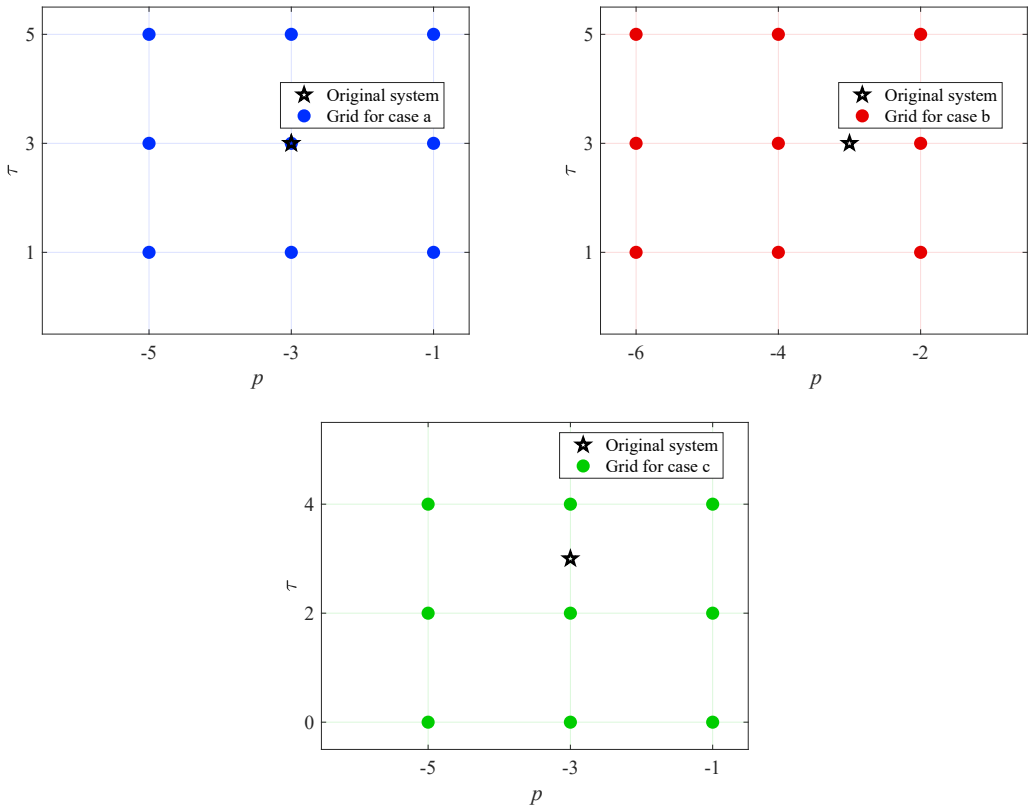


Figure 4.1: Discretization of a $p - \tau$ plane of the three considered cases; the position of the original system pole and delay in (4.27) is indicated for reference.

fact, even for this very simple transfer function in (4.27) that was considered in this illustrative example, the results show that the DRM is extremely sensitive to the delay terms, and an error in the delay estimation can heavily affect the model accuracy.

One way to overcome this inaccuracy issue is to increase the number of points in the 2D grid on the τ-axis (i.e., by increasing n_τ and thus decreasing the discretization term Δ_τ), which can be done with the help of the LSSVM and the kernel trick, as illustrated in the next section.

4.4.2 Estimating a Delayed Rational Kernel-Based Model

The class of kernel-based ML techniques presented in chapter 3 can be applied to fit the data of a transfer function via a DRM [89]. For that regression, the input is the complex frequecy $s \in \mathbb{C}$ and the output is the transfer function $\tilde{H}(s) \in \mathbb{C}$.

The discussion starts by recasting the DRM in (4.32) in terms of the primal-space

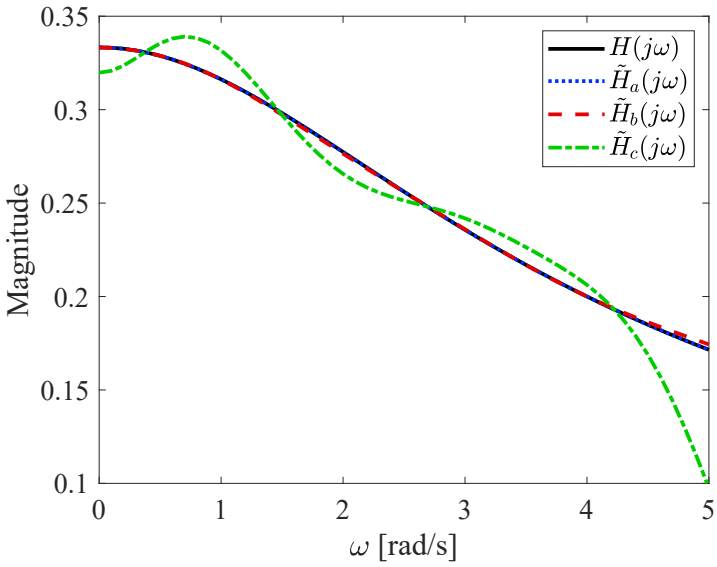


Figure 4.2: Original frequency response of the system (black solid line) and approximations using the three considered discretization cases (blue, red and green lines, according to the dots color in Fig. 4.1).

LSSVM regression in (3.16) which writes:

$$\tilde{H}(s) = \langle \mathbf{w}, \boldsymbol{\varphi}^*(s; \mathbf{p}, \boldsymbol{\tau}) \rangle + b \tag{4.34}$$

$$= \sum_{i=1,j=1}^{n_p,n_\tau} \mathrm{w}_{ij} \varphi_{ij}(s; p_j, \tau_i) + b, \tag{4.35}$$

where $\mathbf{w} = [\mathrm{w}_{11}, \dots, \mathrm{w}_{n_\tau n_p}]^T \in \mathbb{C}^{(n_\tau n_p)}$ is a vector collecting the regression unknowns, the generic parameter $\mathbf{x}$ in (3.6) has been replaced by s (i.e., $d = 1$) and $\boldsymbol{\varphi}(s; \mathbf{p}, \boldsymbol{\tau}) \in \mathbb{C}^{(n_\tau n_p)}$ is a vector collecting the basis functions

$$\boldsymbol{\varphi}(s; \mathbf{p}, \boldsymbol{\tau}) = \left[\varphi_{11}(s; p_1, \tau_1), \dots, \varphi_{n_\tau n_p}(s; p_{n_p}, \tau_{n_\tau}) \right]^T, \tag{4.36}$$

which depend on a vector of poles $\mathbf{p} = [p_1, \dots, p_{n_p}]^T$ and delays $\boldsymbol{\tau} = [\tau_1, \dots, \tau_{n_\tau}]^T$. This formulation can be seen as a discretization in the p-τ space from which each node $\{\tau_i, p_j\}$ leads to a basis of the DRM, and in which all poles are paired with the same combination of delays, in the same manner which was described in section 4.4.1, or in other words, each delay term takes a common set of poles.

Comparing (4.35) and (4.32), the basis functions $\varphi_{ij}(s; p_j, \tau_i)$ can be defined by the

following expression:

$$\varphi_{ij}(s; p_j, \tau_i) = \frac{c_{ij}}{s - p_j} e^{-s\tau_i} \tag{4.37}$$

$$= c_{\tau_i} e^{-s\tau_i} \cdot \frac{c_{p_j}}{s - p_j} \tag{4.38}$$

$$= \varphi_i^{\tau}(s; \tau_i) \cdot \varphi_j^{p}(s; p_j), \tag{4.39}$$

where the effects of poles and time delays where separated into the product of two individual basis functions which depend on only one of those parameters, and $c_{ij} = c_{\tau_i} c_{p_j}$ is a constant introduced to increase the model flexibility without changing the model structure. The overall number of features (or bases) is $N = n_p n_\tau$, which is equivalent to the number of nodes in the aforementioned discretization.

Form the latter development, it is evident that the regression unknowns w_{ij} are proportional to the corresponding residue r_{ij} by the scalar constant $c_{ij} \in \mathbb{R}$:

$$r_{ij} = c_{ij} \mathrm{w}_{ij}. \tag{4.40}$$

It is also clear that the bias term b is equal to the constant term r_0 of the DRM. The use of a finer discretization in the grid of poles and delay will increase the dimensions of the feature space provided by the transformation function $\boldsymbol{\varphi}$ (i.e., the number of basis φ_{ij} included in the model will be larger), complicating the estimation of this regression.

However, the model in (4.35) is depicted via the primal space LSSVM formulation, which is equivalent to the original one in (4.32). The LSSVM framework has an equivalent dual space formulation based on a kernel function and its RKHS, as described in chapter 3. Therefore, aside from the above expansion, a parallel dual space formulation for the LSSVM problem, defined by (3.29), takes the form

$$\tilde{H}(s) = \sum_{k=1}^{K} \alpha_k \mathrm{k}(s, s_k; \mathbf{p}, \boldsymbol{\tau}) + b, \tag{4.41}$$

where the kernel $\mathrm{k}(\cdot, \cdot) : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ is defined according to the basis functions collected in $\boldsymbol{\varphi}$ as:

$$\mathrm{k}(s_k, s_l; \mathbf{p}, \boldsymbol{\tau}) = \langle \boldsymbol{\varphi}(s_k; \mathbf{p}, \boldsymbol{\tau}), \boldsymbol{\varphi}(s_l; \mathbf{p}, \boldsymbol{\tau}) \rangle \tag{4.42}$$

$$= \sum_{i=1}^{n_\tau} \sum_{j=1}^{n_p} \frac{c_{ij} e^{-s_k^* \tau_i}}{(s_k^* - p_j^*)} \frac{c_{ij} e^{-s_l \tau_i}}{(s_k - p_j)}. \tag{4.43}$$

In the above expression, the effects of $\mathbf{p}$ and $\boldsymbol{\tau}$ can be further split by applying the basis separation shown in (4.39) to the above expression:

$$\mathrm{k}(s_k, s_l; \mathbf{p}, \boldsymbol{\tau}) = \underbrace{\sum_{i=1}^{n_\tau} c_{\tau_i}^2 e^{-(s_l + s_k^*)\tau_i}}_{\mathrm{k}_\tau(s_k, s_l; \boldsymbol{\tau})} \underbrace{\sum_{j=1}^{n_p} \frac{c_{p_j}^2}{(s_k^* - p_j^*)(s_l - p_j)}}_{\mathrm{k}_\mathrm{p}(s_k, s_l; \mathbf{p})} \tag{4.44}$$

$$= \langle \boldsymbol{\varphi}^\tau(s_k; \boldsymbol{\tau}), \boldsymbol{\varphi}^\tau(s_l; \boldsymbol{\tau}) \rangle \langle \boldsymbol{\varphi}^p(s_k; \mathbf{p}), \boldsymbol{\varphi}^p(s_l; \mathbf{p}) \rangle . \tag{4.45}$$

where p_j and τ_i span all the poles and delays values considered in their discretization. The resulting kernel can be interpreted as the product of two kernels according to the property in (3.41e), one generated by the chosen delays and the other by the chosen poles. Such two terms, $\mathrm{k}_\tau(s_k, s_l; \boldsymbol{\tau})$ and $\mathrm{k}_p(s_k, s_l; \mathbf{p})$, are then named the delay and the rational kernels, respectively. Each of them, interestingly, can be seen as the sum of several kernel function with a single basis φ_{ij} via the property in (3.41d).

The model in (4.41) is a non-parametric model where the number of estimated coefficients is always equal to $K+1$ (the number of training samples plus one), independent of the number and shape of the basis functions that compose the RKHS. On the other hand, the model in (4.32) requires the estimation of $n_p n_\tau + 1$ coefficients, a number that changes according to the number of basis functions accounted by the model.

Considering the LSSVM framework, the dual formulation model in (4.41) can be suitably estimated by solving the system of linear equations in (3.24). The above kernel depends on the definition of the n_p poles $p_j = p_j' + jp_j''$ and the n_τ delays τ_i. Once defined, such kernel represents a space formed by bases in the following form:

$$\varphi_{ij}(s; p_j, \tau_i) = \frac{c_{ij}}{s - p_j} e^{-s\tau_i}, \tag{4.46}$$

which is equivalent to the basis of the DRM in (4.32) multiplied by the constant c_{ij}.

However, let's recall that (4.41) is a non-parametric model, and no matter how finely the $p - \tau$ grid is discretized, it can always be determined by means of the solution of the linear system in (3.24). Indeed, this discretization is embedded within the kernel function. The key advantage of working with this dual space model is that the dual space formulation does not require an explicit definition of the basis functions $\boldsymbol{\varphi}$ and its formulation can even be applied to the extreme case in which the dimension of $\boldsymbol{\varphi}$ grows to infinity.

According to this fact, let us consider the case in which the $\tau-$axis is infinitely discretized and any τ value between τ_m and τ_M is included in the kernel. This discretization is achieved by defining the constant c_{τ_i} in (4.38) as:

$$c_{\tau_i}^2 = c_\tau^2 = \Delta_\tau = \frac{(\tau_M - \tau_m)}{n_\tau}, \tag{4.47}$$

and the value for τ_i such that:

$$\tau_i = \tau_m + (i-1)\Delta_\tau = \tau_m + (i-1)\frac{(\tau_M - \tau_m)}{n_\tau}. \tag{4.48}$$

When $n_\tau \to \infty$, the delayed kernel k_τ in (4.42) can only be computed through the limit written below:

$$\mathrm{k}_\tau(s_k, s_l) = \lim_{n_\tau \to \infty} \sum_{i=1}^{n_\tau} c_{\tau_i}^2 e^{-(s_k + s_l^*)\tau_i} \tag{4.49}$$

$$= \sum_{i=1}^{n_\tau} \frac{(\tau_M - \tau_m)}{n_\tau} e^{-(s_k + s_l^*)(\tau_m + (i-1)(\tau_M - \tau_m)/n_\tau)} \tag{4.50}$$

which is equivalent to a Riemann sum [106], or in other words, a definite integral.

Hence, the computation of the inner product $\langle \boldsymbol{\varphi}^\tau(s_k; \boldsymbol{\tau}), \boldsymbol{\varphi}^\tau(s_l; \boldsymbol{\tau}) \rangle$ in (4.45) should be performed over the continuous parameter τ. This is analog to making a kernel via (3.40) where $\vartheta = \tau$ and the region Θ is defined as the interval $[\tau_m, \tau_M]$. Taking into account these definitions, the delayed kernel is computed according to:

$$\mathrm{k}_\tau(s_k, s_l; \tau_m, \tau_M) = \int_{\tau_m}^{\tau_M} e^{-(s_l^* + s_k)\tau} d\tau. \tag{4.51}$$

The above delay kernel represents a space with an infinite dimension due to the continuous $\tau-$axis. The discretization along the $p-$axis is given by the other part of the kernel, $\mathrm{k}_p(s_k, s_l)$, which might remain discrete.

The integral in (4.51) can be evaluated to a closed-form expression, which writes:

$$\mathrm{k}_\tau(s_k, s_l; \tau_m, \tau_M) = \begin{cases} \frac{-\left(e^{-\tau_M(s_l^* + s_k)} - e^{-\tau_m(s_l^* + s_k)}\right)}{(s_l^* + s_k)}, & s_k + s_l^* \neq 0; \\ \tau_M - \tau_m, & s_k + s_l^* = 0. \end{cases} \tag{4.52}$$

On the other hand, the rational kernel in (4.42) remains a finite sum over a limited set of poles $\mathbf{p}$. This set of poles will provide the general trend that must be followed by the rational part of the model. As seen in section 4.4.1, the exact delay is the most important part of the model, and thus the rational kernel does not need to contain the exact poles of the system. Meanwhile, $k_\tau(s, s_k)$ accounts for all possible delay terms between τ_m and τ_M. As a matter of fact, the use of such kernel provides a feature space with an infinite number of dimensions, i.e., an infinite number of basis. This feature space guarantees that the exact delays from the modeled system are replicated in the LSSVM model given by:

$$\tilde{H}(s) = \sum_{k=1}^{K} \alpha_k \mathrm{k}(s, s_k; \mathbf{p}, \tau_m, \tau_M) + b, \tag{4.53}$$

An approach including an infinite number of poles as done for the delay kernel could also be possible, and will be considered in future works. However, the resulting integration leads to complicated equations, which makes the subsequent evaluation of the kernel slow, limiting the benefits that it could bring.

