POLITECNICO DI TORINO

Corso di Dottorato in Ingegneria Elettronica
e delle Comunicazioni

Tesi di Dottorato

Nonlinear Black-Box Models of Digital Integrated Circuits via System Identification

Claudio Siviero

Direttore del corso di dottorato

Prof. I.Montrosset

Tutore: Prof. F.Canavero

XIX Ciclo di Dottorato

Summary

This Thesis concerns the development of numerical macromodels of digital Integrated Circuits input/output buffers. Such models are of paramount importance for the system-level simulation required for the assessment of Signal Integrity and Electromagnetic Compatibility effects in high-performance electronic equipments via system-level simulations.

In order to obtain accurate and efficient macromodels, we concentrate on the black-box modeling approach, exploiting system identification methods. The present study contributes to the systematic discussion of the IC modeling process, in order to obtain macromodels that can overcome strengths and limitations of the methodologies presented so far. The performances of different parametric representations, as Sigmoidal Basis Functions (SBF) expansions, Echo State Networks (ESN) and Local Linear State-Space (LLSS) models are investigated. All representations have proven capabilities for the modeling of unknown nonlinear dynamic systems and are good candidates too be used for the modeling problem at hand. For each model representation, the most suitable estimation algorithm is considered and a systematic analysis is performed to highlight advantages and limitations. For this analysis, the modeling process is applied to a synthetic nonlinear device representative of IC ports, and designed to generate stiff responses.

The tests carried out show that LLSS models provide the best overall performance for the modeling of digital devices, even with strong nonlinear dynamics. LLSS models can be estimated by means of an efficient algorithm providing a unique solution. Local stability of models is preconditioned and verified a posteriori.

The effectiveness of the modeling process based on LLSS representations is verified by applying the proposed technique to the modeling of real devices involved in a realistic data communication link (an RF-to-Digital interface used in mobile phones). The obtained macromodels have been successfully used to predict both the functional signals and the power supply and ground fluctuations. Besides, they turn out to be very efficient, providing a significant simulation speed-up for the complete data link.

Acknowledgements

Turin, February 2007

I wish to thank my Ph.D. thesis advisor, Prof. Flavio Canavero for his advises and comments throughout the whole research activity and the preparation of this Thesis. I also wish to thanks Prof. Ivano Maio and Dr. Igor Stievano for supporting my work with continuos helpful suggestions and fruitful discussions. Their enthusiasm and scientific methodology have been a reference for me and their contributions improve the quality of the results.

Besides, I wish to thank Prof. José Carlos Pedro for his excellent guidance and Dr. Pedro Miguel Lavrador for his fruitful co-operation during my foreign experience within the TARGET exchange program at the Instituto de Telecomunicações, Aveiro, Portugal.

Finally, I would like to thank also Dr. Ali Nadir Arslan for his support during my research experience at the Nokia Research Center, Helsinki, Finland.

Table of contents

Chapter 1

Introduction

Present and future high-performance electronic equipments must satisfy higher and higher design requirements imposed by performance and technology constraints. The consequence for the designers is to perform a large number of Signal Integrity (SI) and ElectroMagnetic Compatibility (EMC) assessments. Such assessments are of paramount importance in order to detect and circumvent those sensitive effects like crosstalk, simultaneous switching noise, immunity and radiation [1] that may seriously compromise the achievement of the design objectives. This, in turn, implies an increase of the simulations of the complete system for the prediction of the signals propagation on the interconnects.

In order to perform such simulations, the combination of propagation effects with possibly very complex geometry and with the nonlinear behavior of the active devices makes a direct full-wave approach not feasible. Therefore, a feasible strategy must subdivide the propagation path into separate and well-defined sub-systems typically found along the signal propagation paths, *i.e.,* active devices, transmission-line interconnects, and interconnects with a complex 3D geometry such as vias and connectors. Each sub-system is separately characterized by a macromodel, *i.e.,* a set of equations that are able to reproduce with sufficient accuracy the electrical behavior of the sub-system. The macromodels are then implemented in a code suitable for commercial solvers, *e.g.,* SPICE-like, Analog-Mixed-Signal (AMS). Finally, the simulation of the complete system (*i.e.,* system-level simulation) as a chain of cascaded blocks is performed within the same environment. The

obtained voltage or current signals along the propagation paths or the eye-diagrams and possible other frequency domain figures like the Power Spectral Density (PSD) are computed and used by the designer to evaluate the system performance.

It is well known that such analysis is very challenging, since a broadband characterization of all the sub-systems must be taken into account. Besides, macromodels have to be efficient in order to require a limited amount of simulation time and CPU memory. Specifically, the numerical models representing the active devices play a key role, since they are shaping the signals on the system interconnections. For the linear interconnects (lumped and distributed) there are results and well-established methods already published in literature which lead to accurate and efficient macromodels [2, 3]. On the contrary, nowadays macromodeling of active devices is a challenging task that motivates this research activity. The traditional way for devices amounts to using macromodels based on the internal physical description. Those models are appropriate to system-level simulation and provide a good result accuracy. Unfortunately, the complexity of many devices (*e.g.,* huge number of variables, unknown nonlinear effects) as well as the lack of information on their internal structure often makes difficult the development of traditional physical models. Moreover, whether available, physical models disclose the Intellectual Property (IP) of the devices and, furthermore, they lead to high time-consuming simulations demanding a large amount of CPU memory.

In order to overcome the previous limitations, our attention is focused on the development of **Black-Box** *i.e.,* behavioral models via system identification methods [4]. The behavioral modeling of a system means to look for a mathematical relation among all the relevant external system variables on the basis of the external observation of the system response to suitable stimuli. No knowledge of the system internal structure is required. Application examples can be easily found in several other areas of interest. The control of industrial plants or complex mechanical systems as well as the prediction of economic phenomena demand the availability of such models. It is worth to remark that suitable strategies must be devised for modeling devices of different categories. In this Thesis, we concentrate on the macromodels of ports of digital Integrated Circuits (ICs), for which the representations based on nonlinear **parametric models** are selected. Such an approach has been

already successfully applied to ICs [2, 5, 6, 7, 8]. Nevertheless, the available results are rather preliminary and many relevant issues are still open. Mainly, model stability cannot be easily imposed a-priori or even during the estimation process without impacting on model accuracy. Furthermore, higher order dynamical effects may not be readily represented by these models and model estimation for real devices with multiple ports is troublesome and affects the quality of the estimated models. As an example, the generation of device port models including the effects of the neighboring ports suffers from the increase of complexity of the approximation problem.

Finally, it is worth to mention a complementary activity (not reported in this Thesis) developed during the triennium of studies. This activity has been related to the study of macromodels of Radio Frequency (RF) devices [9]. For this research topic, three months were spent at the Wireless Circuits & Systems Group of Instituto de Telecomunicações, Aveiro (Portugal) where a methodology for obtaining low-pass equivalent behavioral models of Power Amplifiers (PAs) has been developed. The results of such an activity are reported in [10].

The activity of this Thesis mainly contributes to the open issues of the modeling process for ICs. Specifically

- We investigate the performances of different parametric representation whose capabilities of modeling unknown nonlinear systems have been proven by the system identification theory. In particular, such representations have been applied for the first time to the IC modeling.
- For each model representation, suitable estimation algorithm are considered and their advantages and drawbacks are analyzed.
- Finally, the modeling stability issue is addressed.

Outline of the thesis

The structure of this thesis is as follows.

Chapter 2 deals with the general problem of IC ports modeling and outlines possible modeling approaches.

Chapter 3 deals with the approach we mainly discuss in this Thesis: the behavioral modeling via black-box identification. Here we describe the step-by-step for IC ports by means of parametric models.

Chapter 4 describes and compares possible parametric model representations suitable for modeling ICs, such as Sigmoidal Basis Functions (SBF) expansions, Echo State Networks (ESN) and Local Linear State-Space (LLSS) models.

Chapter 5 reports the systematic study of the performances of the proposed parametric representations to the modeling of a synthetic nonlinear dynamic one-port test device. From this test, the parametric representation providing the best overall performance is selected.

Chapter 6 discusses the impact of the proposed macromodel representation to the system-level simulation of data-link for mobile applications.

Finally, **Chapter 7** provides the main conclusions that can be drawn from the results of this Thesis.

Chapter 2

IC macromodels

2.1 Introduction

This Chapter outlines possible modeling approaches for the development of macromodels of ICs suitable for system-level simulation. As an example, Figure 2.1 shows the typical block diagram of a digital IC, mainly composed of the internal logic core and the input/output buffers driving and loading the interconnects themselves. Since digital ICs are complex systems, containing a very complex logic core and a high number of pins (several hundreds for modern microprocessors), there is no hope to effectively model both the IC internal logic and the input/output ports. For this reason and for the prediction of waveforms on interconnects, we require effective and accurate macromodels of digital IC ports. Since digital input/output buffers are nonlinear dynamic devices, we are interested in the development of macromodels described in the time-domain.

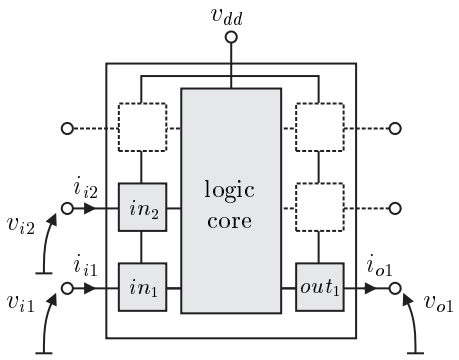

Figure 2.1. Block diagram of a generic digital IC.

In order to be effective for system-level simulation, IC macromodels must fulfill the following requirements

(i) Intellectual Property (IP) protection: macromodels should not disclose the information on the internal structure and technology of devices, in order to discourage any attempts of reverse engineering.

(ii) Accuracy: macromodels must provide responses in agreement with respect to the reference device , in particular for the prediction of higher order effects.

(iii) Efficiency: macromodels involved in system-level simulation have to consume as less simulation time as possible, as well as a limited amount of CPU memory.

(iv) Implementation in any commercial simulator: macromodels must be easily translated in code for each platform being used.

It is worth to remark that the development of macromodels meeting all the above mentioned requirements is a very challenging task. Basically, macromodels can be classified in two classes: physical models and behavioral models. Specifically, a physical model is based on the reconstruction of the internal structure of the device, whereas behavioral model is defined as a set of port characteristic equations (or the equivalent circuit of such equations) obtained from external (possibly virtual) measurements. In the next Sections we discuss the basic approaches belonging to the two methodologies, by highlighting their strengths and limitations.

2.2 Transistor-level models

The traditional way for IC ports modeling amounts to describe the device behavior by means of a detailed physical model based on internal structure. This is the so-called transistor-level description. As an example, Figure 2.2 shows the circuit of a typical 4-stages CMOS 1.2 μm output buffer, whose output terminals are (a) and (b) [11]. In this circuit, v_i denotes the buffer input voltage (*i.e.,* the output of the logic core of the IC), v and i are the buffer voltage and current at the output pin, respectively, and V_{cc} and V_{ss} indicate the power supply voltages.

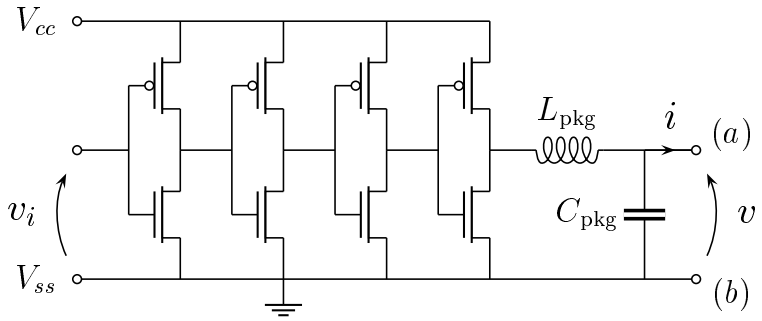

Figure 2.2. Example circuit for a typical 4-stages 1.2 μm CMOS output buffer [11].

This transistor-level model is the most accurate solution, and usually it is considered as the reference for the device. Unfortunately, the model completely discloses the internal details of the device, unless the developer releases an encrypted version of the transistor-level. In addition transistor-level descriptions are generally large in size and their effect is to slow down the simulation. Finally, they can not be easily plugged in any simulator, since they are usually written for a specific simulator in particular when the encryption holds.

2.3 Simplified equivalent circuits

In order to overcome the transistor-level limitations, behavioral modeling appears to be effective. The most common approach to behavioral modeling is via simplified equivalent circuits. A port model is defined by a suitable equivalent circuit, whose parameters can be estimated from input/output data. The information on the IC technology is used to devise the equivalent circuit, *i.e.,* the model structure. An important example of the equivalent circuit approach to behavioral modeling is the widely adopted Input/output Buffer Information Specification IBIS). IBIS is a set of rules defining and formatting data, from which IC port models based on simplified equivalent circuits can be developed [12]. IBIS offers high numerical efficiency, large data library and commercial software tools handling models and complex modeling problems. However, the equivalent circuit approach to behavioral modeling has also inherent limitations. Mainly the estimation of model parameters is easy only by virtual measurements, *i.e.,* from transistor-level models of the

devices, and the physical effects taken into account by the model are decided a priori, when the equivalent circuit defining the model is selected.

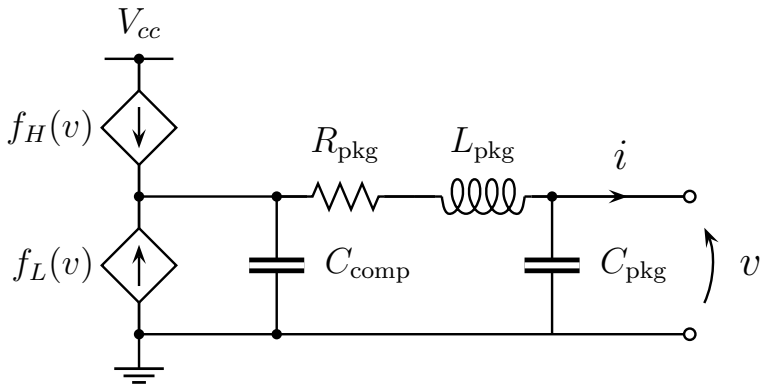

Figure 2.3. IC output port equivalent circuit assumed by IBIS.

As an example, Figure 2.3 shows the simplified equivalent circuit assumed by IBIS for the output port of a generic digital IC like the one shown in Figure 2.2. In this simplified circuit, the electric equivalent of the package is composed of the R_{pkg}, L_{pkg}, C_{pkg} elements and the silicon output port capacitance is assumed linear and modeled by C_{comp}. Finally, the voltage-controlled current source f_H and f_L account for both static characteristics when the port is driven either in LOW or HIGH output state and the dynamics during state switching. Data provided by IBIS, in accordance to the assumed equivalent circuit, must be translated into an executable model (IBIS model) in order to be used in circuit simulation environments.

2.4 Parametric models

A second possible approach to behavioral modeling, the one we mainly address in this Thesis, is via parametric models and system identification methods [4]. Such an approach amounts to the selection and estimation of a suitable nonlinear dynamic parametric model from the waveforms that can be measured at the IC ports. As an example, a general parametric model description of the IC output port in Figure 2.2 writes

$$i(t) = F\left(\boldsymbol{\Theta}\,,\, v(t)\,,\, \mathrm{d}/\mathrm{d}t\right) \tag{2.1}$$

where F is a suitable mathematical representation depending on the parameters collected in vector $\boldsymbol{\Theta}$.

In this approach, the modeled device is considered as a **Black-Box**, *i.e.,* in principle, no knowledge of the internal structure is required and the modeling information is completely contained in the device external responses. Owing to this feature, parametric models do not disclose any IP information and can be effectively estimated from measured transient responses or from simulated responses computed for detailed reference transistor-level models. Parametric models offer high accuracy and an acceptable numerical efficiency. Furthermore, since the model structure is selected by the estimation process itself, parametric models automatically take into account all the physical effects relating input and output data. In addition, they can be effectively translated in any commercial simulator. For such motivations, parametric models would enable any user to easily model sample devices and to simulate critical interconnect structure to assess sensitive SI/EMC effects.

2.5 Summary

Parametric modeling by means of Black-Box identification techniques appears to be appealing for the modeling of IC ports. This is also motivated by the system identification theory, that provides methodologies for developing effective parametric models of any unknown nonlinear systems. In fact, several results have been published on the identification of systems in the control automatic and mechanical area [4, 13]. Indeed, parametric modeling has been recently applied to ICs with good results [2, 5, 6, 7, 8]. Nevertheless, the modeling process still presents many open research issues. Some of them are addressed in the next Chapters.

Chapter 3

Parametric modeling

3.1 Introduction

In this Chapter we describe the step-by-step procedure for developing parametric model of ICs. For the sake of simplicity, in the following general description we focus on an IC port with external voltage and current v and i respectively, as shown in Figure 3.1 (the extension to multiport case is trivial and does not modify the results presented in the next sections).

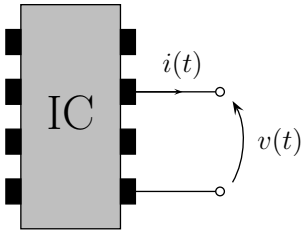

Figure 3.1. Example of IC port under modeling

The parametric modeling procedure of an IC port can be divided into five steps described in the following sections.

(1) Model selection: the starting point amounts to selecting the functional form of model equation, referred to a model representation.

(2) Identification signals: the port needs to be driven by specific signals in order to obtain transient voltage and current signals carrying information on its behavior. The excitation (input) and response (output) signals involved in this step are named identification signals. It is worth

to notice that the identification setup can be reproduced by using either the real device or its detailed transistor-level model.

(3) Model estimation: this step amounts to the numeric computation of the model parameters so that the model responses mimic well the identification signals.

(4) Model validation: once the model has been estimated, an assessment of the accuracy of the model to reproduce the responses to different excitations, is needed.

(5) Model implementation: finally, the translation of the obtained model in a standard circuit simulation environment is performed.

3.2 Model selection

The selection of the parametric model representation is the crucial point of the modeling process, since good models arise only when the model representation is suitable for the system being modeled. The model representation suitable for an IC port is searched within the class of discrete-time parametric models. This is mainly due to the large availability of resources for the estimation of this class of models [4]. Besides, this is the natural choice when the raw data, *i.e.,* the external responses of the IC port, are known as sampled waveforms. A very general and compact equation of a discrete time parametric model for the IC port in Figure 3.1 is

$$i(k) = F\left(\boldsymbol{\Theta}\,;\, v(k)\right) \tag{3.1}$$

where k is the discrete time variable, F is the nonlinear mapping describing the model and $\boldsymbol{\Theta}$ is the vector collecting the parameters. Equation (3.1) describes a unified framework to handle a large number of discrete-time parametric representations. In particular, system identification literature provides a large number of available representations for F that could be applied to the modeling of unknown nonlinear dynamic systems. Typical examples are neural networks based on radial, sigmoidal or spline basis functions, wavelet decomposition, kernel estimators, fuzzy models [13], support

vector machines [14], composite local-linear models [15], Wiener or Volterra polynomials [16], Wiener-Hammerstein models [17].

It is worth to remark that the model for IC ports can be also represented as a sum of a nonlinear static part and a nonlinear dynamic part [18], *e.g.,*

$$i(k) = F_s(v(k)) + F_d\left(\boldsymbol{\Theta}\,;\, v(k)\right) \tag{3.2}$$

where F_s and F_d are the possible static and dynamic parts of the model, respectively. The splitting of static and dynamic contributions is justified in Appendix A. The static part can be easily obtained from measurements, whereas parametric representation of the form (3.1) can be used for F_d. In the following, denoting **splitted** structures or models, we refer to equation (3.2), and using **fully nonlinear** terminology we refer to equation (3.1). It is of worth to note that the splitted representation turns out to have practical advantages in some applications, since it facilities the estimation of F_d [19] and may contribute to better models for some critical ports exhibiting strongly nonlinear dynamics.

3.3 Identification signals

Once the model representation F is chosen, the next step of the modeling process amounts to driving the IC port to obtain transient voltage and current signals carrying information on its behavior. The excitation (input) and response (output) signals involved in this step are named identification signals. As an example, Figure 3.2 shows the typical setup for collecting the identification signals for an IC port. The voltage source $v_s(t)$ is connected to the port via a resistor R_s in order to obtain a non-stiff transient test. In this case the recorded port voltage $\bar{v}(t)$ corresponds to the excitation and the recorded port current $\bar{i}(t)$ corresponds to the response.

As a general rule, the driving waveforms (input identification signals) must be carefully designed in order to excite every possible dynamic behavior of the port [4]. For linear systems, this is easily accomplished by using input identification signals with a frequency content that spans the frequency interval containing the system poles; generally white noise or pseudo-random binary signals are used. For the nonlinear case, unfortunately, only qualitative guidelines are available for the design of the input identification signals.

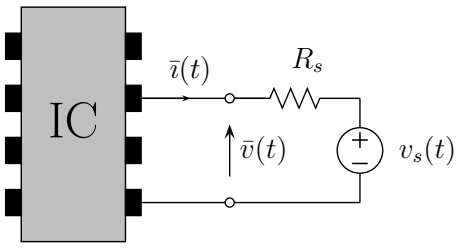

Figure 3.2. Typical identification setup for the IC port of Figure 3.1

Such signals should contain large steps with rise times short enough to excite the fast dynamic behaviors of the system and flat levels allowing the system to approach steady state operations on several operating points. A superimposed small noise signal usually improves the ability of such signals to excite the system dynamics. The final results are multilevel signals with superimposed small noise as the one reported in Figure 3.3.

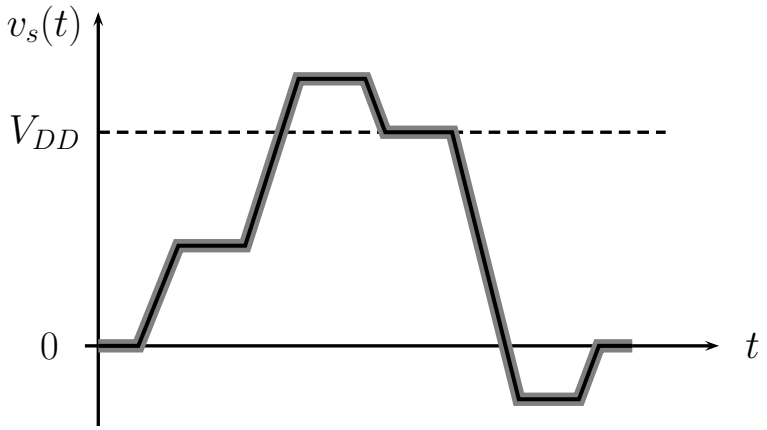

Figure 3.3. Example of a typical multilevel signals with superimposed noise used for v_s in Figure 3.2. In this case the voltage level V_{DD} is referred to the value of the power supply voltage for the IC under modeling.

Of course the rise times of the steps and the durations of the flat parts must be tuned on the fastest and slowest time constants that can be observed in the system responses, and, as a further rule of thumb, the number of different levels should increase, as the nonlinearity of the static characteristic becomes stronger. Again, the design of the input identification signals is a matter of repeated estimation experiments. Specific guidelines required for the generation of identification signals for the modeling of IC buffers are provided in [5]. In addition a systematic study of the effects of estimation

signals on the quality of IC buffer models is reported in [19].

3.4 Parameters estimation

The parameter estimation is obtained by means of standard methods, and amounts to fitting the response of the model to the identification signals recorded in the previous step. The simplest fitting approach is to look for $\boldsymbol{\Theta}$ minimizing the mean square error between the model and the port responses. This means to find

$$\boldsymbol{\Theta} \mid \min \left\{ \frac{1}{N} \sum_{k=1}^{N} \left(\, \bar{\imath}(k) - i(k) \, \right)^2 \right\} \tag{3.3}$$

where $\bar{\imath}(k) = \bar{\imath}(kT_s)$ is the sampled output identification signal, T_s is the sampling period, N is the total number of samples and $i(k)$ is the response of model (3.1) to the sampled input identification signal $\bar{v}(k)$. The sampled identification signals must contain all the information of the original identification signals. Therefore the sampling period T_s must be smaller than the sampling time T_N defined by the Nyquist frequency of the identification signals. On the other hand, the sampling time should not be too small, in order to avoid oversampling and consequent numerical problems in the minimization of (3.3). As a rule of thumb, the ratio T_N/T_s should be on the order of $2 \div 6$.

The minimization of (3.3) amounts to solving a nonlinear least squares problem. For this task, iterative algorithms are proven to be particularly effective. Nonlinear optimization literature provides specific algorithms depending on the choice of the representation F. Typical resources are gradient-based methods [20], genetic algorithms [21], extended Kalman algorithm [22], simulated annealing [23]. It is worth to remark that the problem (3.3) is nonconvex and the solution obtained by means of the iterative algorithm is a local minima. Thereby, a good initial estimate of the model is of paramount importance, since the initial starting point determines in which local minima the algorithm will end up.

3.5 Model validation

The estimation procedure previously described yields, along with the best values of the model parameters, the error of the model in reproducing the output identification signal. Such an error is the first indication of the model quality, because large errors at this stage imply the failure of the modeling process. However, the performances of such models must be further assessed by checking their response to an input signal different to the one used for the identification. In fact, models that reproduce well the identification signal may still exhibit strange behaviors when the input signal is changed. Specifically, model validation addresses the following issues: *(i)* accuracy *i.e.,* the error between the validation reference response and model response; *(ii)* efficiency *i.e.,* speed-up of the computation of the model response; *(iii)* stability *i.e.,* model response should not account for spurious dynamic behaviors (*e.g.,* oscillation, saturation) not present in the modeled device.

It is worth to remark that unstable models must be avoided, even if they reproduce the reference responses well. In fact, numerical simulation of these models for different signal and load conditions may lead to possible unstable behavior. When dealing with nonlinear model, stability analysis is not a trivial issue [24]. In particular the formulation of a general stability requirement is a challenging task. The application of a global stability criterion seems to be too complicate and likely too restrictive. On the other hand, a stability study based on a limited class of excitations and load conditions would be suitable. Nevertheless, this approach would not provide an exhaustive validation of the model. In order to devise a useful and feasible stability criterion, we study the local stability of the parametric model, as suggested in [25]. This methodology seems to be the simplest and it is readily extended from the linear case. Local stability analysis relies on the linearization of the model over each time step of the transient validation test. Each linearized model is then represented in the equivalent state-space form and the eigenvalues of the matrix mapping the states are computed (see more details in Appendix B). The parametric model is considered locally stable if the computed eigenvalues lie within the unitary circle for each time step. However, several test carried out have shown that the presence of some instants of time in which the eigenvalues lie outside the unitary circle does not significantly affect the

final performance of the model. The percentage of points of the validation set having at least one eigenvalue outside the unitary circle over the total number of points can be used as a confidence measure for the stability of the model. This ratio can be used as a criterion to assess the local stability of the obtained model. The smaller the ratio the smaller the probability that the model will become unstable during its operational evolution. Therefore model generation should aim obtaining models not showing any eigenvalue outside the unitary circle.

3.6 Macromodel implementation

The last part of the modeling process is the synthesis of the estimated discrete-time parametric model defined by (2.1) as an equivalent circuit to be implemented in standard simulation environments. For the sake of conciseness, and without loss of generality, we concentrate on the description of the following synthetic nonlinear discrete time parametric model

$$i(k) = a_1 i(k-1) + b_0 \mathrm{e}^{-c_0 v(k)} + b_1 v(k-1) \tag{3.4}$$

where a_1, b_0, c_0 and b_1 are the parameters. For the implementation of the above model as macromodel, it is useful to convert equation (3.4) into a continuous-time state-space realization [5], as described in the following. Equation (3.4) is rewritten into the discrete-time state-space form

$$\begin{cases} x_1(k) - x_1(k-1) &= a_1 x_1(k-1) + b_0 \mathrm{e}^{-c_0 v(k-1)} + b_1 x_2(k-1) + \\ & \quad -x_1(k-1) \\ x_2(k) - x_2(k-1) &= v(k) - x_2(k-1) \\ i(k) &= a_1 x_1(k) + b_0 \mathrm{e}^{-c_0 v(k)} + b_1 x_2(k) \end{cases} \tag{3.5}$$

where $x_1(k) = i(k-1)$ and $x_2(k) = v(k-1)$. The difference operator in (3.5) is then approximated with a differential one, (*e.g.*, $(\mathrm{d}/\mathrm{d}t)z(t) \cong (1/T_s)\,(z(k) - z(k-1))$). In this way, the time variable t is restored and the final equivalent continuous-time state-space representation arises

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \dfrac{1}{T_s} \begin{bmatrix} a_1 x_1(t) + b_0 \mathrm{e}^{-c_0 v(t)} + b_1 x_2(t) - x_1(t) \\ v(t) - x_2(t) \end{bmatrix} \\ \\ i(t) = a_1 x_1(t) + b_0 \mathrm{e}^{-c_0 v(t)} + b_1 x_2(t) \end{cases} \tag{3.6}$$

In order to use models expressed by (3.6) for the numerical system-level simulations, two practical choices are available: 1) convert the equations into circuit equivalents and exploit a SPICE circuit simulator [26]; 2) directly implement their expressions in analog mixed-signal (AMS) simulation environments, like Verilog-AMS [27] and VHDL-AMS [28, 29], that accept and solve differential-algebraic equations.

The conversion of differential-algebraic equations into circuit equivalents and their implementation as SPICE subcircuits is a standard procedure [5]. To do this, the first two rows of (3.6) can be implemented by simple equivalent circuits with voltage controlled sources and the third one by a current controlled source only. As an example, Figure 3.4 shows the circuit synthesis of the second equation of (3.6). The circuit synthesis of the first equation is obtained by properly replacing the controlled source. The complete equivalent circuit of (3.6) can be easily coded as a SPICE-like subcircuit, as shown in Figure 3.5.

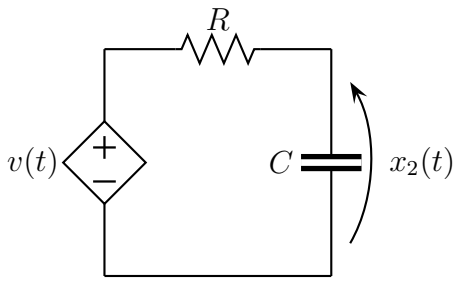

Figure 3.4. RC equivalent circuit for $\frac{\mathrm{d}}{\mathrm{d}t}x_2(t) = \frac{1}{T_s}\left[v(t) - x_2(t)\right]$, $T_s = RC$.

3.7 Open issues

Nowadays, the parametric modeling process described in this Chapter presents several issues that have not been fully investigated yet. In particular, referring to the five steps outlined in the Introduction of this Chapter, the following issues need consideration.

(1) Model selection: properties, advantages/disadvantages and applicability range of a comparative analysis of different parametric representation need to be investigated; in particular, their suitability for IC modeling has still to be determined.

```
.subckt macromodel v ref
+ PARAMS:
* sampling time Ts=Rx*C (Rx=1, C=Ts)
+ Rx = 1
+ Ts = ...
* model parameters
+ a1 = ... b0 = ... b1 = ... c0 = ...

* dx1/dt={y-x1/Ts}
  Cx1 x1 0   {Ts}
  R1  x1 z1  {Rx}
  Ex1 z1 0   value={V(y)}

* dx2/dt={v-x2/Ts}
  Cx2 x2 0   {Ts}
  R2  x2 z2  {Rx}
  Ex2 z2 0   value={V(v,ref)}

* output controlled current source i(t)
  Gy v ref   value={V(y)}

* model representation/structure
  EF  y 0    value={a1*V(x1)+b0*EXP(-1*c0*V(v,ref))+b1*V(x2)}
  RF  y 0    {Rx}

.ends
```

Figure 3.5. SPICE implementation of the parametric model (3.4), via the representation (3.6).

(3) Model estimation: for each representation, estimation algorithms and the assessment of their performances require further investigations.

(4) Model validation: a thorough stability analysis of models is required.

No extra efforts are planned for step (2) because, so far, multilevel signals seem to be appropriate for the IC modeling. However, a study of possible improvements provided by excitations of different shape can surely be an interesting topic for future work. Finally, also step (5) does not need

further consideration since the macromodel implementation is essentially a well-established technicality.