The constants c_{p_j} in (4.38) are set to $c_{p_j} = |p'_j|^{1/2}$. In this way, all the terms $\varphi^p_j(s; p_j)$ at $s = j\omega$ have the same ℓ_2-norm for any value of p_j for which $p'_j \neq 0$. Considering this definition, the ℓ_2-norm of each of the $\varphi^p_j(\omega; p_j)$ basis amounts to:

$$\left\langle \varphi^p_j(\omega; p_j), \varphi^p_j(\omega; p_j) \right\rangle = \int_{-\infty}^{\infty} \frac{|p'_j|}{(j\omega - p_j)(-j\omega - p^*_j)} d\omega = \pi. \tag{4.54}$$

This definition of c_{p_j} amplifies the LSSVM weights w_{ij} associated with the dominant poles of the system (i.e., the poles with a smaller real part). A smaller value for c_{ij} produced by the proposed c_{p_j} means that the weight w_{ij} should increase, aaccording to (4.40), in order to achieve the same residue, and thus the same model.

Separating the rational part of (4.42), and with the considerations stated above, the rational kernel $\mathrm{k}_p(s_k, s_l)$ writes:

$$\mathrm{k}_p(s_k, s_l) = \sum_{j=1}^{n_p} \frac{|p'_j|}{\left(s_k - p_j\right)\left(s^*_l - p^*_j\right)}. \tag{4.55}$$

According to (4.45), the delayed rational kernel results from the multiplication of the delayed kernel in (4.52) by the rational kernel in (4.55). It contains the delayed rational basis of (4.37) with the set of poles $\mathscr{P} = \{p_1, \ldots, p_{n_p}\}$, each of them containing the same infinite number of delays in the range $[\tau_m, \tau_M]$.

4.4.3 Delay Extraction via LSSVM Regression

The previous section described the application of the dual space formulation of the LSSVM on the estimation of a kernel-based DRM. While the dual space LSSVM model in (4.53) with the aforementioned kernel is non-parametric, it can still be linked to an infinite dimensional parametric model expressed by (4.34):

$$\begin{aligned} \tilde{H}(s) &= \left\langle \mathbf{w}, \boldsymbol{\varphi}^*(s; \mathbf{p}, \boldsymbol{\tau}) \right\rangle + b \\ &= \sum_{j=1}^{n_p} \sum_{i=1}^{n_\tau} \mathrm{w}_{ij} e^{-s\tau} \frac{c_\tau c_{p_j}}{s - p_j} + b. \end{aligned} \tag{4.56}$$

The value of w_{ij} is defined via the LSSVM framework, and can be computed by

(3.20):

$$\mathrm{w}_{ij} = \sum_{k=1}^{K} \alpha_k \varphi_{ij}^*(s_k), \tag{4.57}$$

$$= \sum_{k=1}^{K} \alpha_k \frac{c_\tau c_{p_j}}{s_k^* - p_j^*} e^{-s_k^* \tau_i}. \tag{4.58}$$

Since $c_\tau \to 0$ and $n_\tau \to \infty$ were used in the kernel definition, the expression in (4.56) can be further expanded to:

$$\tilde{H}(s) = \sum_{j=1}^{n_p} \int_{\tau_m}^{\tau_M} \mathrm{w}_j'(\tau) e^{-s\tau} \frac{c_{p_j}}{s - p_j} d\tau + b, \tag{4.59}$$

where the weights $\mathrm{w}_j'(\tau)$ refer to the continuous parameter τ and exist for any pole p_j. They take the following form:

$$\mathrm{w}_j'(\tau) = \lim_{c_\tau \to 0} \frac{\mathrm{w}_{ij}}{c_\tau}, \tag{4.60}$$

$$= \lim_{c_\tau \to 0} \frac{1}{c_\tau} \sum_{k=1}^{K} \alpha_k \frac{c_\tau c_{p_j}}{s_k^* - p_j^*} e^{-s_k^* \tau}, \tag{4.61}$$

$$= \sum_{k=1}^{K} \alpha_k \frac{c_{p_j}}{s_k^* - p_j^*} e^{-s_k^* \tau}. \tag{4.62}$$

These weights $\mathrm{w}_j'(\tau)$ are proportional to the residue in the DRM, thus providing information on the most relevant terms for the model construction, i.e., on the values of p_j and specially, τ, that have a larger influence in the model; in fact, a large residue is associated to the delay that is most significant for the part of the model corresponding to the pole p_j. Furthermore, a combined weight $W(\tau)$ that which sums the contributions of all the poles in the final model, leading to a τ-dependency only, can be defined as:

$$W(\tau) = \left(\sum_{j=1}^{n_p} |\mathrm{w}_j'(\tau)|^2 \right)^{1/2}$$

$$= \left(\sum_{j=1}^{n_p} \left(\sum_{k=1}^{K} \frac{\alpha_k c_p}{s_k^* - p_j^*} e^{-s_k^* \tau} \right) \left(\sum_{k=1}^{K} \frac{\alpha_k c_p}{s_k^* - p_j^*} e^{-s_k^* \tau} \right)^* \right)^{1/2}. \tag{4.63}$$

The plot and analysis of the weight $W(\tau)$ provides information about the system delays, with peaks indicating higher "energy" content and corresponding to the dominant delays of the original system. The representativeness of these peaks to identify the

system delays is assured by the fact that one of the conditions of the LSSVM problem is the minimization of $\langle \mathbf{w}, \mathbf{w} \rangle = \sum |\mathrm{w}|^2$ (see (3.17)). Hence, it is unlikely the occurrence of unnecessary high values for $\mathrm{w}'_j(\tau)$.

4.4.4 Illustrative Example

As an illustrative example, we recall the synthetic transfer function with a single real pole at -3 rad/s and a single delay at 3 s from section 4.4.1.

The LSSVM model from (4.53) with $\tau_m = 0$, $\tau_M = 5$ s and the poles $\mathscr{P} = \{-2, -4, -6\}$ rad/s can be easily estimated by computing the kernel and the coefficients $\boldsymbol{\alpha}$. It is important to remark that the above set of poles does not match the exact pole in the transfer function, but are within the same region of space. The plot of the magnitude of $\mathrm{w}'_j(\tau)$ as a function of τ gives us an interesting insight. Figure 4.3 shows these curves for each of the poles included in the kernel. The curves clearly have their peak near the true delay of the original system, $\tau = 3$ s. However, if the weights are observed for one individual pole, the peak is not exactly at the expected value: in fact, it is at $\tau = 3.09$ s for $p_j = -6$ rad/s, $\tau = 3.05$ s for $p_j = -4$ rad/s and $\tau = 2.95$ s for $p_j = -2$ rad/s. Nonetheless, by looking at the combined weight $W(\tau)$, the peak appears at $\tau = 3.00$ s, showing that the use of a higher number of poles compensate the fact that the exact pole of the system is unknown. The delay is shown to be the most important element for obtaining an accurate model, and the use of infinite possible delays by means of the kernel function guarantees that its correct value is considered. In the next section, this procedure for delay identification is described for a more generic case, when the original system is unknown.

4.5 Practical Procedure for delayed rational model Estimation and delay identification

This section generalizes the procedure illustrated in the previous section to the case of a system with multiple delays and reflections like, for example, an electric interconnect. The target model to approximate this system is the DRM in (4.12), with *complex* and *unknown poles*.

4.5.1 Time-Delay Intervals in the Delay Kernel

The delay extraction approach outlined in Sec. 4.4 and better detailed in this section requires a delay-rational kernel function containing all delays within a certain interval. However, the span of this interval was not discussed yet. For causal systems, the time-delay has to be positive, and therefore, $\tau_m \geq 0$. The determination of the maximum

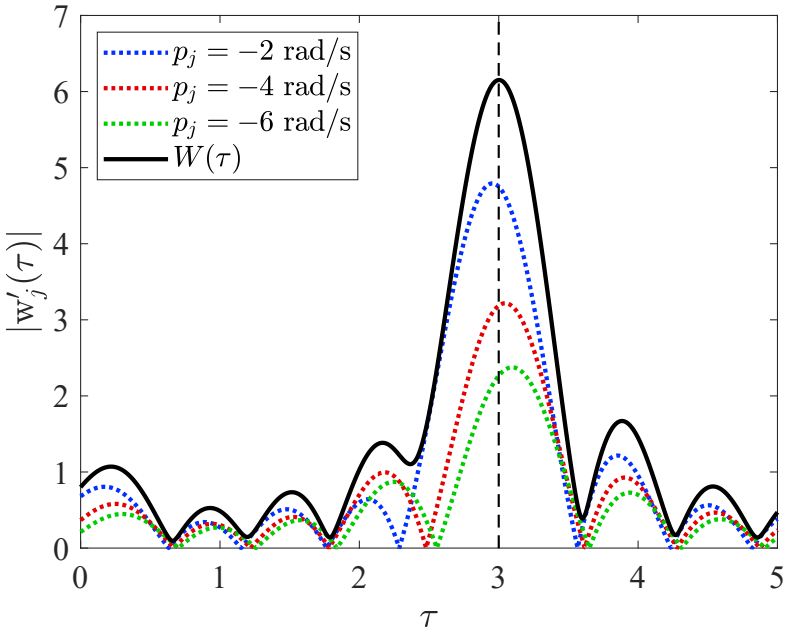


Figure 4.3: Magnitude of the LSSVM weights as a function of τ for each pole included in the kernel.

delay τ_M can rely on the frequency discretization of input data:

$$\begin{bmatrix} s_1 \\ \vdots \\ s_K \end{bmatrix} = \begin{bmatrix} j\omega_1 \\ \vdots \\ j\omega_K \end{bmatrix} = j\,2\pi \begin{bmatrix} f_1 \\ \vdots \\ f_K \end{bmatrix}. \tag{4.64}$$

In fact, assuming that the original frequency response of the system has equally-spaced frequency points, with Δ_f being the separation between two consecutive samples (e.g., the difference between f_1 and f_2), the following empirical relationship holds:

$$\tau_M - \tau_m < \frac{1}{\Delta_f}. \tag{4.65}$$

With $\tau_m = 0$, the maximum time-delay then becomes

$$\tau_M < \frac{1}{\Delta_f}, \tag{4.66}$$

which is the value we use for our examples presented in section 4.6.

4.5.2 The Rational Kernel

The example in section 4.4.4 showed that the delays can be identified without knowing the exact poles that compose the rational part of the system. A sufficiently large number

of poles was able to replace the effect of the exact pole in the estimation of the LSSVM model. Hence, the set $\mathscr{P} = \{p_1, \ldots, p_{n_p}\}$ is empirically defined as an appropriate choice for the kernel poles. The poles $\{p_1, \ldots, p_{n_p}\}$ are randomly chosen with independent real and imaginary parts, both from a Gaussian distribution with mean zero and standard deviation equal to ω_K, i.e., the maximum frequency among the available samples. The real part of the poles is forced to be negative by flipping its sign if p' is positive, in order to ensure that the obtained system is stable. Although this is not the optimal set of poles for the rational kernel, this choice is adopted because it provides a good delay estimation using a relatively small number of poles. Since the rational part of the system (with its corresponding delay removed) should be very simple, usually no more than 10 poles, a n_p on the order of 20 to 30 poles is sufficient to identify the delay values.

4.5.3 Delay Identification Algorithm

After considering the details of the previous subsections, we can devise a procedure to identify the relevant delays of a generic distributed system based on the weights estimated by a LSSVM model. The procedure is simply an extension of what was described at the end of Sec. 4.4 and illustrated by the black line of Fig. 4.3. The complete procedure for time-delay identification can be summarized by Algorithm 2. Once identified by the peak positions, the delays can be used to build a DRM of little complexity.

Algorithm 2 Delay identification algorithm

1: Group the frequency response sample pairs in the sets $\big\{(s_1, H(s_1)), \ldots, (s_K, H(s_K))\big\}$, with $H(s) = H_r(s) + jH_i(s)$ and $s = j\omega$;
2: Draw a random set of complex poles $\mathscr{P} \in \mathbb{C}^{n_p}$ in a reasonable region of the complex plane as discussed in section 4.5.2;
3: Define the minimum and maximum considered delays τ_m and τ_M as stated in section 4.5.1;
4: Compute the kernel for all frequency pairs $\{s_i, s_j\}_{i,j=1}^{K}$ using (4.42), and assembly the linear system in (3.24);
5: Tune the hyperparameter γ such that the model error is acceptable and solve (3.24);
6: Compute $W(\tau)$ according to (4.63);
7: Find the peaks in $W(\tau)$, which are the set of relevant delays $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_{n_\tau}\}$ for the modeled system.

Admittedly, the peaks still need to be found visually, and some discernment is required to distinguish the delay peaks from the side lobes that appear in the plot, as observed in Fig. 4.3. Nonetheless, the main delay peak is more pronounced than its corresponding side lobes, and hence the delay peaks are identified as being higher than its near surroundings. Those few identified delays can be used to estimate compact delay rational models with a small number of poles via one of the techniques described in section 4.2.

4.6 Application examples

In order to exemplify the use of the advocated procedure, four examples are presented in this section. Throughout the examples, comparisons with the state-of-the-art VF technique for the estimation of rational models [23], Hilbert transform [36] approach and Gabor transform [28] method for delay identification are presented.

4.6.1 Example 1: Synthetic transfer functions

The proposed model is first investigated on a synthetic example with known poles and delays [83]. Initially, the following transfer function sampled at 1001 evenly spaced points from $\omega = 0$ to $\omega = 2000$ rad/s is considered:

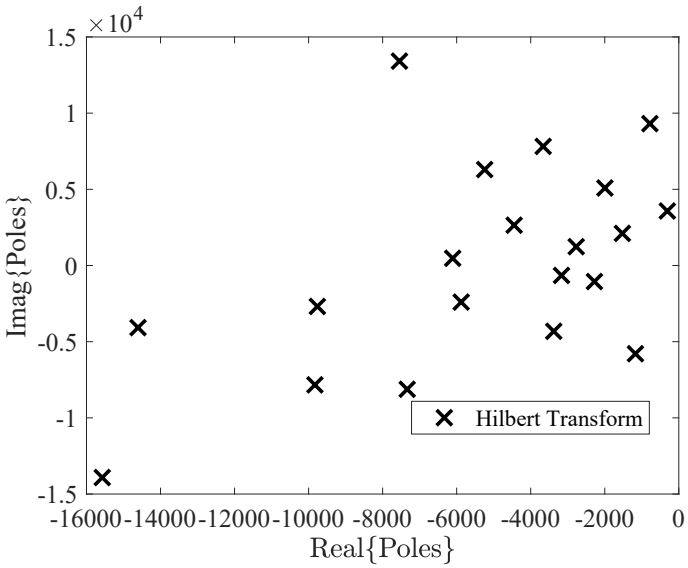
$$H_1(s) = \frac{20\pi}{s + 20\pi} e^{-0.15s}. \tag{4.67}$$

For the delay identification, we adopted a kernel with the 20 random poles shown in Fig. 4.4-(a), and a maximum delay value $\tau_M = 1$ s determined according to subsection 4.5.1. Then, Algorithm 2 is used to compute the weight $W(\tau)$. Such weight is normalized by its maximum value and displayed in Fig. 4.4-(b). The peak identifies exactly the delay of our synthetic transfer function, i.e. $\tau = 0.15$ s. The figure also compares the results of the present method with the time-delays obtained by means of two other methods: the Hilbert transform (gray line) and the Gabor transform (dashed lines). The former method identifies by design only one delay value, at $\tau = 0.1494$ s, which is slightly to the left of the actual delay value. On the other hand, the Gabor transform, after the tuning of the window width provides a pattern which is similar to the one provided by the proposed method. However, the peak in the Gabor transform plot is shifted slightly to the right of the LSSVM peak, and it can be pushed further if smaller values for the window width a are chosen.