The aim of the following Chapters is the development of studies and contributions related the outlined open issues.

Chapter 4

Model representations

4.1 Introduction

In this Section we investigate possible parametric model representations for the modeling of ICs by means of the process described in Chapter 3. For the sake of simplicity, our discussion deals with the approximation of the constitutive relations of the IC port described by an external voltage v and a current i, as shown in Figure 3.1. Even if unknown, the constitutive relation of the IC port can be described by an arbitrary continuous-time state-space representation involving external measurable variables and internal nonmeasurable state variables, as follows

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{g}\left(\mathbf{x}(t)\,, v(t)\right) \\ i(t) &= f\left(\mathbf{x}(t)\,, v(t)\right) \end{cases} \tag{4.1}$$

where $\mathbf{x}$ is the vector of internal state variables, and $\mathbf{g}$ and f are multivariate nonlinear mappings.

For the approximation of the input-output behavior of (4.1), the identification literature provides many parametric model representations [13, 14, 16, 17, 30]. Such models show the so-called universal approximation capability, *i.e.,* for a sufficiently large size of the model an arbitrarily small modeling error can be achieved. According to Section 3.2, the typical parametric to be considered are discrete-time relations. Specifically, such models can be represented by a Nonlinear Input-Output (NIO) description or alternatively by a Nonlinear State-Space (NSS) description. Both classes have strengths

and limitations, as well as the methods available in literature for the estimation of their parameters. For the modeling problem at hand, it is not clear what the real benefits of using NIO rather than NSS models are. Therefore, a systematic study of the two classes is needed.

In this Chapter, the two representations are introduced and discussed in details. In particular, the next sections present the mathematical structure of the models, their general strengths (well known from the literature and from the application of these models to real problems) and the available methods for parameter estimation. All the models considered are good candidates for the IC modeling. It is worth to remark that in the following section we focus on widespread model representations available in literature, that have been proven to provide good results in applications, other than the one addressed in this Thesis. A systematic comparative study for the selection of the best model representation is deferred to Chapters 5 and 6.

4.2 Classification: NIO and NSS models

In this section we present and discuss the features of the NIO and NSS parametric model classes.

Most single-input single-output NIO parametric models can be written as [13]

$$\begin{cases} i(k) &= F\left(\, \boldsymbol{\Theta}\, ;\, \varphi(k)\, \right) \\ \varphi(k) &= \left[\, i(k-1)\, ,\, \ldots\, ,\, i(k-r)\, ,\, v(k)\, ,\, \ldots\, ,\, v(k-r)\, \right]^{\mathrm{T}} \end{cases} \tag{4.2}$$

where k refers to the discrete-time variable, $i(k)$ is the output sequence of the model, $v(k)$ is the input sequence, F is a nonlinear mapping defining the model representation, and $\boldsymbol{\Theta}$ is the vector of model parameters. Vector $\varphi(k)$ is named the vector of regressors collecting the past r samples of the output and the present and past r samples of the input, r being the dynamic order of the model. As outlined in [13], equation (4.2) provides a unified framework to handle models from both system identification area and from other areas like neural networks, wavelets and fuzzy systems.

On the other hand, a generic single-input single-output NSS parametric

model writes

$$\begin{cases} \mathbf{z}(k) &= \mathbf{F}_1 \left(\boldsymbol{\Theta}_1 \, ; \, \mathbf{z}(k-1) \, , v(k-1) \right) \\ i(k) &= F_2 \left(\boldsymbol{\Theta}_2 \, ; \, \mathbf{z}(k) \, , v(k) \right) \end{cases} \tag{4.3}$$

where $i(k)$ and $v(k)$ are the output and input sequence, and

$$\mathbf{z}(k) = [z_1(k) \, , \, \ldots \, , z_n(k)]^{\mathrm{T}} \tag{4.4}$$

is the virtual state vector. The state vector (4.4) is not necessarily an estimate of the real internal state $\mathbf{x}$ of (4.1), even if ideally it could be but, a priori, $\mathbf{z}$ and $\mathbf{x}$ are not even correlated. Model (4.3) is defined by the multivariate nonlinear mappings $\mathbf{F}_1$ and F_2 depending on the vectors of parameter $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$ respectively.

Although the model representations (4.2) and (4.3) can be related to each other [30], the two classes exhibit different features.

NIO models have been successfully applied to real modeling of nonlinear systems with nonmeasurable states in the area of automatic controls. They involve only the input and output measurable variables and reduce the dependence on the nonmeasurable states to a direct dependency on the output dynamics. Furthermore, the estimation of such a class of models can be done by means of well-established methods. All these motivations have driven the interest in NIO models as good candidates for IC modeling. NIO models have been recently proven to accurately reproduce the behavior of a wide class of commercial devices [5, 6, 7]. In particular, they turn out to be very compact, thus leading to models with a very small size. Owing to this, the estimated models, implemented in a simulation environment, are very efficient and allow simulation speed-ups on the order of $10 \div 1000$ with respect to the transistor-level descriptions of devices. Nonetheless, NIO relations have shown inherent limitations. Mainly: *(i)* model estimation for real devices with multiple ports is troublesome and impacts on the quality of estimated models; *(ii)* higher order dynamical effects may not be readily represented by these models; *(iii)* model accuracy depends on the initial guess of parameters and on local minima of the cost function; *(iv)* local stability of models can not be easily imposed *a-priori* or even during the training process without impacting on model accuracy. It is worth to remark that, according to Section 3.5, locally unstable models must be avoided, even if

they well reproduce the reference responses used in the model estimation. In order to address the previous limitations, NSS models appear to be a useful alternative. A state-space model is the most natural representation of a dynamic system. Besides, NSS models have been widely used in the area of automatic controls and system theory to model actual nonlinear systems such as industrial plants or power electric machines. The specificity of NSS modeling arises from the fact that the internal state variables gives more flexibility with respect to NIO models. However, NSS models have a complicate structure due to the presence of a larger number of degrees of freedom. Moreover, the related estimation methods are still under study.

As outlined in Section 4.1, in this activity we concentrate on different example representations belonging to both NIO and NSS classes. All the considered representations have potential strengths and need to be further investigated for our problem. In particular, the representations considered in the systematic study carried out in this thesis are the following

(i) Representations of the NIO class. A general way to define nonlinear mappings F of NIO models (4.2) is to exploit sums of nonlinear functions of regressors [13]. Many different basis functions can be used, giving rise to model representations with significantly different properties. In the PhD thesis [8], macromodels based on Gaussian Radial Basis Function (RBF) expansions [13] have been studied and successfully applied to the macromodeling of the ports of digital ICs [5, 6]. Such models offer remarkable advantages. Mainly, they are robust and have a regular and smooth behavior outside the fitting domain and the estimation of model parameters relies on simple and efficient algorithms. However, for the problem at hand, macromodels based on Sigmoidal Basis Functions (SBF) expansion turned out to be more effective [7]. SBF models show properties that are more suitable for fitting the actual constitutive relations of IC ports and usually lead to simpler (more efficient) macromodels than those based on RBF. Furthermore, the algorithms themselves for the estimation of SBF models, even if require more complex and fully nonlinear procedures, allow for more accurate estimates. This points to exploiting only the SBF models within the NIO class; Section 4.3 provides further details on SBF models.

(ii) Representations of the NSS class. Discrete-time NSS models have been recently studied in literature for the identification of nonlinear dynamic systems and only preliminary results are available [17, 25, 30, 31, 32, 33, 34, 35]. Within the available options, in this study we concentrate on two representations that seem to be attractive: the Echo State Networks (ESN) [34, 36] and the Local-Linear State-Space (LLSS) models [35, 37].

4.3 Sigmoidal Basis Functions (SBF)

Models based on SBF describe F in equation (4.2) as an expansion of sigmoidal functions. The general input-output relation of a SBF model writes

$$i(k) = \sum_{j=1}^{p} \alpha_j \Phi \left(\mathbf{a}_j^T \varphi(k) + b_j \right) \tag{4.5}$$

where Φ is the sigmoidal mother function. In this study $\Phi = \tanh$. In equation (4.5), p is the number of basis functions (size of the model) α_j are linear coefficients, and $\mathbf{a}_j$ and b_j are the nonlinear parameters of the sigmoidal function. In a different and equivalent perspective, the above model is a recurrent neural network with one hidden layer and a linear output unit [13, 38, 39].

The estimation of the parameters of model (4.5) requires the solution of the nonlinear problem defined by (3.3). Within the many possible estimation algorithms we found good results with the well-known Levenberg-Marquardt (LM) based methods [40] in conjunction with the pseudo-random procedure for the selection of the initial guess of parameters, as suggested in [41]. Specifically, we consider two different implementations of the above algorithm, referring to the two schemes most diffused in literature for the estimation of NIO models. The first version is called Static LM (SLM) [42]. This procedure refers to the most common scheme for the estimation of static feedforward neural networks [38]. In fact, this method solves a nonlinear static problem *i.e.,* the past samples of the model output accounted in $\varphi(k)$ are forced to be the samples of the output identification sequence. On the contrary, the second version called Recurrent LM (RLM) [43] solves the problem considering the recurrent nature of the model *i.e.,* $\varphi(k)$ accounts for the feedback of

the model output. The mathematical details of both versions applied to NIO structures are reported in Appendix C. It is worth to notice that neither SLM nor RLM version of the algorithm address the model stability issue. Therefore, SBF model stability can be only verified a-posteriori without any guarantee of obtaining a local stable model.

The starting point of the estimation procedure for SBF models is to define completely the model representation by setting the size parameters *i.e.,* the dynamic order r and the number of basis functions p. The dynamic order is rather a property of the device under modeling and can be determined a priori from the device responses [44] or simply postulated and verified a posteriori. In order to decide the most suitable size p, each estimation run consists of a comparison of models with increasing p value (*e.g.,* $1 \div 10$), with respect to the estimation accuracy. As a rule of thumb, good p values are the smallest ones leading to a good reproduction of the identification signals. Suitable statistical indexes help the selection of p [4].

4.4 Echo State Networks (ESN)

This Section briefly discusses the main features of ESN modeling. This class of models has been recently presented in the literature and provides very good results for the modeling of the complex dynamic behavior of real systems [34, 36, 45]. A comprehensive discussion of ESN can be found in [34] and references therein. This model belongs to the discrete-time NSS class. As an example, for the system in (4.1), an ESN parametric representation writes

$$\begin{cases} \mathbf{z}(k) &= \mathbf{F}_1\left(\mathbf{A}\mathbf{z}(k-1) + \mathbf{b}v(k-1)\right) \\ i(k) &= \mathbf{c}^{\mathrm{T}}\mathbf{z}(k) + c_0 v(k) \end{cases} \tag{4.6}$$

where vector $\mathbf{z} = [z_1,...,z_n]^{\mathrm{T}}$ collects the n internal state variables. Typical values of n are in the range [20, 500] and the nonlinear multivariate mapping $\mathbf{F}_1$ consists of the collection of sigmoidal functions. In this study, $\mathbf{F}_1 = [\,\tanh\,,\,\ldots\,,\,\tanh\,]^{\mathrm{T}}$. The parameters of the model are the square matrix $\mathbf{A}$, vectors $\mathbf{b}$ and $\mathbf{c}$, and the scalar c_0.

It is worth noting that the estimation of (4.6) would in principle require the solution of a nonlinear optimization problem. This limitation, along with the large number of unknown parameters involved in a state-space equation,

would limit this approach for the identification of complex nonlinear dynamical systems. In order to address the previous limitations, the methodology proposed in [34] amounts to adapt only the weights of the network-to-output connections, *i.e.,* the vector $\mathbf{c}$ and the scalar c_0, yet leading to the solution of a linear least squares problem. The other parameters, *i.e.,* matrix $\mathbf{A}$ and vector $\mathbf{b}$, are defined *a-priori* in a way that allows the inclusion in the model of a large number of randomly generated dynamics, possibly including those of the original system under modeling. Matrix $\mathbf{A}$ is chosen to satisfy the "echo state" property, which means that for each input sequence, model (4.6) must present a unique sequence of state variables. In order to fulfill the "echo state" requirement, the square matrix $\mathbf{A}$ is chosen to have a spectral radius ρ (*i.e.,* the largest magnitude of the eigenvalues) smaller than one (e.g., $\rho = 0.7 \div 0.99$). It can be easily proven that the above assumption, along with the choice of the tanh function in the nonlinear mapping $\mathbf{F}_1$, leads to locally stable ESN models. Within the many possible choices for $\mathbf{A}$, that have not been completely investigated yet, a possible solution is proposed in [34]. In the above paper, matrix $\mathbf{A}$ is generated as a sparsely and randomly connected matrix, whose elements are independent random variables that have a certain (high) probability to be zero (*e.g.,* 95%), and the complementary probability, labeled as connectivity (*e.g.,* 5%), to be $\pm a$, where a is real positive number. The number a is then computed to enforce the spectral radius ρ to the desired value. This choice satisfies a sufficient and hence restrictive condition for the "echo state" property. Better solutions allowing a higher number of degrees of freedom exist and are currently under investigation [46]. The details on the criteria used for the initialization of the model and for the estimation algorithm are reported in [34]. Basically, the only size parameter to be tuned during the estimation, is the size n of the state vector. Roughly speaking, a suitable value of n can be selected by performing several model estimation runs with increasing n (*e.g.,* $n = 20\ ,\ 50\ ,\ 100\ ,\ \ldots$) and then selecting the lowest value of n providing a sufficient estimation accuracy.

4.5 Local Linear State-Space (LLSS)

The idea underlying the LLSS modeling methodology is the approximation of the complex dynamic behavior of a nonlinear dynamic system by means

of the composition of local linear models [15]. The whole operating range of the system is partitioned into smaller operating regions where the system behavior is approximated by a linear state-space equation. Even if this idea has been already investigated in the literature, the implementation presented in [35, 37] has several strengths, including the nice feature of providing the automatic computation of local linear models as well as the generation of the weights for the local models from input-output system responses only. Furthermore, the special structure of LLSS models facilitates the estimation procedure. As an example, for the system (4.1) a LLSS model is defined by the following discrete-time NSS representation

$$\begin{cases} \mathbf{z}(k) &= \displaystyle\sum_{j=1}^{p} \rho_j\left(\mathbf{s}(k-1)\right)\left(\,\mathbf{A}_j\mathbf{z}(k-1) + \mathbf{b}_j v(k-1) + \mathbf{o}_j\,\right) \\ i(k) &= \displaystyle\sum_{j=1}^{p} \rho_j\left(\mathbf{s}(k)\right)\left(\,\mathbf{c}_j^{\mathrm{T}}\mathbf{z}(k) + d_j v(k) + q_j\,\right) \end{cases} \tag{4.7}$$

where p is the number of local-linear models, vector $\mathbf{z} = [z_1\,,\,\ldots\,,z_n]^{\mathrm{T}}$ collects the n internal states and $\rho_j(\cdot)$ is the weighting coefficient of the j-th local model. Each local model is defined by the state matrix $\mathbf{A}_j$ and by vectors $\mathbf{b}_j$, $\mathbf{o}_j$ and $\mathbf{c}_j$ and by scalars d_j and q_j. The argument of the weights, *i.e.,* the scheduling vector $\mathbf{s}(k)$, corresponds to the operating point of the system and is in general a function of both input and state variables. Among the possible choices for $\mathbf{s}(k)$, a common solution in local linear modeling (also used in [35, 37]) amounts to collecting the present and past samples of the input sequence $v(k)$ only. This writes

$$\mathbf{s}(k) = [\,v(k)\,,v(k-1)\,,\,\ldots\,,v(k-r)\,]^{\mathrm{T}} \tag{4.8}$$

where r refers to the chosen number of past input variable samples.

It is common practice in local linear modeling to use normalized radial basis functions for the weights $\rho_j(\mathbf{s}(k))$ *i.e.,* each weight varies between zero and one and their sum is forced to be one at each operation point of the system. Therefore the j-th weight writes

$$\rho_j(\mathbf{s}(k)) = \frac{\phi_j(\mathbf{s}(k))}{\displaystyle\sum_{i=1}^{p}\phi_i(\mathbf{s}(k))} \tag{4.9}$$

where $\phi_j(\cdot)$ is the j-th radial basis function defined as

$$\phi_j(\mathbf{s}(k)) = \exp\left(-\frac{\|\mathbf{s}(k) - \mathbf{t}_j\|^2}{\beta_j^2}\right) \tag{4.10}$$

Each radial basis function (4.10) is defined by its position in the space of the scheduling vector (center $\mathbf{t}_j$) and by its spreading (scale parameter β_j).

For the computation of model parameters, *i.e.,* the local model parameters in (4.7) and the parameters defining the weights in (4.10), we perform the solution of the nonlinear optimization problem by applying the LM method for NSS structures (see details in Appendix C). In particular, the LM version proposed in [35, 37] is called Projected LM (PLM). The basic version of the LM algorithm has been suitably modified to handle the non-uniqueness of a state-space representation that may cause ill-conditioning of matrices during model estimation. In addition, parameter initialization is carried out by means of a deterministic procedure, thus avoiding the dependence of the estimated model to the initial guess of parameters. Roughly speaking, the gradient direction search in the PLM algorithm is modified to avoid the directions in the parameter space that do not change the cost function due to a similarity transformation of model matrices (see details in [35, 37]). The initial guess of the parameters defining the local models are set equal to the matrices of a single global stable linear model. The parameters of the global linear model are computed by means of the application of an efficient subspace identification method of the 4SID class [47]. The latter subspace method also provides the automatic computation of the number n of internal state variables, i.e., the size of vector $\mathbf{z}$ in (4.7) and the initial condition of the state variables. Besides, the initial radial weighting functions $\rho_j(\cdot)$ are distributed uniformly over the range of the input sequence. Using linear models for initialization is a nice feature that has been proposed and motivated in [48]. It is worth to remark that the initialization by means of 4SID method provides a bounded-input, bounded-output stable initial model. As pointed out in [35, 37], the proposed PLM algorithm likely leads to the identification of models that are locally linear stable. This is due to the choice of the normalized weights along with the form of the scheduling vector accounting for present and past samples of the input sequence only. Nevertheless, no additional constraints are included to enforce stable models during training and local stability can only be verified a posteriori.

The number of local linear models p and the number of input past samples accounted by the scheduling vector r must be selected before the estimation. Again, we can choose those values by comparing models with increasing p and r values, with respect to the estimation accuracy. Obviously, good p and r values are the smallest ones leading to a good reproduction of the identification signals.

4.6 Summary

In this section, we briefly compare the general strengths and limitations of the model representation described throughout this Chapter. Table 4.1 summarizes the performances related to the estimation time, model size and stability for the three different models under consideration. An up-arrow "↑" means a positive feature, conversely a down-arrow "↓" means a negative feature.

model	estimation time	model size	local stability
SBF	↓	↑↑	↓ (a-posteriori)
ESN	↑	↓	↑ (a-priori)
LLSS	↓	↑	↑ (a-posteriori)

Table 4.1. Comparison of the features of the model representations considered.

From the figures in Table 4.1 it is clear that we cannot choose a-priori the best candidate for model representation. A systematic study of the model performances for IC ports must be carried out. This is done in Chapter 5.

Chapter 5

Assessment of models

5.1 Introduction

In this section, we assess the performances of the model representations described in Chapter 4 by applying the modeling process proposed in Chapter 3 to the port voltage-current relation of a synthetic nonlinear dynamic one-port test device relation. The reference model is the ODE (Ordinary Differential Equations) description of the test device driven by a real voltage source. The reference model is used to compute the responses needed for the model identification and for model validation. At the end of the analysis, we select the model representation that is most feasible for the modeling of IC ports.

5.2 Test device

This section describes and discusses the structure of synthetic nonlinear dynamic one-port test device being modeled. Figure 5.1 shows the circuit schematic of the test device, where the external port voltage and current are v and i respectively. This device is designed in order to be representative of the IC input/output buffers such as drivers and receivers. It is composed by a cascade connection between the two-port element representing a realistic package of an IC and the one-port element representing the nonlinear functional part of the active device. The common port connecting the package and the nonlinear functional part is identified by the voltage v_1 and current i_1. The package is modeled by a lumped network of elements

R_{pkg}, L_{pkg} and C_{pkg}, whose values are those tabulated for the standard package TSSOP48. The nonlinear functional part is mainly characterized by the voltage-controlled current source $f_1(v_1)$, which defines the type of device operation. When $f_1 = 0$, the test device addresses a receiver input port. On the contrary, when $f_1 \neq 0$ the test device represents the output port of a driver in a fixed logic state with f_1 as the port static characteristic. In either receiver or driver configuration, the voltage-controlled current source $f_2(v_2)$ attempts to mimic the typical mechanism of ESD (ElectroStatic Discharge) protection of silicon devices. In particular, f_2 is described by a nonlinear static characteristic that works decreasing the output current i_1 whenever the port voltage v_1 is larger than a threshold set by the battery V_{DD}. Finally, the capacitor C represents the equivalent port capacitance of the functional part and R and L account for the bonding wire link between V_{DD} and f_2.

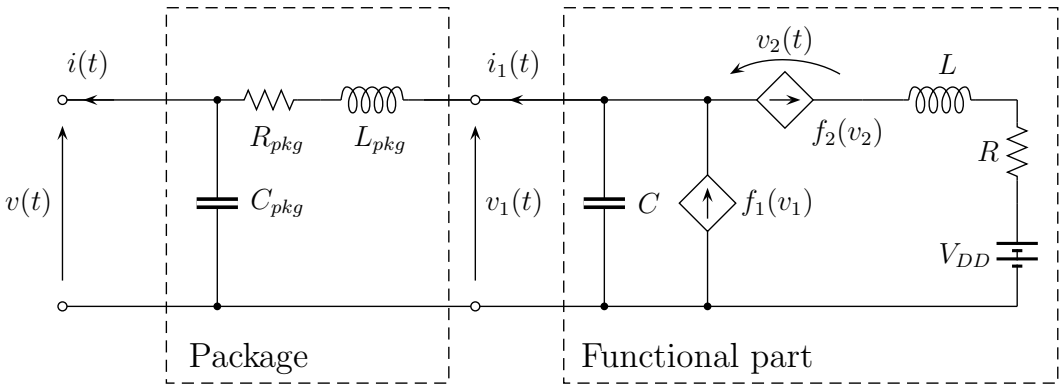

Figure 5.1. Schematic of the one-port test device under modeling.

For the design of the device in Figure 5.1, our goal is to set the circuit elements in a way to obtain a stiff modeling benchmark. In particular, we carefully design the nonlinearity of f_2 and we choose the value of L, since such parameters mainly influence the nonlinear dynamic behavior of the device. We consider the case of driver operation in a fixed logic state with a voltage threshold for the activity of f_2 fixed by $V_{DD} = 1\,\mathrm{V}$. The characteristic of the voltage-controlled current sources f_1 and f_2 are chosen in order to be representative for the typical nonlinearities of silicon devices. In particular, for the characteristic of f_1 we select the parametric equation

$$f_1(v_1) = a_1 - a_2 \mathrm{e}^{-a_3 v_1} - a_4 v_1 \tag{5.1}$$

whereas for the characteristic of f_2 we select the parametric equation

$$f_2(v_2) = b_1 \mathrm{e}^{b_2(v_2 - V_{DD})} \tag{5.2}$$

The tuning of parameters a_1, a_2, a_3 and a_4 in equation (5.1) is done in order to fit the typical current range of the most common technologies of digital IC drivers, with respect to the chosen value of V_{DD}. Conversely, the parameters b_1 and b_2 of equation (5.2) are devised to observe a pronounced activity of f_2 whenever the voltage v_2 is larger than zero. The final shape of the characteristics (5.1) and (5.2) are reported in Figure 5.2.

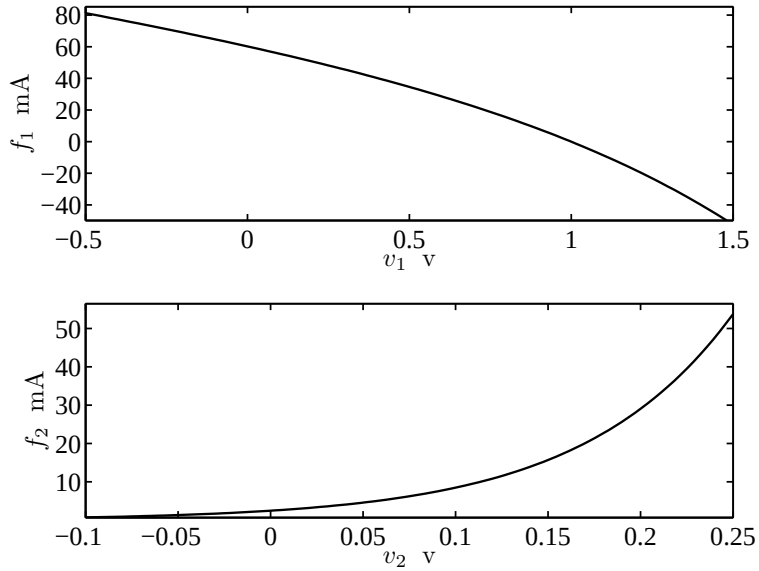

Figure 5.2. Characteristics of the voltage-controlled current sources of the one-port test device in Fig. 5.1: $f_1(v_1)$ described by the equation (5.1) (top panel) and $f_2(v_2)$ described by the equation (5.2) (bottom panel).

The tuning of the value of inductance L is performed in order to observe a different dynamic behavior between the situation in which f_2 provides a significant amount of current and the situation in which f_2 provides a negligible amount of current. Table 5.1 groups the values of the parameters in equations (5.1) and (5.2) and of the passive components in the circuit of Figure 5.1.

Element	Value
a_1	70 mA
a_2	10 mA
a_3	1.35 V^{-1}
a_4	30 mΩ^{-1}
b_1	550 A
b_2	12 V^{-1}
R_{pkg}	0.5 Ω
L_{pkg}	3 nH
C_{pkg}	0.1 pF
R	0.1 Ω
C	5 pF
L	1.5 nH

Table 5.1. Values of the parameters in equations (5.1) and (5.2) and of the passive components composing the test device of Figure 5.1.

5.3 Modeling setup

In this Section we briefly review the application of the modeling process described in Chapter 3 to the test device in Figure 5.1, by means of parametric models. In particular for each step, we highlight the main elements and features impacting on the performances of the obtained model. The next subsections address the following steps of the process

(1) Model selection

(2) Identification signals

(3) Model estimation

(4) Model validation

Step (5), related to the model implementation in a circuit simulator, is skipped since it is not relevant for the application at hand.

5.3.1 Model selection

As outlined in Section 3.2, the first step concerns the selection of model for the test device. Among all possible choices, we select three main classes (see

Chapter 4 for details)

(i) Sigmoidal Basis Functions (SBF)

(ii) Echo State Networks (ESN)

(iii) Local Linear State-Space (LLSS)

For each representation, the model structure must be selected according to the two different choices presented in Section 3.2 (fully nonlinear or splitted, details in Appendix A).

5.3.2 Identification signals

In this step, the collection of the identification signals is addressed. This is done by applying a real voltage source to the one-port test device in Fig. 5.1 and by recording the corresponding port voltage and current by means of a transient simulation. Such an experiment is described by the test setup in Figure 5.3 where the port of the test device is driven by an excitation composed of a voltage source v_s and a series-resistor R_s.

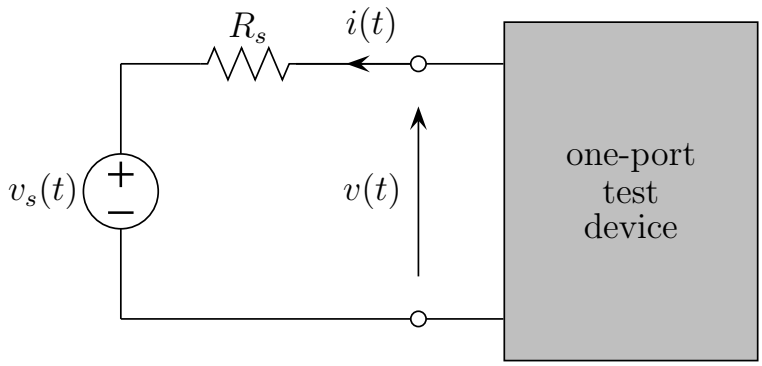

Figure 5.3. Modeling setup for the one-port test device in Fig. 5.1.

For achieving the transient simulation, the continuous-time ODE description of the test setup in Figure 5.3 is selected as the reference model. Such an ODE description is formulated in the canonical form

$$\left\{ \begin{array}{rcl} \mathbf{M}\,(\mathbf{x})\,\dot{\mathbf{x}} & = & \mathbf{h}\,(\mathbf{x},\mathbf{u}) \\ y & = & \mathbf{c}^{\mathrm{T}}\mathbf{x} + \mathbf{d}^{\mathrm{T}}\mathbf{u} \end{array} \right. \tag{5.3}$$

where y is the output variable corresponding to the port current i, $\mathbf{x} = [\, v\,,\, i_1\,,\, v_1\,,\, v_2\,]^{\mathrm{T}}$ is the vector collecting the state variables, and $\mathbf{u} = [\, v_s\,,\, V_{DD}\,]^{\mathrm{T}}$ is the input vector accounting for the time-varying and fixed voltage sources. The first row of the formulation (5.3) is a nonlinear differential equation describing the state evolution. This equation involves the nonlinear state-dependent so-called mass matrix

$$\mathbf{M}\,(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \dfrac{\mathrm{d}f_2(v_2)}{\mathrm{d}v_2} \end{bmatrix} \tag{5.4}$$

and the nonlinear functional depending on the state and input vectors

$$\mathbf{h}\,(\mathbf{x},\mathbf{u}) = \begin{bmatrix} \dfrac{1}{C_{pkg}}\left(-\dfrac{v}{R_s} + i_1 + \dfrac{v_s}{R_s}\right) \\ \dfrac{1}{L_{pkg}}\left(-v - R_{pkg}i_1 + v_1\right) \\ \dfrac{1}{C}\left(f_1(v_1) - f_2(v_2) - i_1\right) \\ \dfrac{1}{L}\left(v_1 - v_2 - V_{DD} - Rf_2(v_2)\right) \end{bmatrix} \tag{5.5}$$

The second row of the formulation (5.3) is a linear algebraic equation describing the output variable from the state and input vectors by means of the vectors $\mathbf{c} = [\, 1/R_s\,,\, 0\,,\, 0\,,\, 0\,]^{\mathrm{T}}$ and $\mathbf{d} = [\, -1/R_s\,,\, 0\,]^{\mathrm{T}}$. The identification signals v and i are obtained by simulating the test setup of Figure 5.3 by solving an initial value problem for the ODE formulation (5.3). The solution is numerically calculated by means of Matlab ODE functions [49].

According to the guidelines outlined in Section 3.3, the driving voltage source v_s is a multilevel signal with superimposed small noise. Such a signal must be devised in order to obtain identification signals rich of information on the device behavior, *i.e.,* the signal v_s must excite the dynamic behavior of the device for values within all possible operating voltages, even extending outside the power supply rails. For this test, we select a series-resistor $R_s = 50\,\Omega$ and a voltage source v_s composed of thirty levels spanning the

range of operating voltage $[0\,\mathrm{V}\,, V_{DD} + \Delta]$, where $\Delta = 0.5\,\mathrm{V}$ is the accepted overvoltage. As outlined in Sec. 3.3, the design of a multilevel stimulus required by the identification of nonlinear dynamical systems is a matter or repeated experiments. For the modeling of digital devices we performed a systematic set of experiments which confirms that the quality of the estimated models is weakly sensitive to the parameters defining the multilevel signals [19]. As an example, a number of levels within the range between five to some tens is sufficient for leading to accurate models reproducing the original system response well. The superimposed noise is of gaussian type with a selected standard-deviation of 0.1 mV. The flat parts of v_s last 6 ns, *i.e.,* a sufficient duration to allow the port to reach steady state operation. The duration of level transition is set to 200 ps, *i.e.,* a typical value for the switching time of modern high-speed devices. The sampling period used to discretize the signals is $T_s = 10\,\mathrm{ps}$. Figure 5.4 shows the multilevel voltage sources previously described and the corresponding identification signals as result of the numerical transient simulation of the test setup of Figure 5.3.