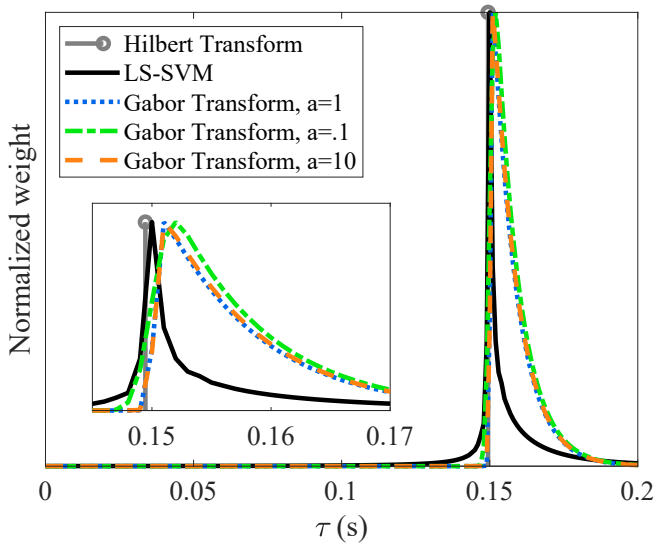
As a further comparison among the three methods, and to stress their performances for a system with multiple delays, let us consider a second synthetic transfer function:

$$H_2(s) = \left(\frac{20\sqrt{10}\pi}{s + 20\pi + 60\pi j} + \frac{20\sqrt{10}\pi}{s + 20\pi - 60\pi j} \right) e^{-0.15s} + \frac{4\pi}{s + 20\pi} e^{-0.3s} \tag{4.68}$$

sampled at the same frequency points as $H_1(s)$. There are two main differences from the previous example: $H_2(s)$ is composed by terms with two different delays $\tau = \{0.15, 0.3\}$ s, and one of this terms is an underdamped system. The normalized $W(\tau)$ computed via the procedure described in section 4.5 is displayed in Fig. 4.5, together with the Hilbert and Gabor transform curves. The main LSSVM weight peak identifies exactly the delays of our synthetic transfer function, i.e. $\tau = 0.15$ s, and a smaller peak is present at the second transfer function delay at $\tau = 0.3$ s. The Hilbert transform also identifies exactly the first and most important delay peak, but misses the second one.



(a)



(b)

Figure 4.4: First synthetic example: Panel (a) shows the 20 random poles used within the LSSVM kernel. Panel (b) plots $W(\tau)$ for delay identification; comparison with Hilbert and Gabor transforms is also displayed.

The Gabor transform peaks are again shifted slightly to the right, but this time additional side peaks appear when the frequency domain window is wide. They disappear by using a narrower window, as shown in the green curve, but these leads to a worse time-domain resolution of the transform, which makes both the first and second peaks appear in more inaccurate values of τ.

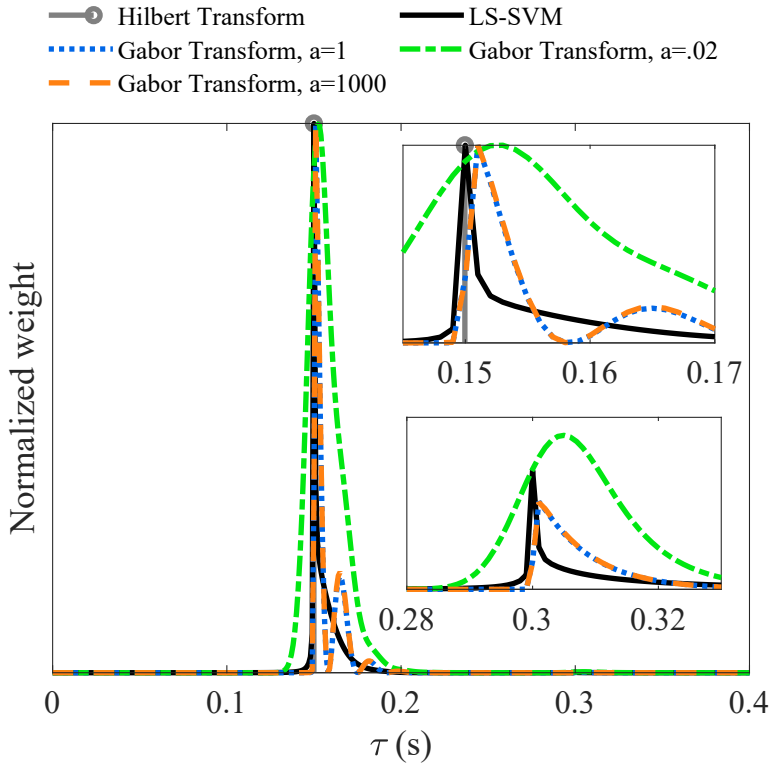


Figure 4.5: Plots of the normalized weight used for delay identification; comparison with Hilbert and Gabor transforms is also displayed, for the second synthetic example.

The computational cost of the LSSVM method might be worrisome, as it depends on the inversion of matrices, which might be computationally when their size becomes large. However, Table 4.1 shows that the impact on the performance was not so large in these examples. Using those $K = 1001$ frequency response samples, the LSSVM required a computational time of 17.57 and 16.45 s, compared to 12.96 and 9.88 s for the Gabor transform. For smaller number of samples, the difference is reduced, up to a point where the LSSVM may be faster than the Gabor transform, e.g., with 334 samples. The Hilbert transform always required a larger computational time than both other methods. Nonetheless, the computational time for LSSVM is an active research area, and such time can be further reduced in the future via alternative methods for the estimation of the dual-space model parameters $\boldsymbol{\alpha}$ such as the gradient descent algorithm [107].

As can be inferred from Table 4.1, the proposed method presents a trade-off between the maximum frequency of the system (or, in other words, the considered bandwidth), the maximum identifiable delay (which is related to the length of the cable: longer cables will require larger maximum delays) and its computational performance (which

Table 4.1: Comparison of the computational time between the three applied delay identification methods for different number of frequency response samples, for the two examples of synthetic transfer functions.

Delay identification method:		LSSVM	Gabor transform	Hilbert transform
$H_1(j\omega)$: Computational time [s]	1001 samples	17.57	12.96	276.47
	501 samples	7.00	6.47	113.06
	334 samples	4.15	5.24	71.62
$H_2(j\omega)$: Computational time [s]	1001 samples	16.45	9.88	220.83
	501 samples	7.35	7.52	110.94
	334 samples	4.50	5.65	82.65

declines when the number of frequency samples is increased). For a fixed computational time, if the maximum frequency is increased, Δ_f increases, and therefore the maximum identifiable delay τ_M is smaller. On the other hand, channels that operate at higher frequencies would be naturally shorter, thus the delays that need to be identified in such channels are smaller, compensating the shrink of τ_M.

4.6.2 Example 2: Distributed circuit

As a second example, the high-speed link circuit in Fig. 4.6 is considered [90]. The link consists of three transmission lines based on microstrips, together with lumped elements that represent the parasitic effects of the link, in order to approximate the structure of a realistic interconnect. The desired model targets the following transfer function:

$$H(s) = V_{out}(s)/E(s) \tag{4.69}$$

for $s = j\omega = j2\pi f$. This structure has been implemented and simulated in HSPICE in an AC simulation, with results consisting on $K = 1001$ frequency points with a frequency spacing $\Delta_f = 20\,\text{MHz}$, resulting in a bandwidth from 0 to 20 GHz. A subset containing 101 samples, randomly selected among the available data, has been used as a test dataset, whilst the remaining 900 samples are used as training set to construct the LSSVM model.

A DRM based on the modeling approach presented in section 4.5 is constructed to approximate $H(s)$. The set of poles $\boldsymbol{p}$ is defined by drawing its real and imaginary parts randomly and independently from a normal distribution $\mathcal{N}(0,16\pi^2 \times 10^18)$ with zero mean and standard deviation of $4\pi \times 10^9$ rad/s. In the cases where the real part of the pole was positive, corresponding to an unstable pole, it was forced to be negative by flipping its sign. However, for this example, the procedure from section 3.4 is applied to tune the other hyperparameters, i.e., the vector of hyperparameters $\lambda = [\gamma, \tau_m, \tau_M]$. This is very handy, e.g., if a small value for the hyperparmeter τ_M is used, the model

may be unable to follow the dynamic phase variations produced by the system. On the other hand, if a more conservative strategy is considered, as an example by using the largest delay interval allowed by the frequency sampling (i.e., $\tau_m = 0\,\text{s}$ and $\tau_M = 1/\Delta_f$), the delay identification procedure can be rather cumbersome, since the values of $W(\tau)$ must be analyzed in a large interval.

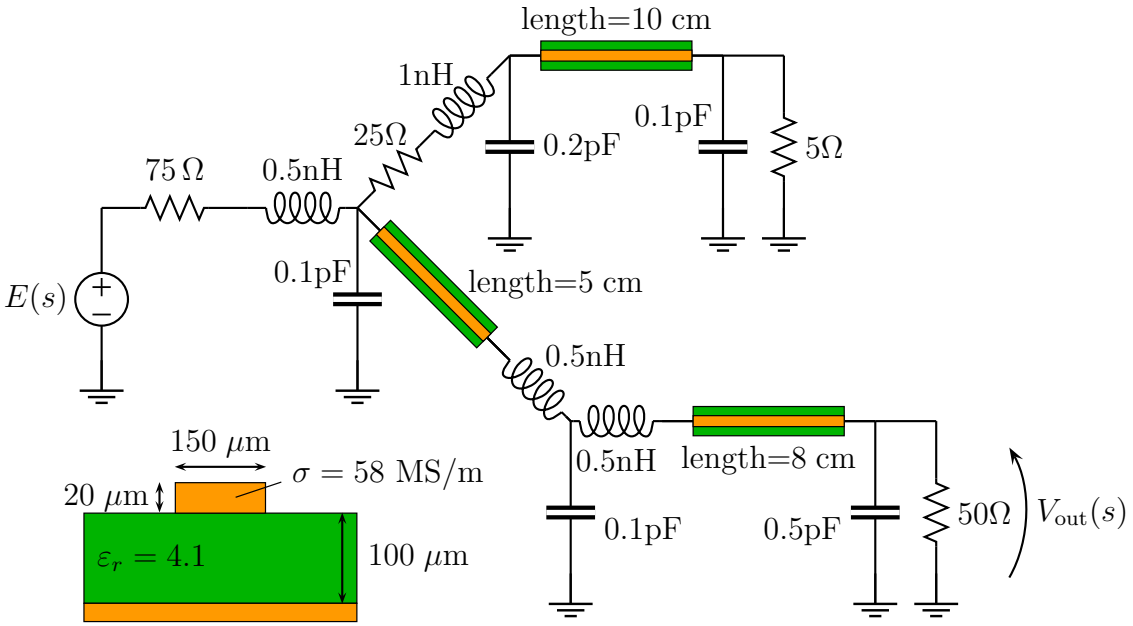


Figure 4.6: Schematic of the circuit modeled in the example 2.

During the model training, the optimum set of hyperparameters γ, τ_m and τ_M is selected via the CV-based Bayesian optimization scheme presented in Sec. 3.4 using 5 folds and 50 iterations. The parameters search space is restricted to the intervals $\gamma \in [10^3, 10^{25}]$ and $\tau_m, \tau_M \in [0,50]$ ns. The constraint $\tau_M > \tau_m$ is not enforced, however it is expected as the logical outcome of the optimization. After the 50 cycles, the obtained optimized parameters are $\gamma^\star = 6.56\times10^{21}$, $\tau_m^\star = 5.68\times10^{-8}$ and $\tau_M^\star = 7.39$ ns. Figures 4.7 and 4.8 show that these parameters provide a very accurate model when applying it to the validation data, where the model output almost perfectly matches the original points of $\boldsymbol{H}(s)$.

Additionally, the proposed LSSVM model of $\boldsymbol{H}(s)$ is used to identify the dominating propagation delays of the system of Fig. 4.6. These delays should be searched only within the optimized delay interval $\tau_m^\star$ to $\tau_M^\star$, making it possible to perform a more fine discretization without incurring into an unreasonable computational time. The delays are identified from $W(\tau)$, which is computed according to (4.63). All this computational procedure took only 940.7 s, of which 937.2 s were used for the estimation of the optimized model and 3.5 s for the delay identification. The plot of $W(\tau)$ is shown in Fig. 4.9. In this figure, the black curve provided by the optimized model is compared with the blue one obtained by means of a basic tuning of the parameters, where τ_m is set to its

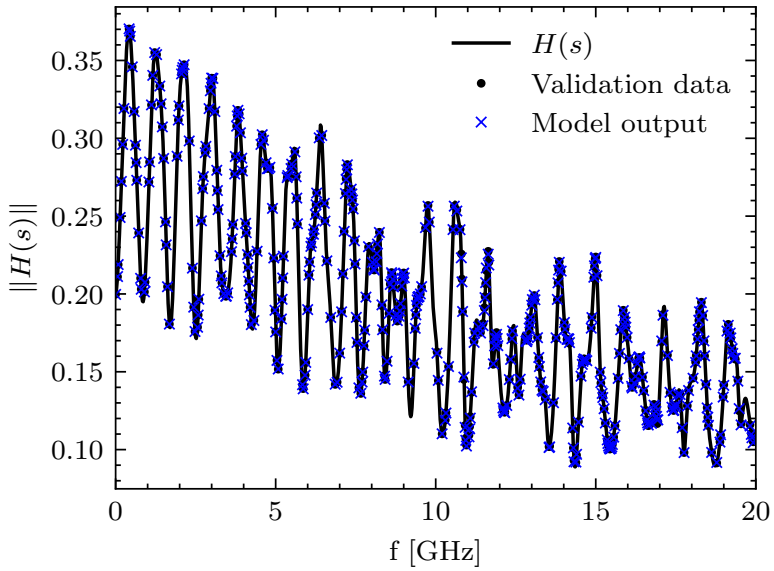


Figure 4.7: Magnitude comparison of the LSSVM model output and the original transfer function $H(s)$.