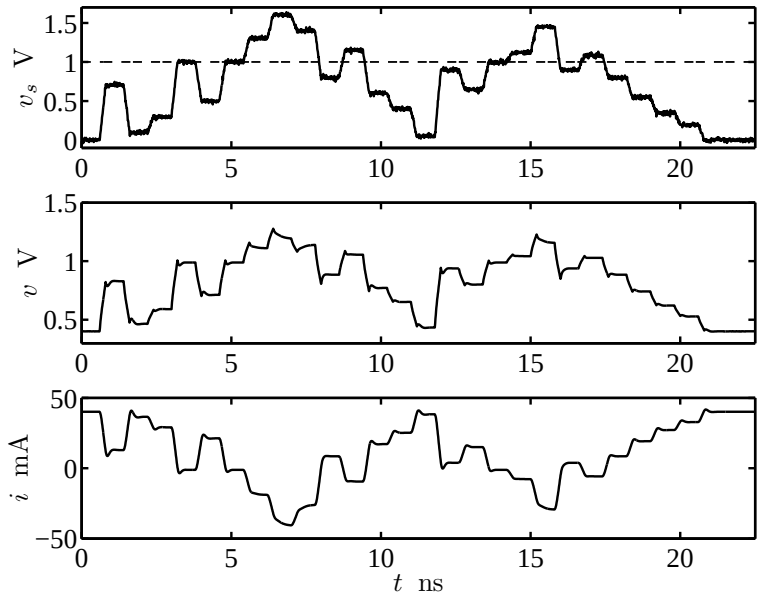

Figure 5.4. Waveforms for the identification test computed for the setup of Figure 5.3. Noisy multilevel voltage source $v_s(t)$ (top panel), identification signals $v(t)$ (middle panel) and $i(t)$ (bottom panel) are shown.

5.3.3 Model estimation

In this step, the unknown parameters defining the model are estimated from the identification signals collected in the previous step. As pointed out in Section 3.4, the parameters are evaluated by minimizing a suitable error function by means of a specific algorithm based on the type of mathematical representation describing the model. In order to evaluate how an estimated model is good in the reproduction of the identification signals, we compute the Mean Squared Error (MSE), defined as

$$\mathrm{MSE} = \frac{1}{N}\sum_{k=1}^{N}\left(\bar{y}(k) - y(k)\right)^2 \tag{5.6}$$

where $\bar{y}(k)$ is the sample of the reference response (in this case the port current of the test device) and $y(k)$ is the sample of the model response. When dealing with SBF and LLSS models, their estimation relies on the application of LM-based iterative algorithms. In this case, the estimation ends when an iteration decreases the MSE of an amount below a small threshold, a priori defined (*e.g.,* 1e-8).

5.3.4 Model validation

According to Section 3.5, accuracy, local stability and efficiency of the estimated model are addressed at this point. The accuracy of the models is quantified by computing the MSE between the model and the test device, for a validation test. The validation test consists in the transient simulation of the test setup of Figure 5.3 based on the ODE formulation (5.3), where v_s is again a multilevel voltage source but different from the one used for the collection of the identification signals and shown in Figure 5.4. In this case, the voltage source v_s is composed of ten levels in the same range spanned by the voltage source depicted in Figure 5.4. The standard-deviation of the superimposed noise and the duration of flat parts are kept to 0.1 mV and 6 ns respectively, whereas the duration of level transition is set to 100 ps. The sampling period used to discretize the signals is $T_s = 10$ ps. Figure 5.5 shows the multilevel voltage sources and the corresponding validation signals computed for the test setup of Figure 5.3.

As discussed in Section 3.5, the local stability is assessed by computing

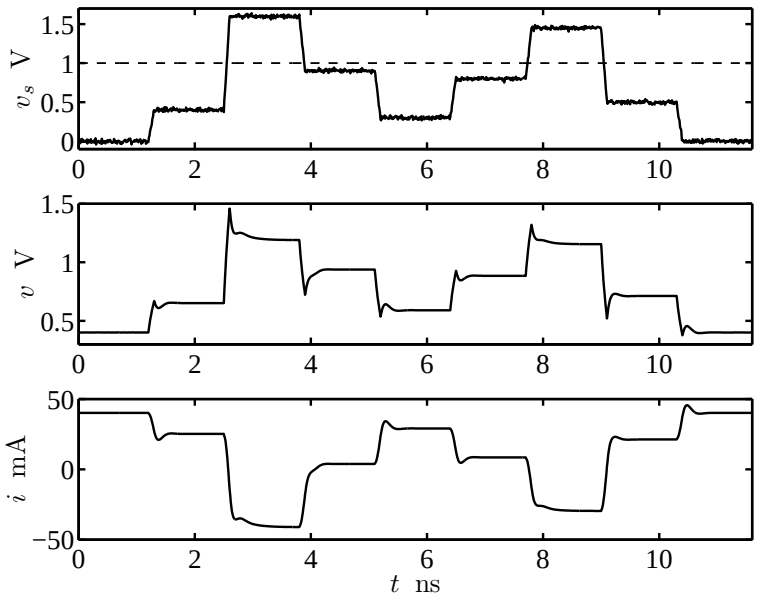

Figure 5.5. Waveforms for the validation test computed for the setup of Figure 5.3. Noisy multilevel voltage source $v_s(t)$ (top panel), validation signals $v(t)$ (middle panel) and $i(t)$ (bottom panel) are shown.

the eigenvalues of the respective linearized model equations [25] (see Appendix B). The eigenvalues are computed for each point explored by the voltage and current responses of the models recorded during the transient simulations of the validation test. With respect to a specific instant of time, some of the eigenvalues may lie outside the unitary circle. The percentage of eigenvalues that lie inside the unitary circle can be used as an indicator of the model stability, for the test at hand. Finally, the efficiency is addressed by quantifying the simulation time required by the different models, for the validation test.

5.4 Model performances

This section describes the effects of the modeling process previously described on the estimation and validation performances of the obtained models. In particular, we investigate the influence of the model representation and of its estimation algorithm.

5.4.1 SBF model

Several tests carried out on the test device under modeling have demonstrated that the performance of SBF models are the same for both fully nonlinear and splitted structures. Therefore, for the sake of simplicity, the following discussion focuses only on the fully nonlinear structure. For this application, the dynamic order of the SBF model (see equations (4.2) and (4.5)) is set to $r = 3$, since we verified that larger values do not lead to better models.

run #	p	MSE of estimation phase	MSE of validation phase	CPU time for model estimation s	Local stability index %
1	6	2.02e-8	3.32e-7	22.0	99.7
2	10	5.46e-9	1.97e-6	15.3	99.8
3	6	5.85e-9	1.12e-5	1.7	96.8
4	7	2.65e-9	3.01e-6	3.0	99.1
5	6	1.81e-8	9.90e-7	1.4	98.3
6	9	2.93e-8	6.19e-7	15.1	99.9
7	7	2.45e-8	1.91e-6	20.2	99.1
8	7	2.17e-9	9.50e-7	3.6	99.7
9	8	5.96e-9	1.22e-6	5.0	99.5
10	9	3.58e-9	9.36e-7	2.8	98.4

Table 5.2. SBF model performances: model size, estimation and validation accuracy, model estimation time and local stability for ten different runs by means of the SLM algorithm [42].

Table 5.2 collects the main figures related to the performances of SBF models estimated by means of ten runs of the static version of the LM algorithm (SLM) [42]. Each row is referred to a specific run and defines the selected model size (p, column 2), the MSE achieved in estimation and validation phases (column 3 and 4), the CPU time required for the estimation of the model (column 5) and the local stability index defined as the percentage of the explored eigenvalues that are inside the unitary circle in the complex plane (column 6). The figures in Table 5.2 highlight that the random initialization of the SLM algorithm does not affect the estimation accuracy of SBF models, since all the run provide comparable values of MSE in the estimation phase. In particular, the MSE level achieved is proven to be satisfactory, as

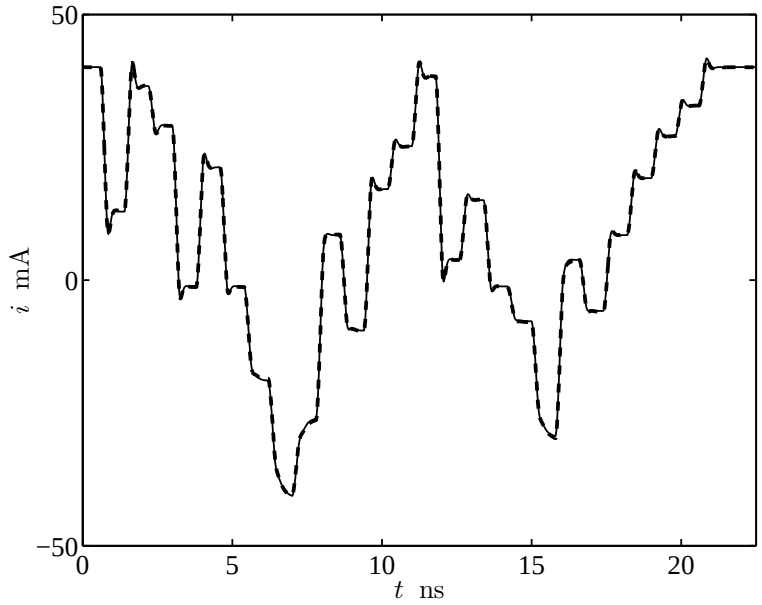

Figure 5.6. Identification port current response. Reference (solid line), and SBF model (dashed line) estimated by the SLM algorithm with the run # 8.

shown in Figure 5.6, that compares the identification port current with the response of the model with the best MSE value (run # 8 in Table 5.2). The responses of the other models are not reported since they are very similar to the best model response. On the contrary, the column of validation MSE indicates how the pseudo-random initialization affects the accuracy of the models when they are excited by a signal different from the one used for the estimation. This remark is confirmed by the validation curves of Figure 5.7, that compares the reference output port current response and the response of all the SBF models listed in Table 5.2. From this comparison it is worth noting that different models provide a quite large variability of response, but the best model (run # 1 in Table 5.2) provides very good results. The last column of Table 5.2, that collects the percentage of model eigenvalues outside the unitary circle, highlights that all the estimated models exhibit a potential local instability during transient simulation. This is also confirmed in Figure 5.8, showing the unitary circle in the complex plane and the position of the eigenvalues explored during the validation test by the best model (run #1). Besides, it is also clear that all the estimated models are very compact, *i.e.,* they are composed by a very limited number of basis functions in the

range $6 \div 10$ (see the second column of Table 5.2).

run #	p	MSE of estimation phase	MSE of validation phase	CPU time for model estimation s	Local stability index %
1	10	2.07e-8	1.70e-4	88.4	12.9
2	10	1.77e-8	2.86e-5	181.5	96.5
3	7	2.22e-8	1.26e-6	157.6	99.8
4	10	8.56e-8	3.47e-6	91.1	37.7
5	4	2.00e-8	3.58e-6	59.1	98.8
6	6	2.81e-8	1.55e-6	83.2	99.7
7	10	1.56e-8	1.62e-6	186.8	99.5
8	9	1.80e-8	1.49e-6	175.0	98.6
9	7	1.90e-8	7.49e-7	67.2	99.4
10	5	5.22e-8	1.02e-6	70.3	76.6

Table 5.3. SBF model performances: model size, estimation and validation accuracy, model estimation time and local stability for ten different runs by means of the RLM algorithm [43].

Similarly, Table 5.3 collects the same comparison for ten runs of the application of the recurrent version of the LM algorithm (RLM) [43]. Similar comments hold for the MSE during estimation, the model size and the local stability. On the contrary, the estimated models exhibit a larger variability, depending on the initial guess of parameters as shown in Figure 5.9. Finally, from the comparison of the two estimation algorithm (SLM and RLM) it is clear that the RLM method is less efficient and requires a larger CPU time for the computation of the model parameters than the static version of the method (compare the 5th column of Tables 5.2 and 5.3).

5.4.2 ESN model

Unlike the SBF case, the modeling with ESN representation has been proven to be accurate only by using the splitted structure. Therefore, the fully nonlinear structure is not considered hereafter. The ESN representation used for modeling the problem at hand has been proven to be accurate enough by selecting the state vector size $n = 100$ (see equation (4.6)).

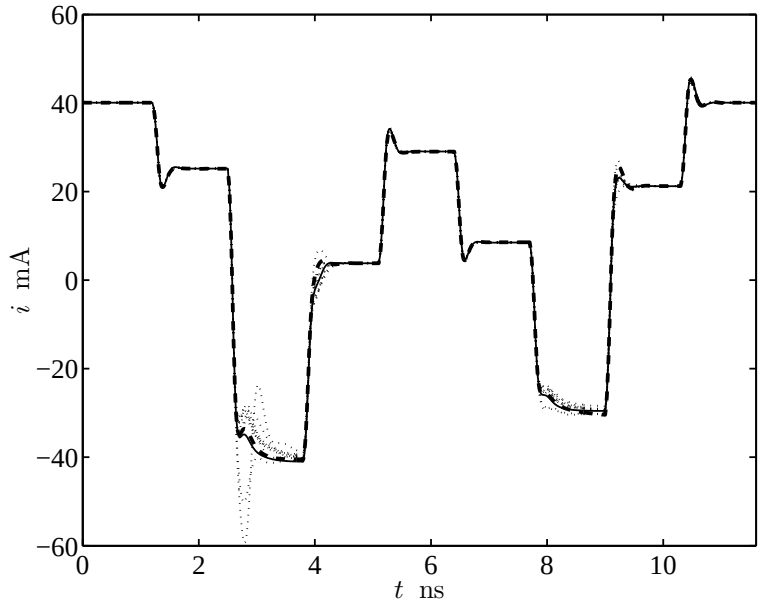

Figure 5.7. SBF model validation: port current response of ten different models obtained by means of the SLM estimation algorithm. Solid line: reference, dashed line: best model (#1), dotted lines: other models.

run #	n	MSE of estimation phase	MSE of validation phase	CPU time for model estimation s	Local stability index %
1	100	5.51e-6	9.19e-6	1.4	100

Table 5.4. ESN model performances: model size, estimation and validation accuracy, model estimation time and local stability for one run by means of the estimation algorithm [36].

Table 5.4 collects the main figures related to the performances of the ESN model estimated by means of the algorithm [36]. The performance indexes are the same as those for the SBF case. However, we report the result of a unique estimation run since we verified that the suggested initialization procedure does not affect the quality of the obtained model. The local stability of the model enforced a-priori is verified. As a confirmation, Figure 5.12 shows the eigenvalues of the linearized ESN model for each point explored during the transient validation test. From Table 5.4 we see that the value of the CPU time required for the estimation is very small, since the estimation relies on a linear least squares algorithm. Finally, Figure 5.11 shows the validation

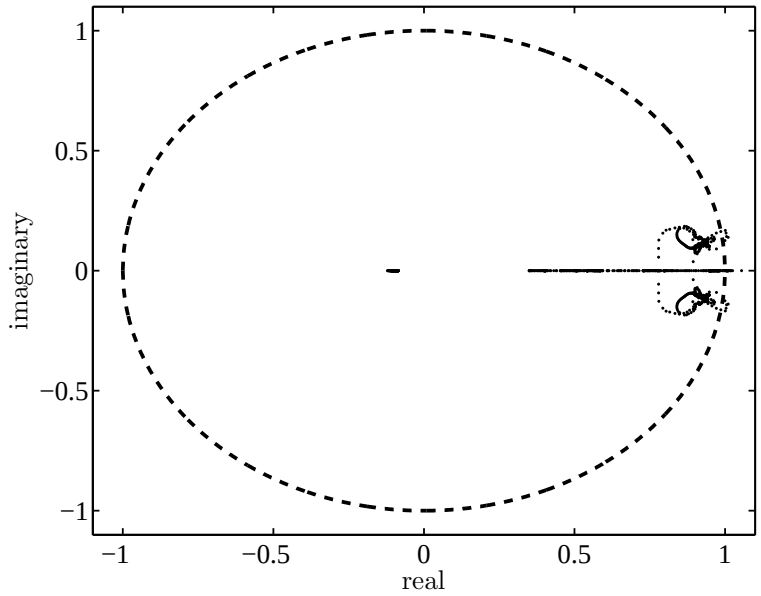

Figure 5.8. Eigenvalues of the linearized SBF model providing the best prediction accuracy shown in Fig. 5.7. The linearization is computed for each point explored during the transient validation test.

curves, thus highlighting a good accuracy of the estimated ESN model. This is also confirmed by the MSE value in Table 5.4.

5.4.3 LLSS model

The modeling of the test device by means of LLSS representation has been proven to be accurate by using either the fully nonlinear or splitted structure. Therefore, as reported for the SBF case, for conciseness we discuss only the performance of the fully nonlinear structures. For this application, the scheduling vector of the LLSS model is composed by using $r = 0$ (see equations (4.7) and (4.8)), since we verified that larger values do not lead to better models.

Table 5.5 collects the main figures related to the performances of the LLSS model estimated with one run by means of the PLM algorithm [37]. The performance indexes are the same as those for the SBF and ESN case. We perform only one run of the estimation since the initialization with the 4SID method [47] is unique and deterministic. In addition, the initialization also provides the dimension of the state vector of each local model, that is $n = 4$ for this case. The reached MSE in estimation has been proven to be

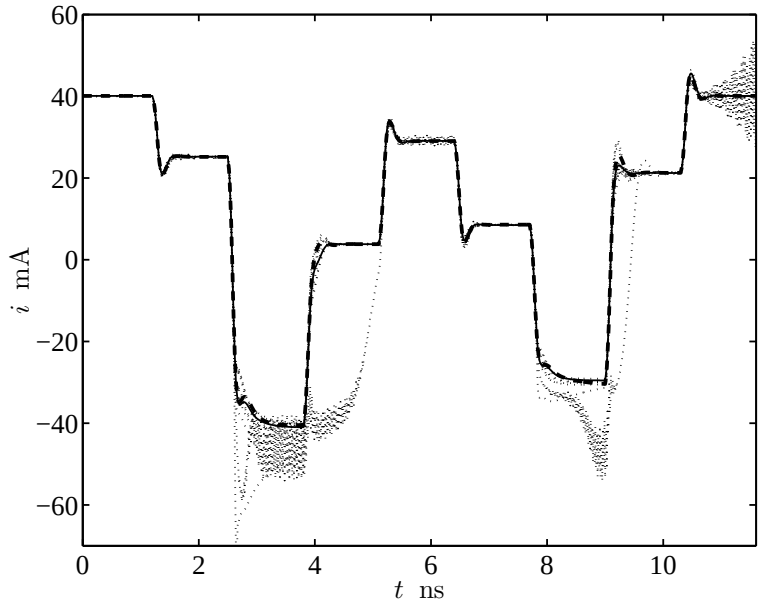

Figure 5.9. SBF model validation: port current response of ten different models obtained by means of the RLM estimation algorithm. Solid line: reference, dashed line: best model (#9), dotted lines: other models.

run #	p	MSE of estimation phase	MSE of validation phase	CPU time for model estimation s	Local stability index %
1	5	4.68e-8	3.24e-7	64.2	100

Table 5.5. LLSS model performances: model size, estimation and validation accuracy, model estimation time and local stability for one run by means of the PLM estimation algorithm [37].

sufficient for a model composed of $p = 5$ local linear models, thus providing a model with a compact size. Besides, Figure 5.13 confirms also the good accuracy of the model in reproducing the validation signals. From Table 5.5, we see how the CPU time required for the estimation is comparable with the one recorded for the SBF models by means of RLM algorithm. Finally, the local stability of the obtained LLSS model is effectively verified a-posteriori, according to the eigenvalues shown in Figure 5.14.

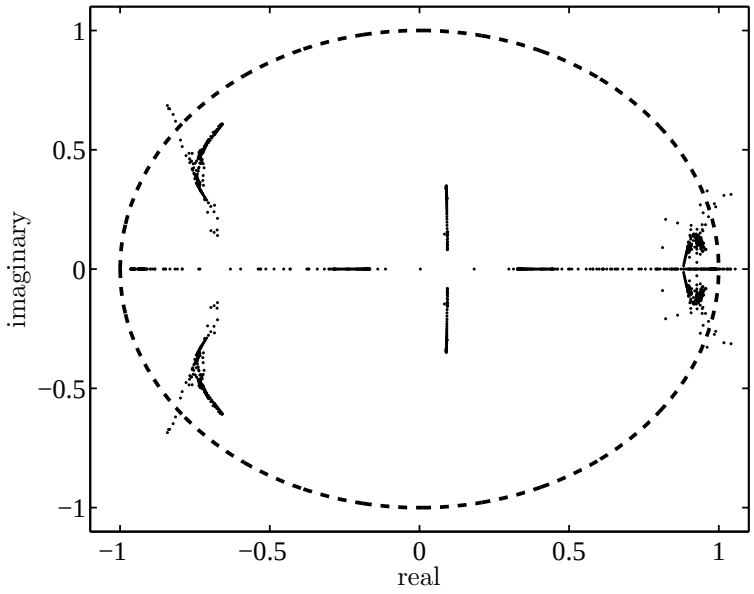

Figure 5.10. Eigenvalues of one of the linearized SBF models providing the best prediction accuracy shown in Fig. 5.9. The linearization is computed for each point explored during the transient validation test.

5.4.4 Efficiency comparison

As a final comparison, Table 5.6 compares the efficiency between the reference ODE model of the one-port test device and the different parametric model representations for the computation of the curves reported in Figure 5.7, 5.9, 5.11 and 5.13. We can clearly observe how both SBF and LLSS models provide a significant speed-up with respect to the reference model. On the contrary, ESN model does not exhibit a significant speed-up, due to the large size of the model.

model	simulation time	speed-up
reference	40 s	x1
SBF	0.2 s	x200
SBF	0.2 s	x200
ESN	16 s	x2.5
LLSS	0.8 s	x50

Table 5.6. Model efficiency performances: Matlab model simulation time and related speed-up for the validation test.

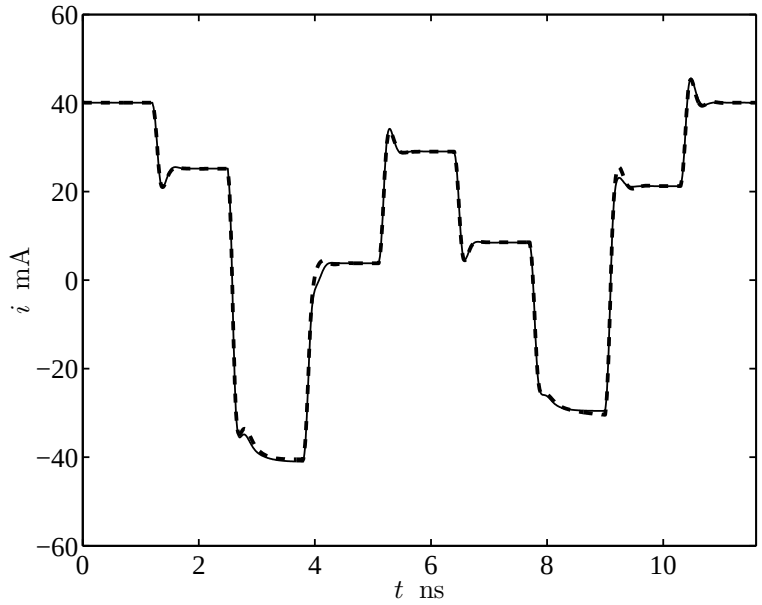

Figure 5.11. ESN validation: port current response of the model obtained by means of the estimation algorithm [36]. Solid line: reference, dashed line: model.

5.5 Summary

This section reports the final comparison on the performance of the selected parametric model representations for the approximation of the constitutive relation of the test device in Figure 5.1. Table 5.7 compares the performances of accuracy, local stability and efficiency, using the same symbols of Table 4.1.

model	accuracy	local stability	efficiency
SBF	↓	↓	↑
ESN	↓	↑	↓↓
LLSS	↑	↑	↑

Table 5.7. Comparison of the features of the model representations for the test device of Figure 5.1.

From Table 5.7, it is clear that for the problem at hand, the LLSS parametric representation is the one providing the best overall behavior. Therefore, this class is adopted for the modeling of real devices involved in a mobile data-link, as reported in Chapter 6.

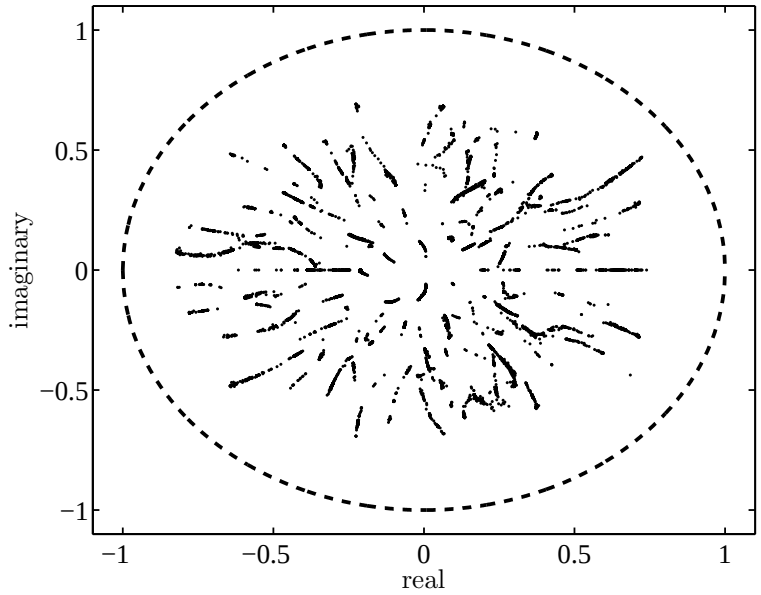

Figure 5.12. Eigenvalues of the linearized ESN model providing the prediction shown in Fig. 5.11. The linearization is computed for each point explored during the transient validation test.

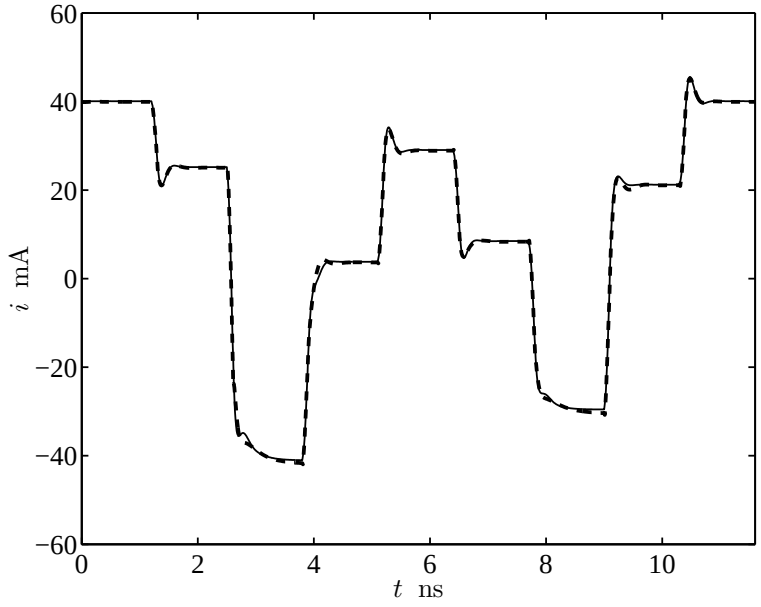

Figure 5.13. LLSS validation: port current response of the model obtained by means of the PLM estimation algorithm [37]. Solid line: reference, dashed line: model.

.

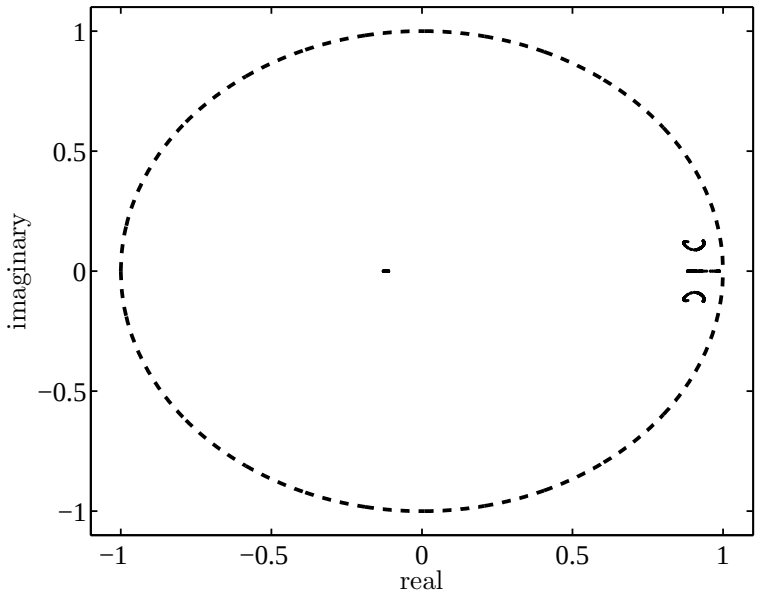

Figure 5.14. Eigenvalues of the linearized LLSS model providing the prediction shown in Fig. 5.13. The linearization is computed for each point explored during the transient validation test.

Chapter 6

Application example

6.1 Introduction

The aim of this section is to discuss the impact of device macromodels on the accuracy of signal integrity and performance predictions for critical data link of high-performance electronic equipments. Specifically, the LLSS parametric representation is addressed for the development of macromodels of an output buffer (*i.e.,* driver) and an input buffer (*i.e.,* receiver) composing a real data-link for mobile phone application. In the following we present the structure and the operating features of the mobile data-link being simulated. Finally, we conclude by presenting the results on the accuracy and the efficiency of the predictions of both the functional signals and the power supply and ground noise.

6.2 Mobile data-link

The example under study is a data-link for mobile applications[1]. Such a data-link represents the RF-to-Digital interface. Figure 6.1 shows the structure under test, which is composed of a driver (left side) and a receiver (right side) communicating via an interconnect, and energized by a common power supply network. A high-speed Nokia CMOS single-ended transceiver ($V_{DD} = 1.8\,\mathrm{V}$) is used in place of the driver and receiver. The interconnection between driver and receiver is a 3 cm-long MCM land, which is modeled from

[1] Courtesy of Nokia Research Center, Radio Technologies laboratory, Helsinki (Finland).

the geometrical details as an ideal transmission line with a characteristic impedance $Z_c = 100\,\Omega$ and a p.u.l. capacitance $C = 5\,\mathrm{pF/m}$. The power supply network considered is modeled by a lumped equivalent, but no transition or junctions are included in the transmission path, for the sake of simplicity. The reference for this structure is the Eldo [50] simulation in which the driver and the receiver are represented by their transistor-level models. For both devices, specialized macromodels based on LLSS representation are derived from the response of such transistor levels by following the modeling process described in Chapter 3 and applied to the test device in Chapter 5. The macromodels are implemented in Eldo subcircuits. The macromodel related to the driver describes the output and power-supply port behavior, whereas the macromodel related to the receiver describes the input and power-supply port behavior. Further details of the model structure of the devices can be found in Appendix D. The simulation of the data-link is carried out by using an input data pattern composed of a 50 bit-long sequence with 5 ns bit time and 500 ps rise time. Our goal for this application is to provide the comparison between the waveforms computed by using the transistor-level models and the waveforms computed by using the macromodels. Two items are addressed. The first consist of the comparison of the functional signals as the driver output voltage v_{21} and the receiver input voltage v_{12}. The second is related to the prediction of the power-supply and ground noise by comparing the voltage fluctuations v_{31} and v_{41}.