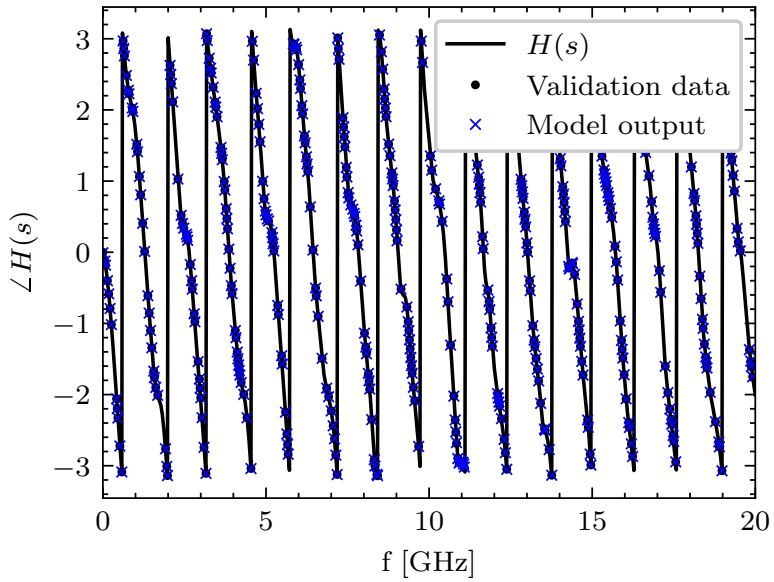


Figure 4.8: Phase comparison of the LSSVM model output and the original transfer function $H(s)$.

minimum and τ_M and γ are set to their maximum possible values. It is observed that the peaks of the black curve are much more identifiable than in the blue curve, while it is also less noisy. Moreover, τ_M in the blue curve goes up to 50 ns, which results

in a worse resolution of the τ-axis if a fixed number of points is considered in the discretization. The peaks of such plot correspond to the propagation delays produced by the original transfer function. For example, the first marked peak occurs at 0.76 ns, while the two transmission lines in the main signal path have a total length of 13 cm. Neglecting the delays introduced by the lumped components of the circuit in Fig 4.6, such values would represent a propagation speed of 1.71×10^8 m/s, compatible with the real speed in such structures. The additional peaks are also clear in the plot, together with smaller peaks that can be identified if necessary.

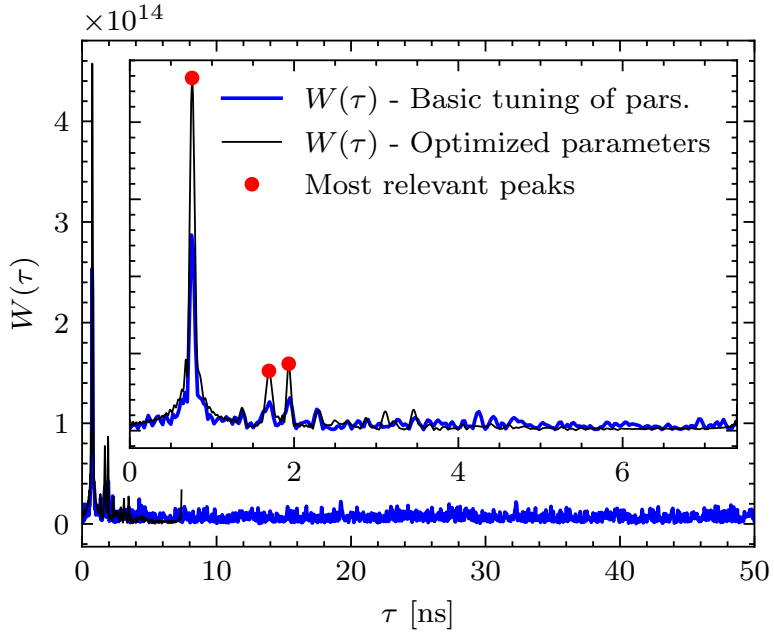


Figure 4.9: Plot of $W(\tau)$ obtained from the LSSVM model of $H(s)$ indicating the dominating propagation delays of the system of Fig. 4.6. The detail amplifies it in the interval from $\tau_m^\star$ to $\tau_M^\star$.

4.6.3 Example 3: Circuit with multiple transmission line paths

For the next example, an extra complexity is added when comparing to example 2: the circuit now has more lumped components and more transmission lines in its constitution [83], as shown in Fig. 4.10. The considered transfer function writes:

$$H(j\omega) = V_o(j\omega)/V_i(j\omega). \tag{4.70}$$

By inspecting the circuit, it can be seen that $H(j\omega)$ should present a larger number of delays, due to multiple propagation paths with distinct lengths between V_i and V_o. This circuit is composed by 5 transmission lines with different lengths (TL1: 0.47 m;

TL2: 1.38 m; TL3: 0.31 m; TL4: 1.19 m; TL5: 0.28 m), resulting in 3 different paths from the input to the output, and additional paths generated by reflections at the various splitting points of the circuit. In addition to the transmission lines, the lumped resistors and capacitors create line mismatches and add corrugations to the voltage and current waveforms.

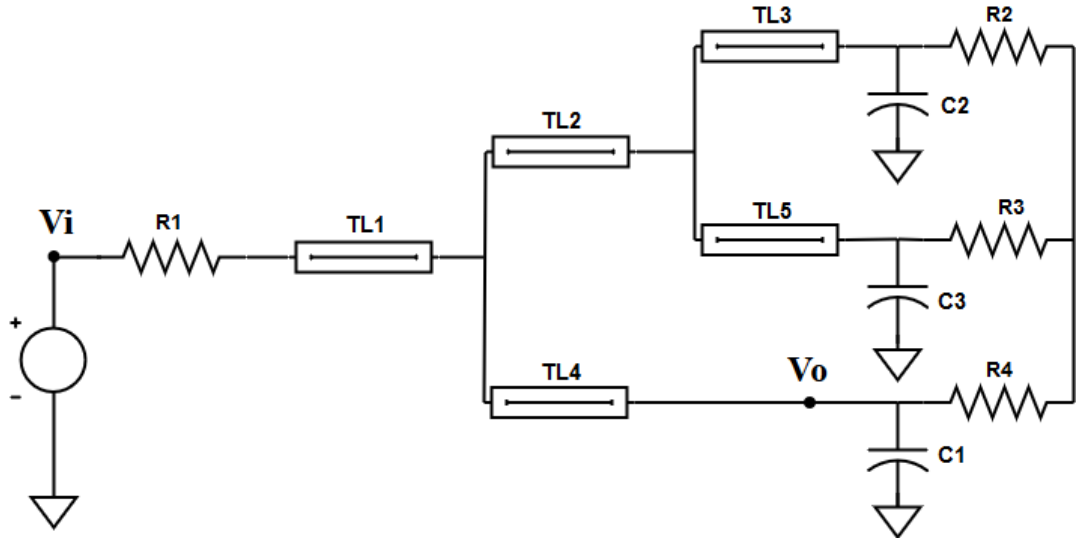


Figure 4.10: Schematic of the circuit used in example 3.

The above circuit is simulated in SPICE, and $H(j\omega)$ is obtained at 1229 equally-spaced frequency points between 0 and 2 GHz. The proposed method is applied in the same sequence as described in example 1, but this time 25 kernel poles are adopted, due to the higher complexity of the transfer function. The obtained time-delay curve is shown in Fig. 4.11. In the same figure, the delays from the Hilbert and Gabor transform methods are also shown, for comparison. The detail of the first peak (shown in an inset of Fig. 4.11) is interesting: it reveals that the peak shape is almost coincident with the Gabor transform curve, but identifies a delay slightly higher than the Hilbert transform. Also for the additional peaks, the time-delay curve computed via the proposed approach presents a remarkable resemblance to the one obtained through the Gabor transform, although all peaks are slightly to the left of the Gabor transform ones.

The curves on the previous figure require the discretization of the τ-axis. Together with the number of frequency samples, this discretization affect the computational time of the curves. Table 4.2 shows how this time changes according to the number of τ points considered, in this example. It is seem that, for a small number of points, the Gabor transform performs better, but as the number of points increases, the LSSVM becomes more efficient. The model estimation does not depend on the τ-axis discretization, and thus, it is computationally easy to compute the weights at a new point, with (4.63). The computational time of the Hilbert transform is higher, and it does not require any discretization.

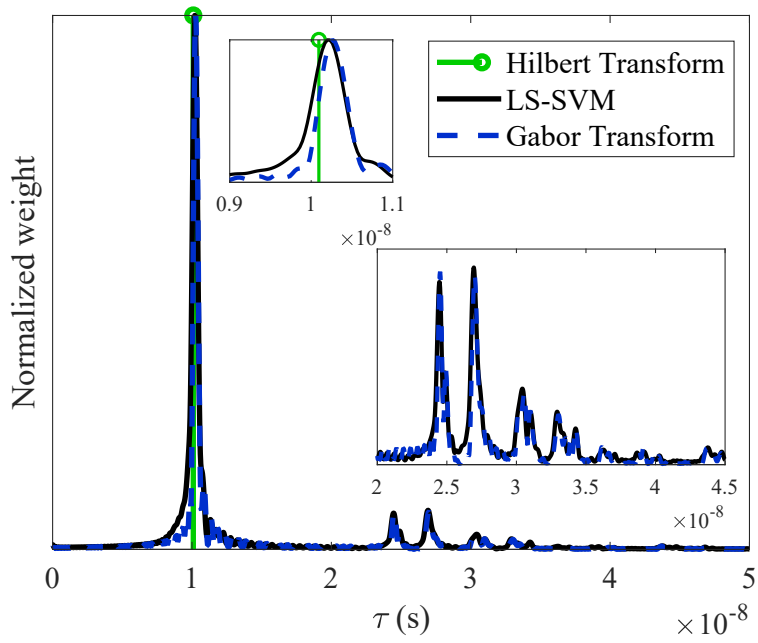


Figure 4.11: Example 3: plots of $w_p(\tau)$ for delay identification; comparison with Hilbert (green line) and Gabor (dashed blue curve) transforms is also displayed.

Table 4.2: Computational time required for delay identification in example 3 as a function of the number of considered points in the τ-axis.

Number of points in the τ-axis	Computational time [s]		
	LSSVM	Gabor transform	Hilbert transform
1,000	27.0	5.36	261.8
3,333	65.3	55.8	
5,000	89.0	135.2	
10,000	165.9	566.0	

Five significant delays are identify from the curve of Fig. 4.11: $\tau_1 = 10.199$ ns, $\tau_2 = 24.453$ ns, $\tau_3 = 26.941$ ns, $\tau_4 = 30.443$ ns, $\tau_5 = 32.931$ ns. Smaller peaks or peaks very close to larger ones are ignored. Those peaks are used now for building a DRM with a finite number of delays, i.e., the kernel-based DRM is used as a replacement for the Gabor transform on the delay identification step of estimating the DRM in (4.12). By using the five identified delays with Algorithm 1, the original transfer function $H(j\omega)$ is fitted using 12 poles in total. This approximation is shown in Fig. 4.12, indicating an excellent phase reconstruction over the entire bandwidth (lower panel).

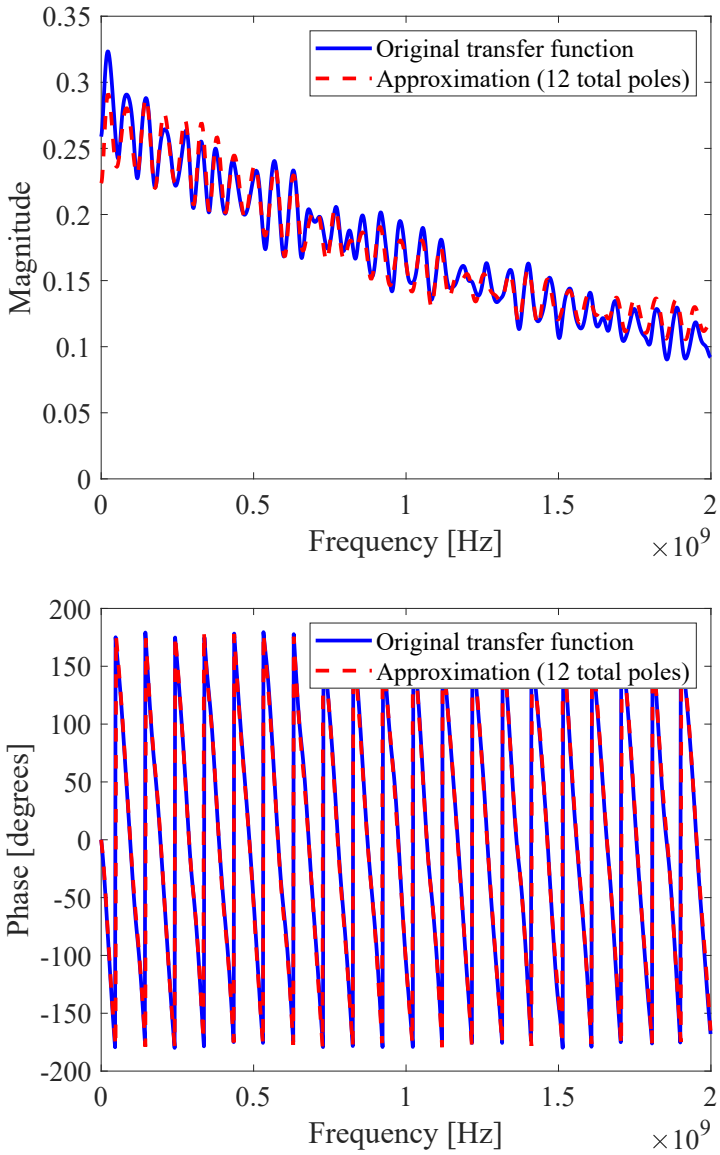


Figure 4.12: Magnitude and phase plots of the reconstructed transfer function of example 3.

In table 4.3, we compare the DRMs obtained using Algorithm 1 and the delays obtained through the proposed LSSVM method, with three alternatives: a DRM obtained considering the delays from the Gabor transform and Algorithm 1; a single delay DRM that uses the Hilbert transform delay and VF to obtain the rational part; and a direct VF approximation. It is shown that a VF approximation with similar accuracy would require at least 40 poles, if the model accounts for the Hilbert transform delay, or 52 poles, if no delay is considered. This is more than three times the number of poles

adopted by the proposed model. Moreover, with a very small change in the delays to the ones identified via the Gabor transform, i.e., $\tau_1 = 10.260$ ns, $\tau_2 = 24.514$ ns, $\tau_3 = 27.002$ ns, $\tau_4 = 30.566$ ns, $\tau_5 = 33.023$ ns, the model obtained using such delays does not achieve a good accuracy. In other words, the Gabor transform method would not be able to achieve the low order model presented in this example, unless it performed an optimization of the obtained delays. The issue seems to rest on τ_1: by replacing it with the LSSVM value, which is only 0.061 ns smaller, a good accuracy is restored. It is believed that $\check{H}_1(j\omega_k)$ in (4.13) might become non-causal when $H(j\omega_k)$ is compensated by a delay larger than its intrinsic delay (i.e., if τ_1 is larger than the true delay in $H(j\omega_k)$). This issue might hinder the rational approximation. Overall, the obtained results demonstrates that the identified delays help reducing the complexity of the model.

Table 4.3: Summary of the error between system response $H(j\omega)$ in example 3 and models $\tilde{H}(j\omega)$ used to approximate it.

	LSSVM + Alg. 1	Gabor tr. + Alg. 1	Hilbert tr. + VF	VF
Error - L_2−norm	0.462	1.940	0.457	0.587
Error - L_∞-norm	0.040	0.138	0.033	0.037
Total poles	12	13	40	52

As a final comparison, we can look at the total computational time required to obtain these DRM or VF models, which is presented in Table 4.4. That data corresponds to the cases where 10,000 τ-values were evaluated to obtain the plot in Fig. 4.11. It is shown that VF is faster, but this was also the method with the highest number of poles. The proposed method took around five times longer than VF, but it requires around 4 times less poles, and had a better performance than the other DRM alternatives.

Table 4.4: Comparison of the computational time required for delay identification and rational fitting through the analyzed methods in example 3.