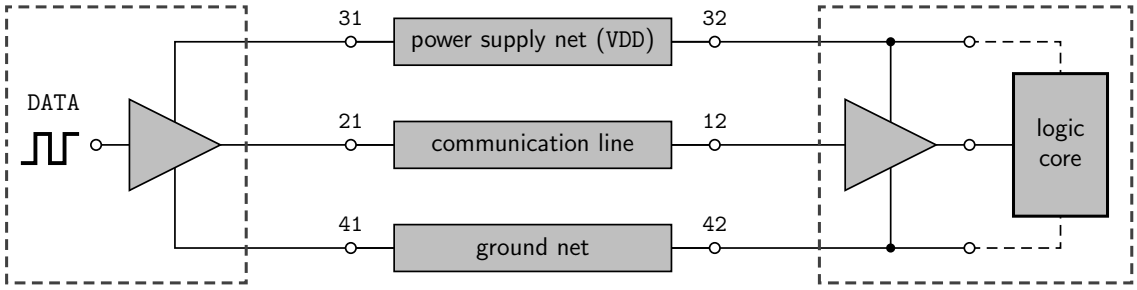

Figure 6.1. Structure setup of the mobile data-link investigated. The relevant blocks and the nodes of interest are reported.

6.3 Results

Figure 6.2 and 6.3 show the reference and predicted waveforms of the voltage v_{21} and v_{12} respectively, within the interval time between 30 ns and 165 ns. A very good correlation among the different curves, especially during the transition, indicates that LLSS macromodels are capable of providing accurate timing information. In fact, the timing error on v_{21} and v_{12}, computed as the maximum delay between the reference and the predicted waveforms at 0.9 V level, turns out to be always less than 2% of bit time for v_{21} and 3% for v_{31}, evaluated over the entire bit sequence.

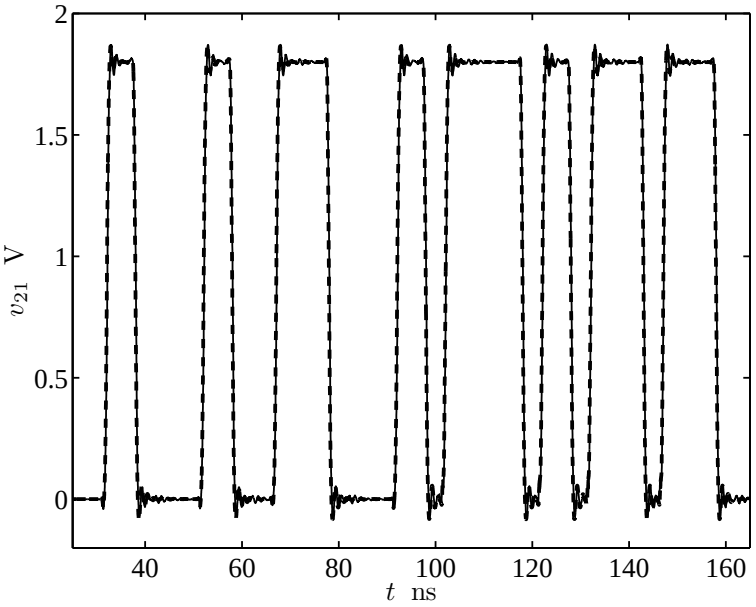

Figure 6.2. Driver output voltage waveforms v_{21} for the simulation of the setup in Figure 6.1. Solid line: reference; dashed line: LLSS macromodels.

Figure 6.4 and 6.5 show the reference and predicted fluctuations of the voltage v_{31} and v_{41} respectively, within the interval time between 35 ns and 75 ns. As for the comparison in Figure 6.2 and 6.3, the accuracy of the LLSS macromodels is confirmed, and the timing errors turn out to be on the order of less than 3% of the bit time over the entire bit sequence for both v_{31} and v_{41}.

Finally, Table 6.1 shows a comparison of the CPU time required by Eldo simulation of the reference 50 bit-long transmission of the setup in Figure 6.1 and the corresponding time of the simulation using LLSS macromodels. The

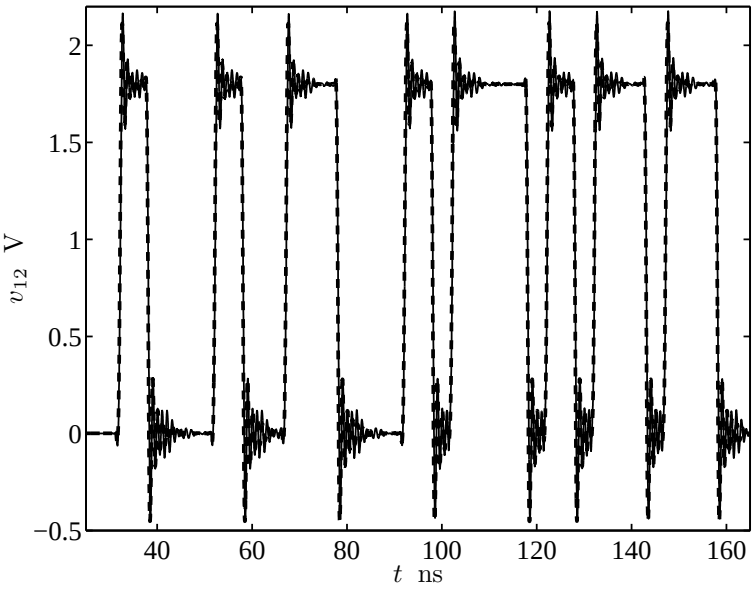

Figure 6.3. Receiver input voltage waveforms v_{12} for the simulation of the setup in Figure 6.1. Solid line: reference; dashed line: LLSS macromodels.

speed-up factor is on the order of 30.

Macromodel (Eldo)	Simulation time
Transistor-level	36 min 26 sec
LLSS	1 min 45 sec

Table 6.1. Time comparison for the simulation of the setup in Figure 6.1 (see text for details).

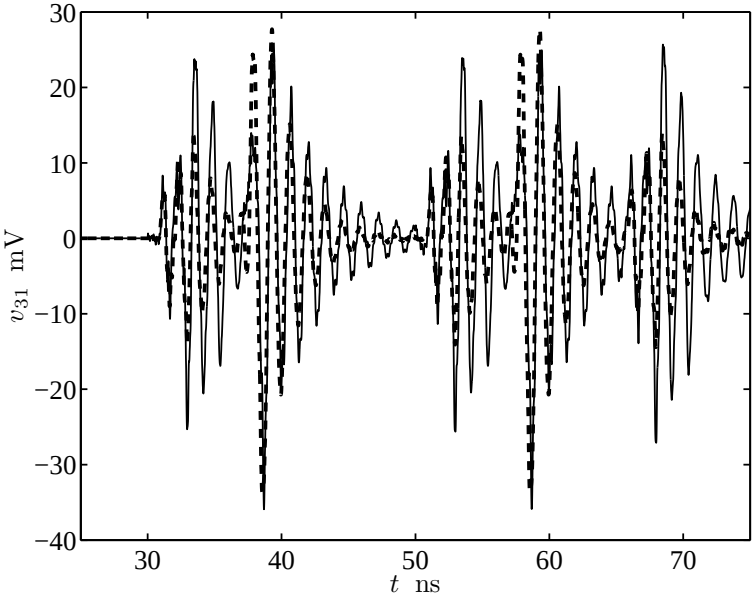

Figure 6.4. Driver power-supply voltage fluctuations v_{31} for the simulation of the setup in Figure 6.1. Solid line: reference; dashed line: LLSS macromodels.

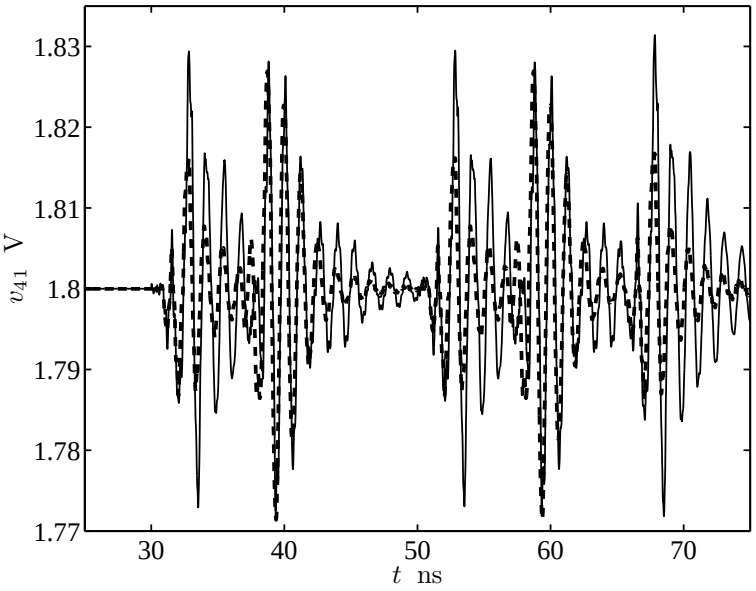

Figure 6.5. Driver ground voltage fluctuations v_{31} for the simulation of the setup in Figure 6.1. Solid line: reference; dashed line: LLSS macromodels.

Chapter 7

Conclusions

The activity carried out in this Thesis concerns the development of numerical macromodels of digital IC ports. Such models are of paramount importance for the assessment of SI/EMC effects in high-performance electronic equipments via system-level simulations.

In order to obtain accurate and efficient macromodels, among the available resources, we concentrate on the black-box modeling approach via system identification methods. Such an approach, that has been recently applied to ICs, amounts to selecting and estimating a suitable nonlinear parametric model from device responses. Parametric models and system identification methods provide an effective modeling tool yielding good IC models. The obtained models allow to accurately reproduce sensitive effects like crosstalk or radiation, can be easily obtained either from simulated or measured transient port signals only and that can be readily implemented in any commercial tool as SPICE subcircuits or AMS descriptions.

In spite of these advantages, the parametric relations used so far have some inherent limitations. Mainly, model stability cannot be easily imposed a-priori or even during the estimation process without impacting on model accuracy. In addition, since the estimation relies on an iterative algorithm, the quality of the obtained model depends on the initial guess of parameters. Furthermore, higher order dynamical effects may not be readily represented by these models and model estimation for real devices with multiple ports is troublesome and impact on the quality of the estimated models.

The present study contributes to the systematic discussion of the open

research issues of the IC modeling process to obtain macromodels overcoming the limitations of the representations and methodologies considered so far. Specifically, the performance of different parametric representations as Sigmoidal Basis Functions (SBF) expansions, Echo State Networks (ESN) and Local Linear State-Space (LLSS) models are investigated. All the representations have proven capabilities for the modeling of unknown nonlinear dynamic systems and are good candidates to be used for the modeling problem at hand. However, the latter two representations have never been applied to the IC modeling and they worth to be investigated. For each model representation, the most suitable estimation algorithm is considered and a systematic analysis is performed to highlight advantages and limitations. For this analysis, the modeling process is applied to a synthetic nonlinear device representative for the IC port class, designed to obtain a stiff modeling benchmark.

The tests carried out show that LLSS models provide the best overall performance for the modeling of digital devices, even with strong nonlinear dynamics. LLSS models can be estimated by means of an efficient estimation algorithm that provides a unique solution. Local stability of models is preconditioned and verified a-posteriori.

The effectiveness of the modeling process based on LLSS representations is verified by applying the proposed technique to the modeling of real devices involved in a realistic data communication link. The link considered is a RF-to-Digital interface used in a mobile phone. The obtained macromodels have been successfully used to predict both the functional signals and the power supply and ground fluctuations. Besides, they turn out to be very efficient, providing a significant speed up of the simulation of the complete data link.

Future work

We are currently working to further explore the features and capabilities of LLSS representation. In particularly, we are interested to formulate a tighter criteria to analyze and enforce the model stability during the estimation of model parameters. Besides, the modeling methodology presented in this study need be extended to account for the enhanced features of devices and applications. In particular, current models do not account for incomplete

state-transitions and cannot be employed for the system-level simulations overclocked systems. Besides, other relevant effects like the Electromagnetic Immunity (EMI) of devices has not been completely investigated and thus included in the present IC models yet.

Bibliography

[1] R. Goyal, "Managing signal integrity," *IEEE Spectrum*, pp. 54–58, March 1994.

[2] F. G. Canavero, S. Grivet-Talocia, I. A. Maio, I. S. Stievano, "Linear and nonlinear macromodels for system-level signal intgrity and EMC assessment", *IEICE Transactions on Communications - Special Issue on EMC*, vol. E88-B, no. 8, pp. 1121–1126, Aug. 2005.

[3] R. Achar, M. S. Nakhla, "Simulation of high-speed interconnects," *Proceedings of the IEEE*, vol. 89, no. 5, pp, 693–728, May 2001.

[4] L. Ljung, "System identification: theory for the user," Prentice-Hall, 1987.

[5] I. S. Stievano, F. G. Canavero, I. A. Maio, "Parametric macromodels of digital i/o ports," *IEEE Transactions on Advanced Packaging*, Vol. 25, No. 2, pp. 255–264, May, 2002.

[6] I. S. Stievano, I. A. Maio, F. G. Canavero, "Behavioral models of i/o ports from measured transient waveforms," *IEEE Transactions on Instrumentation and Measurement*, vol. 51, no. 6m, pp. 1266–1270, Dec. 2002.

[7] I. S. Stievano, I. A. Maio, F. G. Canavero, "Mπlog macromodeling via parametric identification of logic gates," *IEEE Transaction on Advanced Packaging*, Vol. 27, No. 1, pp. 15–23, Feb. 2004.

[8] I. S. Stievano, "Behavioral modeling of nonlinear circuit elements: application to signal integrity and electromagnetic compatibility," Ph.D. Thesis, Politecnico di Torino, Italy, March, 2001.

[9] J. C. Pedro, S. A. Maas, "A comparative overview of microwave and wireless power-amplifier behavioral modeling approaches," *IEEE Transactions on Microwave Theory and Techniques*, vol. 53, no. 4, pp. 1150–1163, Apr. 2005.

[10] C. Siviero, P. M. Lavrador, J. C. Pedro, "A frequency domain extraction procedure of low-pass equivalent behavioral models of microwave PAs," in Proc. of the *European Microwave Integrated Circuits Conference (EuMIC)*, Manchester (UK), pp. 253–256, Sep. 10–13, 2006.

[11] J. M. Rabaey, "Digital integrated circuits - a design perspective." Prentice &

Hall Electronics and VLSI Series, 1996.

[12] "Ibis cookbook for v4.0," available on http://www.eigroup.org/ibis/-default.htm, Sep. 16, 2005.

[13] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P. Y. Glorennec, H. Hjalmarsson, A. Juditsky, "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, Vol. 31, No. 12, pp. 1691–1724, 1995.

[14] H. R. Zhang, X. D. Wang, C. J. Zhang, X. S. Cai, "Robust identification of non-linear dynamic systems using support vector machine," *IEE Proceedings Science, Measurement and Technology*, Vol. 153, Issue 3, pp. 125–129, May 2006.

[15] R. Murray Smith and T. A. Johansen, "Multiple model approaches to modeling and control," Taylor & Francis, 1997.

[16] M. Schetzen, "The Volterra and Wiener theories of nonlinear systems," John & Wiley Sons, 1980.

[17] J. Vörös, "Identification of nonlinear dynamic systems using extended Hammerstein and Wiener models," *Control Theory and Advanced Technology*, vol. 10, no. 4, part 2, pp. 1203–1212, Jun. 1995.

[18] A. Ponchet, J. L. Ponchet, G. S. Moschytz, "On the input/output approximation of nonlinear systems," in Proc. of the *IEEE International Symposium on Circuit and Systems / ISCAS-95*, pp. 1500–1503, May 1995.

[19] C. Siviero, I. S. Stievano, I. A. Maio, "Behavioral modeling of IC output buffers: a case study," in Proc. of the *2005 IEEE PhD Research in Microelectronics and Electronics (PRIME)*, Lausanne (Switzerland), pp. 366–369, July 25–28, 2005.

[20] D. G. Luenberger, "Linear and nonlinear programming," Addison-Wesley Publishing Company, 1989.

[21] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning," Kluwer Academic Publishers, 1989.

[22] S. Singhal, L. Wu, "Training multilayer perceptrons with the extended Kalman algorithm," *Advances in Neural Information Processing Systems 1*, pp. 133–140, D.S. Touretzky, 1989.

[23] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi, "Optimization by simulated annealing," *Science*, Vol. 220, No. 4598, May 1983.

[24] H. K. Kalihl, "Nonlinear systems," Prentice-Hall, 2004.