Method	Computational time [s]		
	Delay id.	Rational fit.	Total
LSSVM + Alg. 2	165.9	0.898	166.8
Gabor tr. + Alg. 2	566.0	0.719	566.7
Hilbert tr. + VF	261.8	0.283	262.1
VF	-	32.31	32.31

4.6.4 Example 4: SpaceWire Cable

Simulated data of the SpaceWire link described in chapter 2, containing a 10 m long SpaceWire cable with compatible connectors and PCB adapters [71] is considered as

a final application example of the presented method. The SpaceWire cable structure is rather complicated (4 twisted-shielded pairs surrounded by an external shield), thus requiring an 18 x 18 scattering matrix to fully describe the electromagnetic behaviour of the link. The simulation accounted for the entire structure of the SpaceWire cable (full details are given in Chapter 2), but, for simplicity, only the scattering parameters of a single wire, from end to end, are modeled here. The 1-wire link is then represented by the following scattering matrix $\mathbf{S}(j\omega)$:

$$S(j\omega) = \begin{bmatrix} S_{1,1}(j\omega) & S_{1,2}(j\omega) \\ S_{2,1}(j\omega) & S_{2,2}(j\omega) \end{bmatrix}, \tag{4.71}$$

where $S_{1,2}(j\omega)$ and $S_{2,1}(j\omega)$ are equal due to the reciprocity of the passive link. The frequency response of this link is sampled at $k = 1, \ldots, 2000$ equally-spaced points between 0 and 1 GHz.

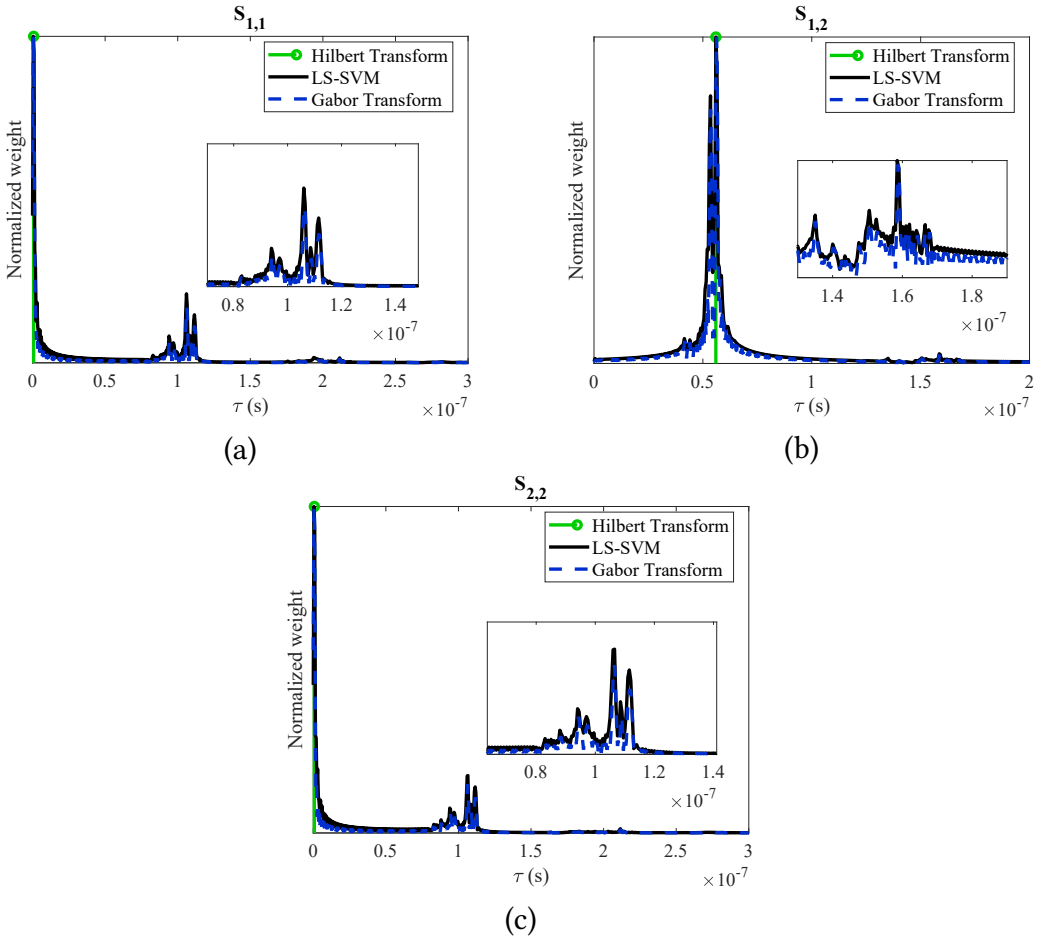


Figure 4.13: Example 4: plots of $W(\tau)$ for delay identification; comparison with Hilbert (green line) and Gabor (dashed blue curve) transforms is also displayed.

The proposed method is applied to each of the elements of the matrix individually, and then a multiport model is achieved by combining the individual results [48]. A remarkable simplification when dealing with multiport systems comes from the fact that the kernel matrix $\mathbf{\Omega}$ depends only on the set of poles $\mathscr{P}$, τ_m, τ_M and the sampled frequency points $\boldsymbol{\omega}_k$, which are the same for all elements of the scattering matrix. The kernel does not depend on the frequency response itself. Therefore, the kernel matrix used to solve (3.24) can be computed only once, simplifying the computational process.

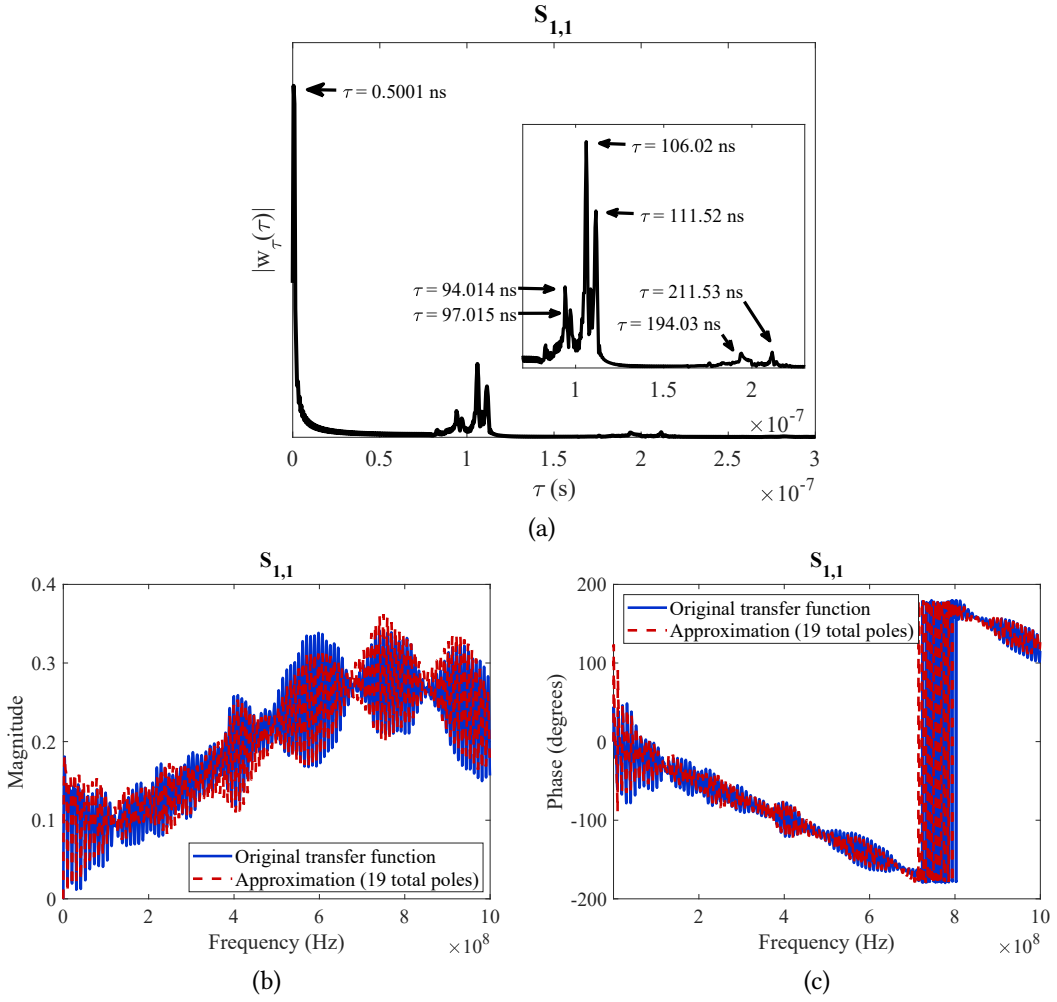


Figure 4.14: Example 4: (a) – Plot of $W(\tau)$ of $S_{1,1}$ for delay identification. (b) and (c) – Comparison of the model with original data for the magnitude and phase of $S_{1,1}(j\omega)$.

The proposed method is applied to $S_{1,1}(j\omega)$, $S_{1,2}(j\omega)$ and $S_{2,2}(j\omega)$ in the same manner as in previous example, with 25 kernel poles, $\tau_m = 0$ and $\tau_M = 1/\Delta_f = 2$ μs. In parallel, and for the sake of comparison, the Hilbert and Gabor transform methods are applied to the scattering matrix of the SPW link. The window considered for the Gabor transform considers $a = 1$. Figure 4.13 shows the results of the three methods,

normalized so that their highest peak has the same value for all three plots. The plots document a very good agreement of the three techniques.

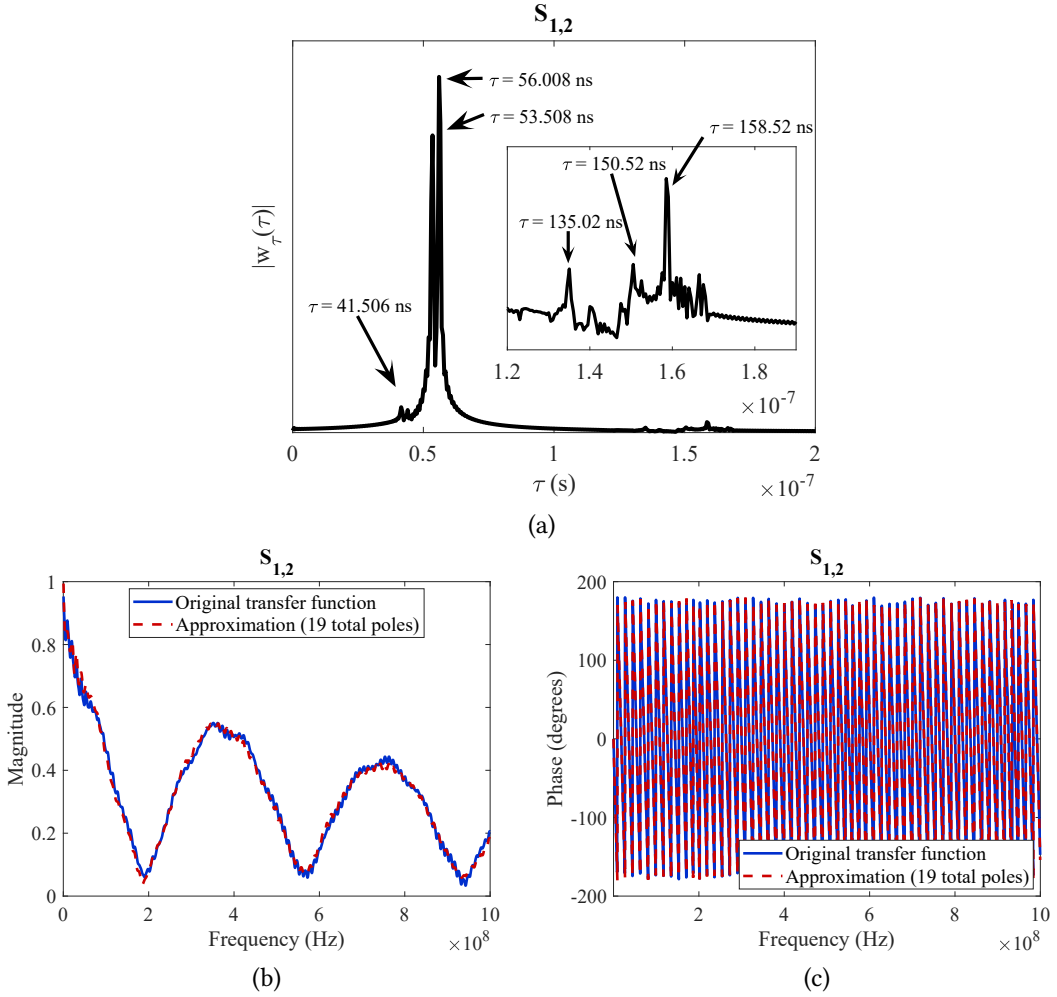


Figure 4.15: Example 4: (a) – Plot of $W(\tau)$ of $S_{1,2}$ for delay identification. (b) and (c) – Comparison of the model with original data for the magnitude and phase of $S_{1,2}(j\omega)$.

For $S_{1,1}(j\omega)$, Fig. 4.14-(a) shows the identified delays, of which the first one happens at $\tau = 0.5$ ns and represents a reflection occurring a few centimeters into the propagation path, where the PCB and connector transitions into the SpaceWire cable. Another 6 delay terms were identified with order of magnitude in the hundreds of ns. Furthermore, $\tau = 0$ is a candidate delay, representing the port reflection, due to a possible mismatches between the port and the input impedance of the structure; however, its identification is compromised by the very close first peak. Figures 4.14-(b) and 4.14-(c) show the magnitude and phase of the $S_{1,1}$ approximation, performed with 8 delays and 19 total poles.

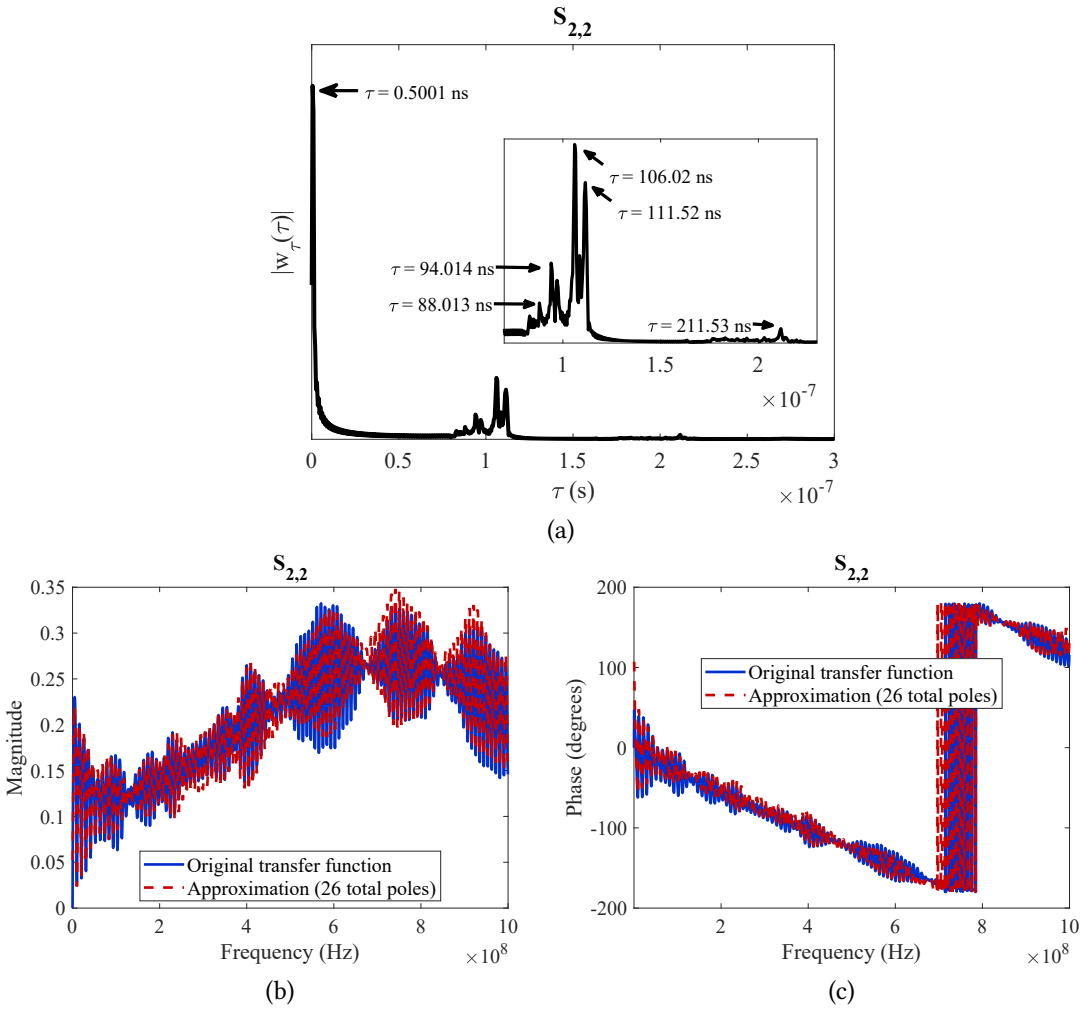


Figure 4.16: Example 4: (a) – Plot of $W(\tau)$ of $S_{2,2}$ for delay identification. (b) and (c) – Comparison of the model with original data for the magnitude and phase of $S_{2,2}(j\omega)$.

For $S_{1,2}(j\omega)$, Fig. 4.15-(a) shows the identified delays. Since $S_{1,2}$ represents the end-to-end transmission of the link, the first peak appears after a considerable delay, contrary to the reflection term $S_{1,1}$. In total, 6 main delays were identified. Figures 4.15-(b) and 4.15-(c) show the approximation of the $S_{1,2}$ transfer function performed with the identified 6 delays and 19 total poles: the good quality of this approximation is evident.

The time-delays of $S_{2,2}$ are identified from the curve displayed in Fig. 4.16-(a). They are similar to the ones of $S_{1,1}$, although not exactly the same, because the link is non-symmetrical. In total, 7 delays terms, including $\tau = 0$ due to direct reflection, are identified. In addition, 26 poles in total are needed for a good quality approximation, as shown in Figs. 4.16-(b) and 4.16-(c).

The three alternative models presented in 4.6.3 are built for comparison. A summary

of the L_2 and L_∞-norm error between the original data and the constructed models is shown in Table 4.5. It is evident that the proposed DRM model presents a much lower complexity when compared with the mere rational VF model. In this example, it achieved a similar performance than the model that uses the Gabor transform delays. The Hilbert transform delay also led to an accurate and low order model, but only for the transmission scattering parameter $S_{1,2}$. For the reflections, this method is not applicable, and thus it would require the high order VF models to build the complete multiport system.

Table 4.5: Summary of the error between available data for the link in Example 4 and models used to approximate it.

	LSSVM + Alg. 1	Gabor tr. + Alg. 1	Hilbert tr. + VF	VF
$S_{1,1}$ error - L_2	0.841	1.052	-	0.894
$S_{1,1}$ error - L_∞	0.098	0.070	-	0.090
$S_{1,1}$ order	19	24	-	159
$S_{1,2}$ error - L_2	0.907	0.905	0.871	0.735
$S_{1,2}$ error - L_∞	0.052	0.052	0.132	0.152
$S_{1,2}$ - order	19	20	35	122
$S_{2,2}$ error - L_2	0.849	1.122	-	0.823
$S_{2,2}$ error - L_∞	0.074	0.074	-	0.083
$S_{2,2}$ - order	26	26	-	162

Table 4.6: Computational time required for delay identification and rational fitting of the multiport system from example 4.

Cumputational time [s]		LSSVM + Alg. 1	Gabor tr. + Alg. 1	Hilbert tr. + VF	VF
Delay estimation	Kernel computation	43.9	-	-	-
	$S_{1,1}$	103.1	126.3	-	-
	$S_{1,2}$	106.0	123.4	443.1	-
	$S_{2,2}$	104.4	123.6	-	-
Rat. fitting	$S_{1,1}$	1.86	1.99	44.06	44.06
	$S_{1,2}$	1.63	1.56	0.16	53.58
	$S_{2,2}$	1.67	2.30	44.06	53.76
Total:		362.6	379.2	531.4	151.4

The total computational time required to build the four aforementioned models is also relevant. It is compared in Table 4.6, for both the delay identification and the rational fitting. We see that the faster complete procedure is VF (which could be even faster

using techniques specific for multiport systems [12]). Nonetheless, the total time spent by the DRM techniques is reasonable, given the remarkable reduction in the number of required poles. Moreover, the fact of computing the kernel matrix only once provides an advantage for the LSSVM in terms of delay estimation time.

4.7 Recapitulation

DRM models are one of the best alternatives to reduce the complexity in models of distributed systems. However, the extended use of such a clever implementation is hindered by the difficulty in estimating accurately both the poles and delays needed for their representation. A new method to assist on the construction of a DRM was presented here. The approach consists in approximating the frequency response of the system by means of the dual space formulation of the LSSVM with an ad-hoc kernel based on an infinite number of delayed-rational basis; and the analysis of the weights of this approximation which allows us to identify the dominant time-delays present in the distributed system, that represent the most critical elements of the model.

Comparing the results of this work with similar existing delay-identification methods, it can be readily observed that the present technique, like the Gabor transform, has the ability to identify multiple propagation delays. However, the Gabor transform suffers from the necessity of tunning the width of its window function, which can push the identified delays to values larger than the actual value. Meanwhile the Hilbert transform is limited to a single delay. The distinctive feature of the discussed approach is its DRM structure, inspired by the physical structure of electronic systems of large dimensions, where the signal propagation is subject to time-delays, losses, attenuation, etc. The inclusion of all these features during the delay-estimation phase leads to delay-values which are more appropriate to the estimation of a low-order DRM.

Moreover, the kernel-based model is obtained via a non-parametric approach, which allows the inclusion of more delay terms than what is feasible via parametric models. Such approach also allows the further optimization via traditional ML techniques which lead to good generalization properties in the obtained model.

Chapter 5

Conclusion

This thesis discussed techniques suitable for the modeling and simulation of electric interconnects. It presented a physical-based approach for the modeling of a SPW cable link, where each component of the link is modeled individually and then their corresponding blocks are cascaded together. Models with different levels of detail are presented, and the modeling approach is validated via experimental measurements of the link's scattering parameters. The comparison between model and measurements show that a high level of detail is necessary in order to obtain a highly accurate model.

On the other hand, this thesis also developed kernel-based black-box models suitable for the modeling of systems with delays such as cables and long interconnects. Kernel-based techniques are a very flexible tool to deal with those systems. Their structure can consider all the effects present in them such as propagation delay, reflections, attenuation, losses, crosstalk, etc. Indeed, they provide a way to model the data using a very large or infinite number of bases. This fact makes it possible to ensure that the true propagation delays of the distributed system are accounted in the estimated model, by establishing the kernel model based on a RKHS which accounts for an infinite number of delayed rational basis functions.

The appropriate kernel described above was designed and applied in the modeling of distributed systems. A machine learning technique called LSSVM is extended to complex-valued data, and used to model the frequency responses of long interconnect structures with the designed kernel. The advantage of such model is that it is a non-parametric model in which the number of parameters that should be estimated is always equal to the number of training samples plus one, and therefore its estimation remains simple even though an infinite number of basis functions is used. The resulting model provides an accurate characterization of the target structures.

When the model is estimated, its parametric equivalent explained in chapter 3 is used to identify the dominant propagation delays of the system. Those delays appears as maxima in a function which depends on the time-delay parameter, and provide accurate results when comparing to the previous available technique for this task which is based on the Gabor transform.

The added flexibility of the machine learning model enables also the tuning of some extra parameters, or hyperparameters. These parameters can be tuned via a k-fold cross validation and a Bayesian optimization, ensuring that the model provides good generalization properties for data not used during the training phase. Such optimization provided an accurate kernel-based model after few iterations. The optimized model provides also an easier way to identify the system dominant propagation delays, where the delays should be searched in a small interval given by the optimized parameters, making it a simpler task than if the whole possible interval would be considered.

Once the delay terms are identified, only the rational part of a DRM needs to be estimated in case a DRM with a finite number of delays is desirable. This can be done through conventional procedures shown in section 4.2, and results in models with much less poles than with conventional techniques. The lower complexity of the resulting DRM implies that an equivalent circuit for the distributed system would require less dynamic components, and thus would be processed more efficiently by time-domain simulators. A clear task that can benefit from this speed-up is the assessment of the integrity of signal propagating in a system with electric interconnects. The DRM has also the advantage of being causal, as positive delays are used in the transfer function; furthermore, its phase is very accurate at all frequencies, while alternative models could lose accuracy for the parts of the transfer function with smaller amplitudes.

5.1 Future Work

The flexibility provided by the infinite dimensional feature space of kernel-based models presented in this work is unique; in fact, the considered poles and time-delay interval can be adapted according to the knowledge about the system, e.g., a model can be estimated first and its parameters can be used to provide information necessary to refine such model. For example, a strategy of handling multiple time-delay intervals centered on the identified delay peaks deserves future investigations. Further studies might also establish ideal poles to be used in the estimation of the DRM.

The integration of the estimated DRM into transient simulations is also an additional necessity for the advance of this technique. A first option is based on the estimation of a DRM with a finite number of delays, such as the SPW link model presented in Example 4 of section 4.6.4, and its synthesis via standard circuit components [48]. An alternative for a direct conversion of a kernel-based model to a SPICE compatible structure could use the capabilities of Verilog-A [108], the numerical inverse Laplace transform (NILT) [109], or a combination of both. In this regard, another useful development might come from the use of kernel-based models known to produce sparse solutions, such as the standard support vector machine (SVM) [58] or the relevance vector machine (RVM) [110]. Such sparse models contain most of the coefficients α_k equal to zero, and thus should gain efficiency if the model is directly employed in a SPICE simulation.

The extension of the proposed models to a large number of parameters should be

straightforward. From the properties in 3.3, a way to enrich the model with additional design parameters is by multiplying the delayed rational kernel by another kernel accounting for the design parameters of the structure according to (3.41e). The Gaussian kernel has already shown good capabilities in modeling systems with a high number of parameters [61], and therefore might be appropriate for this task. However, the dimension of the kernel matrix will grow if additional parameters are added, as the amount of training data will be given by multiplying the size of the frequency response by the amount of parametric training samples. This will require better numerical techniques for data compression and model estimation, as the size of the kernel matrix will be multiplied when the number of parameters increase, and its inversion to solve the system of equations in (3.24) should become impractical. Luckily, there are optimization techniques commonly used on other classes of machine learning problems, but which can be applied directly to optimize the model parameters $\boldsymbol{\alpha}$ in the LSSVM, e.g., the gradient descent algorithm [107]. The implementation of this technique should mitigate the issue of applying the LSSVM on large datasets.

The application of the LSSVM framework as a simple regularized parametric regression might be useful for the iterative algorithms presented in sections 4.1 and 4.2, and therefore deserve some investigation. Specifically, for algorithms that tend to overfit noisy data or in situations where the involved hyperparameters are not estimated yet, the addition of a regularizer might lead to more stable solutions, improving the method in those specific cases.

The kernel-based models presented in this work can be imagined also for systems without the explicit representation of delays. The use of standard kernels presented in section 3.3.3 can already achieve accurate models via the compression of the target frequency response [91]. Nonetheless, the rational models in section 4.1 have been traditionally used to model structures with delay, and it would be natural to propose a kernel based on the RKHS generated by rational basis functions with $p = p' + j\,p''$, according to

$$\mathrm{k}(s, s_k) = \int_{p''_{min}}^{p''_{max}} \int_{p'_{min}}^{p'_{max}} \frac{1}{s - p} \frac{1}{s_k^* - p^*} dp'\,dp''. \tag{5.1}$$

Such approach would lead to rational models with infinite poles, where it would not be required to explicitly define the number and value of such poles. However, the evaluation of such kernel is not simple, and therefore its use in kernel-based models still needs some effort. However, the above kernel-based interpretation can be seen as a promising approach to the application of kernel-based formulations in the modeling of electrical interconnects.

Nomenclature

Acronyms / Abbreviations

ANN	Artificial neural network
AWE	Asymptotic waveform evaluation
CAD	Computer-aided design
CM	Common-mode
CV	Cross validation
DAE	Differential algebraic equations
DM	Differential-mode
DRM	Delayed-rational model
DTFT	Discrete-time Fourier transform
DUT	Device under test
DVF	Delayed vector fitting
EM	Electromagnetic
EMC	Electromagnetic compatibility
EMI	Electromagnetic interference
FFT	Fast-Fourier transform
GPR	Gaussian process regression
LS	Least-squares
LSSVM	Least-squares support vector machine
LVDS	Low-voltage digital signaling

ML Machine learning

MNA Modified nodal analysis

MOC Method of characteristics

MOR Model order reduction

MSE Mean-squared error

MTL Multiconductor transmission line

NILT Numerical inverse Laplace transform

PCA Principal component analysis

PCB Printed circuit board

PUL Per-unit-length

RBF Radial basis functions

RKHS Reproducing kernel Hilbert Space

RVM Relevance vector machine

SK Sanathanan-Koerner

SPI Signal and power integrity

SPICE Simulation program with integrated circuit emphasis

SPW SpaceWire

SSE Sum of squared error

STFT Short-time Fourier transform

SVD Singular value decomposition

SVM Support vector machine

TL Transmission line

VF Vector fitting

VNA Vector network analyzer

WR Waveform relaxation

Bibliography

[1] R. Achar and M. S. Nakhla, "Simulation of High-Speed Interconnects," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 693–728, 2001. DOI: 10.1109/TPWRD.2016.2609039.

[2] G. Moore, "Development, implementation and management of a system level emc design mitigation plan," in *2003 IEEE Symposium on Electromagnetic Compatibility. Symposium Record (Cat. No.03CH37446)*, vol. 1, 2003, 173–176 vol.1. DOI: 10.1109/ISEMC.2003.1236586.

[3] C. R. Paul, *Analysis of multiconductor transmission lines / Clayton R. Paul.* 2nd ed. Hoboken, New Jersey: Wiley-Interscience, 2007, ISBN: 9780470547212.

[4] T. J. Brazil, "Nonlinear, transient simulation of distributed rf circuits using discrete-time convolution," in *2007 IEEE International Symposium on Circuits and Systems*, 2007, pp. 505–508. DOI: 10.1109/ISCAS.2007.378681.

[5] C. Siviero, R. Trinchero, S. Grivet-Talocia, I. S. Stievano, G. Signorini, and M. Telescu, "Constructive Signal Approximations for Fast Transient Simulation of Coupled Channels," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, no. 10, pp. 2087–2096, 2019.

[6] T. Palenius and J. Roos, "Comparison of reduced-order interconnect macromodels for time-domain simulation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 9, pp. 2240–2250, 2004. DOI: 10.1109/TMTT.2004.834562.

[7] V. Loggia, S. Grivet-Talocia, and H. Hu, "Transient Simulation of Complex High-Speed Channels via Waveform Relaxation," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 11, pp. 1823–1838, Nov. 2011, ISSN: 2156-3950. DOI: 10.1109/TCPMT.2011.2167146. [Online]. Available: http://ieeexplore.ieee.org/document/6036159/.

[8] T. Menkad and A. Dounavis, "Resistive Coupling-Based Waveform Relaxation Algorithm for Analysis of Interconnect Circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 7, pp. 1877–1890, Jul. 2017, ISSN: 1549-8328. DOI: 10.1109/TCSI.2017.2665973. [Online]. Available: http://ieeexplore.ieee.org/document/7879107/.

[9] T. Dhaene and D. de Zutter, "Selection of lumped element models for coupled lossy transmission lines," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 7, pp. 805–815, 1992. DOI: 10.1109/43.144845.

[10] F.-Y. Chang, "The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines," *IEEE Transactions on Microwave Theory and Techniques*, vol. 37, no. 12, pp. 2028–2038, 1989, ISSN: 00189480. DOI: 10.1109/22.44118. [Online]. Available: http://ieeexplore.ieee.org/document/44118/.

[11] S. Grivet-Talocia, H.-M. Huang, A. Ruehli, F. Canavero, and I. Elfadel, "Transient analysis of lossy transmission lines: An efficient approach based on the method of characteristics," *IEEE Transactions on Advanced Packaging*, vol. 27, no. 1, pp. 45–56, 2004. DOI: 10.1109/TADVP.2004.825467.

[12] S. Grivet-Talocia and B. Gustavsen, *Passive Macromodeling*, K. Chang, Ed. Hoboken, New Jersey: John Wiley & Sons, Ltd, 2015, ISBN: 9781119140931. DOI: https://doi.org/10.1002/9781119140931. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119140931.

[13] R. Griffith, E. Chiprout, Q. Zhang, and M. Nakhla, "A cad framework for simulation and optimization of high-speed vlsi interconnections," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 39, no. 11, pp. 893–906, 1992. DOI: 10.1109/81.199888.

[14] T. Tang, M. Nakhla, and R. Griffith, "Analysis of lossy multiconductor transmission lines using the asymptotic waveform evaluation technique," *IEEE Transactions on Microwave Theory and Techniques*, vol. 39, no. 12, pp. 2107–2116, 1991. DOI: 10.1109/22.106547.

[15] L. Li, G. Bridges, and I. Ciric, "Efficient simulation of interconnect networks with frequency-dependent lossy transmission lines," *IEEE Microwave and Wireless Components Letters*, vol. 12, no. 4, pp. 131–133, 2002. DOI: 10.1109/7260.993291.

[16] J. Bracken, V. Raghavan, and R. Rohrer, "Interconnect simulation with asymptotic waveform evaluation (awe)," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 39, no. 11, pp. 869–878, 1992. DOI: 10.1109/81.199886.

[17] A. Dounavis, X. Li, M. Nakhla, and R. Achar, "Passive closed-form transmission-line model for general-purpose circuit simulators," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 12, pp. 2450–2459, 1999. DOI: 10.1109/22.808992.

[18] A. Dounavis, R. Achar, and M. Nakhla, "Efficient passive circuit models for distributed networks with frequency-dependent parameters," *IEEE Transactions on Advanced Packaging*, vol. 23, no. 3, pp. 382–392, 2000. DOI: 10.1109/6040.861551.

[19] D. Kuznetsov and J. Schutt-Aine, "Optimal transient simulation of transmission lines," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 43, no. 2, pp. 110–121, 1996, ISSN: 10577122. DOI: 10.1109/81.486433. [Online]. Available: http://ieeexplore.ieee.org/document/486433/.

[20] W. Beyene and J. Schutt-Aine, "Accurate frequency-domain modeling and efficient circuit simulation of high-speed packaging interconnects," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 10, pp. 1941–1947, 1997, ISSN: 00189480. DOI: 10.1109/22.641798. [Online]. Available: http://ieeexplore.ieee.org/document/641798/.

[21] D. Romano, G. Antonini, U. Grossner, and I. Kovačević-Badstübner, "Circuit synthesis techniques of rational models of electromagnetic systems: A tutorial paper," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 32, no. 5, e2612, 2019, e2612 jnm.2612. DOI: https://doi.org/10.1002/jnm.2612. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jnm.2612. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jnm.2612.

[22] C.-C. Chou and J. E. Schutt-Ainé, "Equivalent circuit synthesis of multiport s parameters in pole–residue form," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 11, pp. 1971–1979, 2021. DOI: 10.1109/TCPMT.2021.3115113.

[23] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Transactions on Power Delivery*, vol. 14, no. 3, pp. 1052–1061, Jul. 1999, ISSN: 08858977. DOI: 10.1109/61.772353. [Online]. Available: http://ieeexplore.ieee.org/document/772353/.

[24] I. Stevanovic, B. Wunsch, G. L. Madonna, and S. Skibin, "High-Frequency Behavioral Multiconductor Cable Modeling for EMI Simulations in Power Electronics," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1392–1400, May 2014, ISSN: 1551-3203. DOI: 10.1109/TII.2014.2307198. [Online]. Available: http://ieeexplore.ieee.org/document/6750077/.

[25] Y. Ye, D. Spina, Y. Xing, W. Bogaerts, and T. Dhaene, "Numerical modeling of a linear photonic system for accurate and efficient time-domain simulations," *Photon. Res.*, vol. 6, no. 6, pp. 560–573, Jun. 2018. DOI: 10.1364/PRJ.6.000560. [Online]. Available: http://www.osapublishing.org/prj/abstract.cfm?URI=prj-6-6-560.

[26] S. Roy and A. Dounavis, "Transient Simulation of Distributed Networks Using Delay Extraction Based Numerical Convolution," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 3, pp. 364–373, Mar. 2011, ISSN: 0278-0070. DOI: 10.1109/TCAD.2010.2090065. [Online]. Available: http://ieeexplore.ieee.org/document/5715600/.

[27] A. Charest, D. Saraswat, M. Nakhla, R. Achar, and N. Soveiko, "Compact Macromodeling of High-Speed Circuits via Delayed Rational Functions," *IEEE Microwave and Wireless Components Letters*, vol. 17, no. 12, pp. 828–830, Dec. 2007, ISSN: 1531-1309. DOI: 10.1109/LMWC.2007.910468. [Online]. Available: http://ieeexplore.ieee.org/ielx5/7260/4385748/04385732.pdf?tp=%7B%5C&%7Darnumber=4385732%7B%5C&%7Disnumber=4385748%7B%5C%%7D5Cnhttp://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=%7B%5C&%7Darnumber=4385732%7B%5C&%7DqueryText%7B%5C%%7D3Dcharest+%7B%5C%%7D22compact+macromodeling%7B%5C%%7D22%20http://ieeexplore.ieee.org/document/.

[28] S. Grivet-Talocia, "Delay-Based Macromodels for Long Interconnects via Time-Frequency Decompositions," in *2006 IEEE Electrical Performane of Electronic Packaging*, IEEE, Oct. 2006, pp. 199–202, ISBN: 1-4244-0668-4. DOI: 10.1109/EPEP.2006.321228. [Online]. Available: http://ieeexplore.ieee.org/document/4115388/.

[29] A. Kahng and S. Muddu, "An analytical delay model for rlc interconnects," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 12, pp. 1507–1514, 1997. DOI: 10.1109/43.664231.

[30] M. S. Ullah and M. H. Chowdhury, "Analytical models of high-speed rlc interconnect delay for complex and real poles," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 6, pp. 1831–1841, 2017. DOI: 10.1109/TVLSI.2017.2654921.

[31] Bing Zhong, Tao Hu, D. Fu, S. Dvorak, and J. Prince, "A study of a hybrid phase-pole macromodel for transient simulation of complex interconnects structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 8, pp. 1250–1261, Aug. 2005, ISSN: 0278-0070. DOI: 10.1109/TCAD.2005.850817. [Online]. Available: http://ieeexplore.ieee.org/document/1487564/.

[32] B. Gustavsen, "Time delay identification for transmission line modeling," in *Proceedings. 8th IEEE Workshop on Signal Propagation on Interconnects*, Heidelberg, Germany: IEEE, 2004, pp. 103–106, ISBN: 0-7803-8470-9. DOI: 10.1109/SPI.2004.1409018. arXiv: 1011.1669v3. [Online]. Available: http://ieeexplore.ieee.org/document/6222912/%20http://ieeexplore.ieee.org/document/1409018/.

[33] B. Gustavsen and J. Nordstrom, "Pole Identification for The Universal Line Model Based on Trace Fitting," *IEEE Transactions on Power Delivery*, vol. 23, no. 1, pp. 472–479, Jan. 2008, ISSN: 0885-8977. DOI: 10.1109/TPWRD.2007.911186. [Online]. Available: http://ieeexplore.ieee.org/document/4408686/.

[34] B. Gustavsen, "Optimal Time Delay Extraction for Transmission Line Modeling," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 45–54, 2017. DOI: 10.1109/TPWRD.2016.2609039.

[35] J. Cho, J. Ahn, J. Kim, *et al.*, "Low- and High-Frequency Extrapolation of Band-Limited Frequency Responses to Extract Delay Causal Time Responses," *IEEE Transactions on Electromagnetic Compatibility*, vol. 63, no. 3, pp. 888–901, Jun. 2021, ISSN: 0018-9375. DOI: 10.1109/TEMC.2020.3025132. [Online]. Available: https://ieeexplore.ieee.org/document/9214516/.

[36] R. Mandrekar and M. Swaminathanz, "Causality Enforcement in Transient Simulation of Passive Networks through Delay Extraction," in *Proceedings. 9th IEEE Workshop on Signal Propagation on Interconnects*, Garmisch-Partenkirchen, Germany, 2005, pp. 25–28, ISBN: 0780390547. DOI: 10.1109/SPI.2005.1500884.

[37] S. N. Lalgudi, E. Engin, G. Casinovi, and M. Swaminathan, "Accurate Transient Simulation of Interconnects Characterized by Band-Limited Data With Propagation Delay Enforcement in a Modified Nodal Analysis Framework," *IEEE Transactions on Electromagnetic Compatibility*, vol. 50, no. 3, pp. 715–729, Aug. 2008, ISSN: 0018-9375. DOI: 10.1109/TEMC.2008.924394. [Online]. Available: http://ieeexplore.ieee.org/document/4603133/.

[38] A. P. Reghu, R. V. Sanjika Devi, K. Vrinda, and D. G. Kurup, "Macromodeling of high frequency interconnects based on accurate delay identification," in *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2018, pp. 1636–1640. DOI: 10.1109/ICACCI.2018.8554721.

[39] L. Barannyk, H. Tran, A. Elshabini, and F. Barlow, "Time Delay Extraction from Frequency Domain Data Using Causal Fourier Continuations for High-Speed Interconnects," *Electronics*, vol. 4, no. 4, pp. 799–826, Oct. 2015, ISSN: 2079-9292. DOI: 10.3390/electronics4040799. [Online]. Available: http://www.mdpi.com/2079-9292/4/4/799.

[40] A. Charest, M. Nakhla, and R. Achar, "Delay Extracted Stable Rational Approximations for Tabulated Networks With Periodic Reflections," *IEEE Microwave and Wireless Components Letters*, vol. 19, no. 12, pp. 768–770, Dec. 2009, ISSN: 1531-1309. DOI: 10.1109/LMWC.2009.2033493. [Online]. Available: http://ieeexplore.ieee.org/document/5320191/.

[41] P. Triverio, S. Grivet-Talocia, and A. Chinea, "Identification of Highly Efficient Delay-Rational Macromodels of Long Interconnects From Tabulated Frequency Data," *IEEE Transactions on Microwave Theory and Techniques*, vol. 58, no. 3, pp. 566–577, Mar. 2010, ISSN: 0018-9480. DOI: 10.1109/TMTT.2010.2040349. [Online]. Available: http://ieeexplore.ieee.org/document/5406045/.

[42] A. Chinea, S. Grivet-Talocia, H. Hu, *et al.*, "Signal Integrity Verification of Multi-chip Links Using Passive Channel Macromodels," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 6, pp. 920–933, Jun. 2011, ISSN: 2156-3950. DOI: 10.1109/TCPMT.2011.2138136. [Online]. Available: http://ieeexplore.ieee.org/document/5779713/.

[43] A. Charest, M. Nakhla, R. Achar, D. Saraswat, N. Soveiko, and I. Erdin, "Time Domain Delay Extraction-Based Macromodeling Algorithm for Long-Delay Networks," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 1, pp. 219–235, Feb. 2010, ISSN: 1521-3323. DOI: 10.1109/TADVP.2009.2029560. [Online]. Available: http://ieeexplore.ieee.org/document/5286253/.

[44] M. Sahouli and A. Dounavis, "Delay Extraction-Based Modeling Using Loewner Matrix Framework," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 7, no. 3, pp. 424–433, Mar. 2017, ISSN: 2156-3950. DOI: 10.1109/TCPMT.2017.2650138. [Online]. Available: http://ieeexplore.ieee.org/document/7833170/.

[45] ——, "Delay rational macromodels of long interconnects using Loewner Matrix," in *2016 IEEE 25th Conference on Electrical Performance Of Electronic Packaging And Systems (EPEPS)*, IEEE, Oct. 2016, pp. 103–106, ISBN: 978-1-5090-6110-5. DOI: 10.1109/EPEPS.2016.7835428. [Online]. Available: http://ieeexplore.ieee.org/document/7835428/.

[46] M. De Lauretis, G. Antonini, and J. Ekman, "A delay-rational model of lossy multiconductor transmission lines with frequency-independent per-unit-length parameters," *IEEE Transactions on Electromagnetic Compatibility*, vol. 57, no. 5, pp. 1235–1245, 2015. DOI: 10.1109/TEMC.2015.2423327.

[47] E. Biello, G. Antonini, M. De Lauretis, and J. Ekman, "A delay-rational model of electromagnetic interference on multiconductor transmission lines," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 31, no. 4, e2342, 2018, e2342 jnm.2342. DOI: https://doi.org/10.1002/jnm.2342. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jnm.2342. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/jnm.2342.

[48] A. Chinea, P. Triverio, and S. Grivet-Talocia, "Delay-Based Macromodeling of Long Interconnects From Frequency-Domain Terminal Responses," *IEEE Transactions on Advanced Packaging*, vol. 33, no. 1, pp. 246–256, Feb. 2010, ISSN: 1521-3323. DOI: 10.1109/TADVP.2008.2010525. [Online]. Available: http://ieeexplore.ieee.org/document/4811943/.

[49] M. Luo and K.-M. Huang, "An Extended Delay-Rational Macromodel for Electromagnetic Interference Analysis of Mixed Signal Circuits," *Progress In Electromagnetics Research*, vol. 127, no. April, pp. 189–210, 2012, ISSN: 1559-8985. DOI: 10.2528/PIER12022406. [Online]. Available: http://www.jpier.org/PIER/pier.php?paper=12022406.

[50] M. Sgueglia, A. Sorrentino, M. de Magistris, D. Spina, D. Deschrijver, and T. Dhaene, "A novel parametric macromodeling technique for electromagnetic structures with propagation delays," in *2017 IEEE 21st Workshop on Signal and Power Integrity (SPI)*, IEEE, May 2017, pp. 1–4, ISBN: 978-1-5090-5616-3. DOI: 10.1109/SaPIW.2017.7944016. [Online]. Available: http://ieeexplore.ieee.org/document/7944016/.

[51] Y. Zyari, M. Rolain, F. Ferranti, G. Vandersteen, and P. Bronders, "Multi-delay rational modeling of lumped-distributed systems," in *2017 IEEE MTT-S International Microwave Symposium (IMS)*, IEEE, Jun. 2017, pp. 1624–1627, ISBN: 978-1-5090-6360-4. DOI: 10.1109/MWSYM.2017.8058945. [Online]. Available: http://ieeexplore.ieee.org/document/8058945/.

[52] M. Zyari, Y. Rolain, and F. Ferranti, "Rational multi-delay models for long interconnects," in *2017 IEEE 21st Workshop on Signal and Power Integrity (SPI)*, IEEE, May 2017, pp. 1–4, ISBN: 978-1-5090-5616-3. DOI: 10.1109/SaPIW.2017.7944015. [Online]. Available: http://ieeexplore.ieee.org/document/7944015/.

[53] J. Cullum, A. Ruehli, and T. Zhang, "A method for reduced-order modeling and simulation of large interconnect circuits and its application to peec models with retardation," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 4, pp. 261–273, 2000. DOI: 10.1109/82.839662.

[54] W. Michiels, E. Jarlebring, and K. Meerbergen, "Krylov-based model order reduction of time-delay systems," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 4, pp. 1399–1421, 2011. DOI: 10.1137/100797436. eprint: https://doi.org/10.1137/100797436. [Online]. Available: https://doi.org/10.1137/100797436.

[55] E. R. Samuel, L. Knockaert, and T. Dhaene, "Model Order Reduction of Time-Delay Systems Using a Laguerre Expansion Technique," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 6, pp. 1815–1823, Jun. 2014, ISSN: 1549-8328. DOI: 10.1109/TCSI.2013.2295011. [Online]. Available: http://ieeexplore.ieee.org/document/6728665/.

[56] K.-T. Fang, R. Li, and A. Sudjianto, *Design and Modeling for Computer Experiments (Computer Science & Data Analysis)*. Chapman & Hall/CRC, 2005, ISBN: 1584885467.

[57] K. P. Murphy, *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013, ISBN: 9780262018029 0262018020.

[58] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, 1st ed. Cambridge University Press, 2000, ISBN: 0521780195.

[59] R. Trinchero, P. Manfredi, I. S. Stievano, and F. G. Canavero, "Machine Learning for the Performance Assessment of High-Speed Links," *IEEE Transactions on Electromagnetic Compatibility*, vol. 60, no. 6, pp. 1627–1634, 2018.

[60] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. D. Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific, 2002, pp. 1–308, ISBN: 978-981-4487-59-7.

[61] R. Trinchero, M. Larbi, H. M. Torun, F. G. Canavero, and M. Swaminathan, "Machine Learning and Uncertainty Quantification for Surrogate Models of Integrated Devices With a Large Number of Parameters," *IEEE Access*, vol. 7, pp. 4056–4066, 2019.

[62] R. Trinchero and F. G. Canavero, "Combining LS-SVM and GP Regression for the Uncertainty Quantification of the EMI of Power Converters Affected by Several Uncertain Parameters," *IEEE Transactions on Electromagnetic Compatibility*, pp. 1–8, 2020.

[63] Z. A. Memon, R. Trinchero, P. Manfredi, F. Canavero, and I. S. Stievano, "Compressed machine learning models for the uncertainty quantification of power distribution networks," *Energies*, vol. 13, no. 18, 2020, ISSN: 1996-1073. DOI: 10.3390/en13184881. [Online]. Available: https://www.mdpi.com/1996-1073/13/18/4881.

[64] T. Nguyen, T. Lu, K. Wu, and J. Schutt-Aine, "Fast Transient Simulation of High-Speed Channels Using Recurrent Neural Network," pp. 1–11, Jan. 2019. arXiv: 1902.02627. [Online]. Available: http://arxiv.org/abs/1902.02627.

[65] S. Parkes, P. Armbruster, and M. Suess, "The SpaceWire on-board data-handling network," *European Space Agency, Bulletin 145*, pp. 28–39, Feb. 2011.

[66] "ECSS-E-ST-50-12C Rev.1 - SpaceWire – Links, nodes, routers and networks," *ESA Requirements and Standards Division*, pp. 1–137, 2019.

[67] D. Bockelman and W. Eisenstadt, "Combined differential and common-mode scattering parameters: theory and simulation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 43, no. 7, pp. 1530–1539, Jul. 1995, ISSN: 00189480. DOI: 10.1109/22.392911. [Online]. Available: http://ieeexplore.ieee.org/document/392911/.

[68] "ESCC Detail Specification No. 3401/071 - Connector, Electrical, Rectangular, Microminiature, Solder Bucket Contacts, with EMI Backshell - Based on Type MDM," *European Space Agency*, pp. 1–25, 2013.

[69] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. Springer, 2013, ISBN: 978-1-4614-6848-6.

[70] K. Enouf and S. Hermant, "Compact, impedance-matched spacewire connector development — "micromach spacewire": Spacewire components, long paper," in *2016 International SpaceWire Conference (SpaceWire)*, 2016, pp. 1–7. DOI: 10.1109/SpaceWire.2016.7771640.

[71] F. Treviso, R. Trinchero, and F. G. Canavero, "Validation of a Physical-Based Model for a Spacewire Cable," in *2019 ESA Workshop on Aerospace EMC (Aerospace EMC)*, Budapest, Hungary: IEEE, May 2019, pp. 1–6, ISBN: 978-9-0826-8478-0. DOI: 10.23919/AeroEMC.2019.8788945. [Online]. Available: https://ieeexplore.ieee.org/document/8788945/.

[72] M. Suess, S. Parkes, and P. Crawford, "SpaceWire Cable Characterization," in *International SpaceWire Convention 2007 (ISC 2007)*, Dundee, Scotland, 2007, pp. 1–8.

[73] A. Huynh, M. Karlsson, and S. Gong, "Mixed-mode s-parameters and conversion techniques," in *Advanced Microwave Circuits and Systems*, V. Zhurbenko, Ed., Rijeka: IntechOpen, 2010, ch. 1. DOI: 10.5772/8419. [Online]. Available: https://doi.org/10.5772/8419.

[74] J. Tippet and R. Speciale, "A Rigorous Technique for Measuring the Scattering Matrix of a Multiport Device with a 2-Port Network Analyzer," *IEEE Transactions on Microwave Theory and Techniques*, vol. 30, no. 5, pp. 661–666, May 1982, ISSN: 0018-9480. DOI: 10.1109/TMTT.1982.1131118. [Online]. Available: http://ieeexplore.ieee.org/document/1131118/.

[75] T. Ruttan, B. Grossman, A. Ferrero, V. Teppati, and J. Martens, "Multiport VNA measurement," *IEEE Microwave Magazine*, vol. 9, no. 3, pp. 56–69, Jun. 2008, ISSN: 1527-3342. DOI: 10.1109/MMM.2008.919919. [Online]. Available: http://ieeexplore.ieee.org/document/4519496/.

[76] M. Sedaghat, R. Trinchero, and F. Canavero, "Compressed machine learning-based inverse model for the design of microwave filters," in *2021 IEEE MTT-S International Microwave Symposium (IMS)*, 2021, pp. 13–15. DOI: 10.1109/IMS19712.2021.9574884.

[77] M. Larbi, R. Trinchero, F. G. Canavero, P. Besnier, and M. Swaminathan, "Analysis of Parameter Variability in an Integrated Wireless Power Transfer System via Partial Least Squares Regression," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, p. 1, 2020.

[78] B. Sudret, S. Marelli, and J. Wiart, "Surrogate models for uncertainty quantification: An overview," in *2017 11th European Conference on Antennas and Propagation (EUCAP)*, 2017, pp. 793–797. DOI: 10.23919/EuCAP.2017.7928679.

[79] Y. Ye, D. Spina, P. Manfredi, D. Vande Ginste, and T. Dhaene, "A comprehensive and modular stochastic modeling framework for the variability-aware assessment of signal integrity in high-speed links," *IEEE Trans.Electromagn. Compat.*, vol. 60, no. 2, pp. 459–467, 2018.

[80] P. Manfredi, D. V. Ginste, I. S. Stievano, D. De Zutter, and F. G. Canavero, "Stochastic transmission line analysis via polynomial chaos methods: an overview," *IEEE Electromagnetic Compatibility Magazine*, vol. 6, no. 3, pp. 77–84, 2017. DOI: 10.1109/MEMC.0.8093844.

[81] F. Treviso, R. Trinchero, P. Keski-Opas, I. Kelander, and F. G. Canavero, "Sensitivity analysis of passive intermodulation due to electrical contacts," *IEEE Transactions on Electromagnetic Compatibility*, pp. 1–10, 2022. DOI: 10.1109/TEMC.2022.3142963.

[82] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.

[83] F. Treviso, R. Trinchero, and F. G. Canavero, "Multiple Delay Identification in Long Interconnects via LS-SVM Regression," *IEEE Access*, vol. 9, pp. 39 028–39 042, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3063713. [Online]. Available: https://ieeexplore.ieee.org/document/9369356/.

[84] P. Bouboulis, S. Theodoridis, C. Mavroforakis, and L. Evaggelatou-Dalla, "Complex Support Vector Machines for Regression and Quaternary Classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 6, pp. 1260–1274, Jun. 2015, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2014.2336679. [Online]. Available: https://ieeexplore.ieee.org/document/6868310.

[85] R. Boloix-Tortosa, J. J. Murillo-Fuentes, I. Santos, and F. Perez-Cruz, "Widely Linear Complex-Valued Kernel Methods for Regression," *IEEE Transactions on Signal Processing*, vol. 65, no. 19, pp. 5240–5248, Oct. 2017, ISSN: 1053-587X. DOI: 10.1109/TSP.2017.2726991. [Online]. Available: http://ieeexplore.ieee.org/document/7979572/.

[86] F. A. Tobar, A. Kuh, and D. P. Mandic, "A novel augmented complex valued kernel lms," in *2012 IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2012, pp. 473–476. DOI: 10.1109/SAM.2012.6250542.

[87] R. Boloix-Tortosa, J. J. Murillo-Fuentes, and S. A. Tsaftaris, "The generalized complex kernel least-mean-square algorithm," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5213–5222, 2019. DOI: 10.1109/TSP.2019.2937289.

[88] S. De Iaco, "New spatio-temporal complex covariance functions for vectorial data through positive mixtures," *Stochastic Environmental Research and Risk Assessment*, Feb. 2022, ISSN: 1436-3259. DOI: 10.1007/s00477-022-02171-9. [Online]. Available: https://doi.org/10.1007/s00477-022-02171-9.

[89] F. Treviso, R. Trinchero, and F. G. Canavero, "Machine Learning Applied to the Blind Identification of Multiple Delays in Distributed Systems," in *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*, Rome, Italy: IEEE, Aug. 2020, pp. 1–4, ISBN: 978-9-4639-6800-3. DOI: 10.23919/URSIGASS49373.2020.9232325. [Online]. Available: https://ieeexplore.ieee.org/document/9232325/.

[90] ——, "Bayesian Optimization of Hyperparameters in Kernel-Based Delay Rational Models," *IEEE Electromagnetic Compatibility Magazine*, vol. 10, no. 2, pp. 90–93, 2021, ISSN: 2162-2264. DOI: 10.1109/MEMC.2021.9477255. [Online]. Available: https://ieeexplore.ieee.org/document/9477255/.

[91] 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, 2022, ISSN: 2079-9292. DOI: 10.3390/electronics11040551. [Online]. Available: https://www.mdpi.com/2079-9292/11/4/551.

[92] D. Posa, "Parametric families for complex valued covariance functions: Some results, an overview and critical aspects," *Spatial Statistics*, vol. 39, p. 100 473, 2020, ISSN: 2211-6753. DOI: https://doi.org/10.1016/j.spasta.2020.100473.

[93] S. Saitoh and Y. Sawano, *Theory of reproducing kernels and applications*. Springer, 2016.

[94] C. R. Souza, *Kernel Functions for Machine Learning Applications*, Mar. 2010. [Online]. Available: http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications (visited on 02/10/2022).

[95] M. Genton, "Classes of kernels for machine learning: A statistics perspective.," *Journal of Machine Learning Research*, vol. 2, pp. 299–312, Jan. 2001. DOI: 10.1162/15324430260185646.

[96] S. Boughorbel, J.-P. Tarel, and N. Boujemaa, "Conditionally positive definite kernels for svm based image recognition," in *2005 IEEE International Conference on Multimedia and Expo*, 2005, pp. 113–116. DOI: 10.1109/ICME.2005.1521373.

[97] S. De Iaco, M. Palma, and D. Posa, "Covariance functions and models for complex-valued random fields," *Stochastic Environmental Research and Risk Assessment*, vol. 17, no. 3, pp. 145–156, Sep. 2003, ISSN: 1436-3259. DOI: 10.1007/s00477-003-0129-5. [Online]. Available: https://doi.org/10.1007/s00477-003-0129-5.

[98] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi, *Scikit-optimize/scikit-optimize*, Sep. 2020. DOI: 10.5281/zenodo.4014775. [Online]. Available: https://doi.org/10.5281/zenodo.4014775.

[99] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016. DOI: 10.1109/JPROC.2015.2494218.

[100] E. Brochu, V. M. Cora, and N. de Freitas, *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*, 2010. arXiv: 1012.2599 [cs.LG].

[101] B. Ghojogh and M. Crowley, *The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial*, 2019. arXiv: 1905.12787 [stat.ML].

[102] M. Espinoza, J. A. K. Suykens, R. Belmans, and B. De Moor, "Electric load forecasting," *IEEE Control Systems Magazine*, vol. 27, no. 5, pp. 43–57, 2007.

[103] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing (2nd Ed.)* USA: Prentice-Hall, Inc., 1999, ISBN: 0137549202.

[104] P. Triverio, S. Grivet-Talocia, M. S. Nakhla, F. G. Canavero, and R. Achar, "Stability, causality, and passivity in electrical interconnect models," *IEEE Transactions on Advanced Packaging*, vol. 30, no. 4, pp. 795–808, 2007. DOI: 10.1109/TADVP.2007.901567.

[105] A. Mertins, *Signal Analysis*. John Wiley & Sons, Ltd, 1999, ISBN: 9780470841839. DOI: https://doi.org/10.1002/0470841834. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/0470841834.

[106] D. Hughes-Hallett, *Calculus: Single and Multivariable*. Wiley, 2008, ISBN: 9780470089149. [Online]. Available: https://books.google.it/books?id=0llEQgAACAAJ.

[107] N. Soleimani, R. Trinchero, and F. Canavero, "Vector-Valued Kernel Ridge Regression for the Modeling of High-Speed Links," Limoges, France, Jul. 2022.

[108] A. Krim, A. Lakrim, and D. Tahri, "Two vhdl-ams-based models of multi-conductor power cables for emi simulations," *Electrical Engineering*, vol. 103, no. 1, pp. 647–661, Feb. 2021, ISSN: 1432-0487. DOI: 10.1007/s00202-020-01108-1. [Online]. Available: https://doi.org/10.1007/s00202-020-01108-1.

[109] P. Moreno and A. Ramirez, "Implementation of the numerical laplace transform: A review task force on frequency domain methods for emt studies, working group on modeling and analysis of system transients using digital simulation, general systems subcommittee, ieee power engineering society," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 2599–2609, 2008. DOI: 10.1109/TPWRD.2008.923404.

[110] M. Tipping, "The relevance vector machine," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12, MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper/1999/file/f3144cefe89a60d6a1afaf7859c5076b-Paper.pdf.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was LuaLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi =scudo`. This class is available in every up-to-date and complete TEX-system installation.