[25] C. Alippi, V. Piuri, "Neural modeling of dynamic systems with nonmeasurable state variables", *IEEE Transaction on Instrumentation and Measurement*, Vol. 48, No. 6, pp. 1073–1080, Dec. 1999.

[26] "The Spice Page," EECS Department of the University of California at Berkeley, available on http://bwrc.eecs.berkeley.edu/Classes/IcBook/SPICE.

[27] "Overview on Verilog-AMS," Accellera Verilog Analog Mixed-Signal Group, available on http://www.verilog.org/verilog-ams/htmlpages/overview.html.

[28] *IEEE Standard VHDL Analog and Mixed-Signal Extensions*, IEEE Standard 1076.1-1999, Mar. 18, 1999.

[29] E. Christen, K. Bakalar, "VHDL-AMS - A hardware description language for analog and mixed-signal applications," *IEEE Transactions on Circuits and Systems II*, vol. 46, no. 3, Oct., pp. 1263–1271, Oct. 1999.

[30] I. Rivals and L. Personnaz, "Black-box modeling with state-space neural networks," in *Neural and Adaptive Control Technology*, World Scientific Singapore, pp. 237–264, 1996.

[31] J. Paduart, J. Schoukens, R. Pintelon, T. Coen, "Nonlinear state space modelling of multivariate systems," in Proc. of the *14th IFAC Symposium on System Identification*, Newcastle (Australia), pp. 565–569, Mar. 29–31, 2006.

[32] C. Gan, K. Danai, "Model-based recurrent neural network for modeling nonlinear dynamic systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 30, pp. 344-351, Apr. 2000.

[33] H. J. Palanthandalam-Madapusi, S. Gillijns, B. De Moor, D. S. Bernstein, "Subsystem identification for nonlinear model updating," in Proc. of the *American Control Conference*, Minneapolis, Minnesota, USA, pp. 3056–3062, Jun. 14–16, 2006.

[34] H. Jaeger, "The echo state approach to analyzing and training recurrent neural networks," GMD - German National Research Institute for Computer Science, GMD Report 148, 2001.

[35] V .Verdult, "Nonlinear system identification: a state-space approach", Ph.D. Thesis, University of Twente, The Netherlands, March, 2002.

[36] H. Jaeger, "Tutorial on training recurrent neural networks, covering BPTT, RTRL, EKF and the echo state network approach," Fraunhofer Institute AIS, GMD Report 159, 2002.

[37] V .Verdult, L. Ljung, M. Verhaegen, "Identification of composite local linear state-space models using a projected gradient search," *International Journal of Control*, Vol. 65, Nos. 16/17, pp. 1385–1398, 2002.

[38] S. Haykin, "Neural networks - a comprehensive foundation," Prentice Hall,

1999.

[39] K. S. Narendra, K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, pp. 4–27, Jan. 1990.

[40] J. J. Moreé, "The Levenberg Marquardt algorithm: implementation and theory," in *Numerical Analysis*, vol. 630 of *Lecture Notes in Mathematics*, pp. 106–116, Springer Verlag, 1978.

[41] D. Nguyen, B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," Proc. of the *International Joint Conference on Neural Networks (IJCNN)*, San Diego, CA, USA, pp. 21-26, Jun. 17–21, 1990.

[42] M. T. Hagan, M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, Vol. 5, No. 6, pp. 989–993, Nov. 1994.

[43] Y. H. Fang, M. C. E. Yagoub, F. Wang, Q. J. Zhang, "A new macromodeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 48, no. 12, pp. 2335–2344, Dec. 2000.

[44] M. Autin, M. Biey, M. Hasler, "Order of discrete time nonlinear systems determined from input/output signals," in Proc. of the *IEEE International Symposium on Circuit and Systems / ISCAS-92*, San Diego CA, pp. 296–299, May 10–13, 1992.

[45] M. Salmen, P. G. Ploger, "Echo State Networks used for motor control," in Proc. of the *International Conference on Robotics and Automation*, pp. 1953–1958, Apr. 18–22, 2005.

[46] M. Buehener, P. Young, "A tighter bound for the echo state property," *IEEE Transactions on Neural Networks*, Vol. 17, No. 3, May 2006.

[47] M. Verhaegen, "Identification of the deterministic part of MIMO state space models given in innovations form from input-output data," *Automatica*, Vol. 30, No. 1, pp. 61–74, 1994.

[48] J. Sjöberg, "On estimation of nonlinear black-box models: how to obtain a good initialization," In Proc. of the *1997 IEEE Workshop Neural Networks for Signal Processing VII*, Amelia Island Plantation FL, pp. 72–81, Sep. 1997

[49] L. F. Shampine, M. W. Reichelt, "The Matlab ODE suite," *SIAM Journal on Scientific Computing*, Vol. 18, pp. 1–22, 1997.

[50] "Eldo user's manual," Mentor Graphics Corporation, 2006.

[51] I. A. Maio, I. S. Stievano, P. Savi, F. G. Canavero, "Single and multi-piece behavioral models of IC output buffers," in Proc. of the *Electronics Systemintegration Technology Conference (ESTC 2006)*, Dresden (Germany), pp. 209–214, Sep. 5–7, 2006.

[52] J. Sjöberg, "Nonlinear system identification with neural networks," Ph.D. Thesis, Linköping University, Sweden, 1995.

Appendix A

Model structures

This Appendix justifies the splitting of a device port equation into the sum of a static and a dynamic contribution, and the representation of the latter in terms of a nonlinear parametric equation. In general, a constitutive relation of a nonlinear dynamic system can be described by an arbitrary state-space representation involving external measurable variables and internal nonmeasurable state variables, as follows

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{g}\left(\mathbf{x}(t),\mathbf{u}(t)\right) \\ y(t) &= f\left(\mathbf{x}(t),\mathbf{u}(t)\right) \end{cases} \tag{A.1}$$

where y is the output variable, $\mathbf{x}$ is the vector of internal state variables, $\mathbf{u}$ is the vector of input variables, and $\mathbf{g}$ and f are multivariate nonlinear mappings. If we designate by $f_s(\mathbf{u})$ the above nonlinear mapping in static conditions, (A.1) can be recast in the following form

$$\begin{cases} \dot{\mathbf{x}}(t) &= \mathbf{g}\left(\mathbf{x}(t),\mathbf{u}(t)\right) \\ y(t) &= f_s(\mathbf{u}(t)) + \left[f\left(\mathbf{x}(t),\mathbf{u}(t)\right) - f_s(\mathbf{u}(t))\right] \end{cases} \tag{A.2}$$

where the output equation turns out to be interpreted as the sum of a static and a dynamic part, *i.e.,*

$$y(t) = f_s(\mathbf{u}(t)) + f_d\left(\mathbf{x}(t),\mathbf{u}(t)\right) \tag{A.3}$$

where $f_d\left(\mathbf{x},\mathbf{u}\right) = \left[f\left(\mathbf{x},\mathbf{u}\right) - f_s(\mathbf{u})\right]$ is a new mapping such that $f_d\left(\mathbf{x},\mathbf{u}\right) = 0$ for constant $\mathbf{u}(t)$.

Appendix B

Local stability analysis

B.1 Introduction

For the case of nonlinear discrete-time parametric models, local stability analysis can be easily verified by linearizing model equation. For both NIO and NSS structures described in Chapter 4, the procedure amounts to representing both models as state-space relations. It is worth to remark that NSS are native state-space models, whereas the alternate NIO can be easily converted in a state-space form as suggested in [52]. Then, the first-order Taylor approximation of the equation updating the state vector is computed for each time-sample. For the k-th sample, this writes

$$\mathbf{x}(k) + \Delta\mathbf{x} = \mathbf{x}(k) + \mathbf{A}\Delta\mathbf{x} + \dots \tag{B.1}$$

where $\mathbf{x}(k)$ is the state vector at the current time sample, $\Delta\mathbf{x}$ is a generic incremental vector and $\mathbf{A}$ is the square matrix describing the first-order term. Then, in the second step the computation of the eigenvalues of $\mathbf{A}$ in (B.1) is performed. The model is local (asymptotically) stable if all the eigenvalues of $\mathbf{A}$ have magnitude smaller than one, *i.e.,* on the complex plane all the eigenvalues lie inside the unitary circle.

In the following, the mathematical steps for the computation of the linearization (B.1) for both the NIO and the NSS representations are reported. For the sake of simplicity, the discussion is focused on single-input single-output model where y and u are the output and input variables, respectively.

B.2 Nonlinear input-output model

According to equation (4.2), the general relation for a NIO model writes

$$\left\{ \begin{array}{rcl} y(k) & = & F\,(\,\boldsymbol{\Theta}\,;\,\varphi(k)\,) \\ \varphi(k) & = & [\,y(k-1)\,,\,\ldots\,,y(k-r)\,,u(k)\,,\,\ldots\,,u(k-r)\,]^{\mathrm{T}} \end{array} \right. \tag{B.2}$$

In order to apply the above local stability criterion, we must convert model (B.2) into an equivalent state-space form [52]. This writes

$$\left\{ \begin{array}{l} \mathbf{x}(k) = \left[\begin{array}{c} F\left(\,\boldsymbol{\Theta}\,;\,\left[\mathbf{x}^{\mathrm{T}}(k-1)\,,u(k)\,,\,\ldots\,,u(k-r)\right]^{\mathrm{T}}\right) \\ x_1(k-1) \\ x_2(k-1) \\ \vdots \\ x_{r-1}(k-1) \end{array} \right] \\ \\ y(k) = x_1(k) \end{array} \right. \tag{B.3}$$

where $\mathbf{x}(k) = [\,y(k)\,,\,\ldots\,,y(k-r+1)\,]^{\mathrm{T}}$ is the equivalent state vector. With respect to the representation (B.3), the matrix $\mathbf{A}$ of equation (B.1) is expressed as

$$\mathbf{A} = \left[\begin{array}{cccc} a_1 & a_2 & \ldots & a_r \\ 1 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 1 & 0 \end{array} \right] \tag{B.4}$$

where the i-th term of the first row of (B.4) is computed as

$$a_i = \frac{\partial F\,(\,\boldsymbol{\Theta}\,;\,\varphi(k)\,)}{\partial y(k-i)} \tag{B.5}$$

B.3 Nonlinear state-space model

According to equation (4.3), the general form of a NSS model writes

$$\left\{ \begin{array}{rcl} \mathbf{x}(k) & = & \mathbf{F}_1\,(\,\boldsymbol{\Theta}_1\,;\,\mathbf{x}(k-1)\,,u(k-1)\,) \\ y(k) & = & F_2\,(\,\boldsymbol{\Theta}_2\,;\,\mathbf{x}(k)\,,u(k)\,) \end{array} \right. \tag{B.6}$$

where $\mathbf{x} = [x_1\ ,\ \ldots\ ,\ x_n]^{\mathrm{T}}$. Since (B.6) is a state-space representation, we can directly evaluate (B.1). In this case matrix $\mathbf{A}$ is described as

$$\mathbf{A} = \begin{bmatrix} \dfrac{\partial F_{1,1}}{\partial x_1} & \cdots & \dfrac{\partial F_{1,1}}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \dfrac{\partial F_{1,n}}{\partial x_1} & \cdots & \dfrac{\partial F_{1,n}}{\partial x_n} \end{bmatrix} \tag{B.7}$$

Appendix C

Levenberg-Marquardt method

C.1 Introduction

In this Appendix, the widely used Levenberg-Marquardt algorithm [40] for the computation of the parameters of a nonlinear discrete-time parametric model is presented. This algorithm, that is a gradient-based iterative method, has been proven to be particularly effective and numerically efficient. A large number of application exist in the nonlinear optimization, system identification and neural network literature [13, 20].

For the sake of simplicity, the following discussion is based on a single-input single-output system, with input $\bar{u}$ and output $\bar{y}$. In general, as already outlined in Chapter 3, the unknown parameters defining a parametric model are estimated from transient responses of the device under modeling (referred to as the estimation data set). For the single-input single-output case, the estimation data set is composed by an input excitation vector $\bar{\mathbf{u}} = [\, \bar{u}(1) \,,\, \ldots \,,\, \bar{u}(N) \,]^{\mathrm{T}}$ and by the vector collecting the system response $\bar{\mathbf{y}} = [\, \bar{y}(1) \,,\, \ldots \,,\, \bar{y}(N) \,]^{\mathrm{T}}$. The model parameters are then computed by minimizing a suitable error function of the estimation data set and of the responses of the model y. The vector collecting the model response will be indicated as $\mathbf{y} = [\, y(1) \,,\, \ldots \,,\, y(N) \,]^{\mathrm{T}}$ hereafter.

The estimation problem can be stated as the standard nonlinear optimization form

$$\min_{\boldsymbol{\Theta}} C(\boldsymbol{\Theta}) \tag{C.1}$$

where $C : \mathbb{R}^L \rightarrow \mathbb{R}$ is a nonlinear cost function, $\boldsymbol{\Theta} = [\, \theta_1 \,,\, \ldots \,,\, \theta_L \,]^{\mathrm{T}}$ being the

vector of parameters. Specifically, for our problem the cost function in (C.1) is defined as

$$C(\boldsymbol{\Theta}) = \mathbf{e}^{\mathrm{T}}(\boldsymbol{\Theta})\mathbf{e}(\boldsymbol{\Theta}) \tag{C.2}$$

where $\mathbf{e}(\boldsymbol{\Theta}) = \bar{\mathbf{y}} - \mathbf{y}$, $\boldsymbol{\Theta}$ being embedded within $\mathbf{y}$.

As suggested in [40], the procedure that updates the model parameters $\boldsymbol{\Theta}$ at the i-th iteration is defined by the following rule

$$\boldsymbol{\Theta}^{(i+1)} = \boldsymbol{\Theta}^{(i)} - \left[\mathbf{J}^{\mathrm{T}}(\boldsymbol{\Theta}^{(i)})\mathbf{J}(\boldsymbol{\Theta}^{(i)}) + \mu^{(i)}\mathrm{I}\right]^{-1}\mathbf{J}^{\mathrm{T}}(\boldsymbol{\Theta}^{(i)})\mathbf{e}(\boldsymbol{\Theta}^{(i)}) \tag{C.3}$$

where $\mu^{(i)}$ is the LM regularization parameter, I is the identity matrix and $\mathbf{J}$ is the jacobian matrix of the cost function over the parameters, defined as

$$\mathbf{J}(\boldsymbol{\Theta}) = -\begin{bmatrix} \dfrac{\partial y(1)}{\partial \theta_1} & \dfrac{\partial y(1)}{\partial \theta_2} & \cdots & \dfrac{\partial y(1)}{\partial \theta_L} \\ \dfrac{\partial y(2)}{\partial \theta_1} & \dfrac{\partial y(2)}{\partial \theta_2} & \cdots & \dfrac{\partial y(2)}{\partial \theta_L} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial y(N)}{\partial \theta_1} & \dfrac{\partial y(N)}{\partial \theta_2} & \cdots & \dfrac{\partial y(N)}{\partial \theta_L} \end{bmatrix} \tag{C.4}$$

In the following we briefly report the mathematical details related to the computation of the elements of matrix (C.4) for both the NIO and the NSS classes of model representations.

C.2 Nonlinear input-output model

NIO representation is described in Chapter 4. For this class of model, the computation of the element of the jacobian matrix (C.4) depends on the strategy adopted for the evaluation of the model response $y(k)$ to the dataset input $\bar{u}(k)$. There are basically two versions.

The first way is referred to the Static LM (SLM). For this method, the model response writes

$$\begin{cases} y(k) &= F\left(\,\boldsymbol{\Theta}\,;\,\varphi(k)\,\right) \\ \varphi(k) &= \left[\,\bar{y}(k-1)\,,\,\ldots\,,\bar{y}(k-r)\,,\bar{u}(k)\,,\,\ldots\,,\bar{u}(k-r)\,\right]^{\mathrm{T}} \end{cases} \tag{C.5}$$

where past samples of the model output in $\varphi(k)$ are replaced by the corresponding samples of the estimation dataset output. For this version, the evaluation of each element of matrix (C.4) writes

$$\frac{\partial y(k)}{\partial \theta_i} = \frac{\partial F\left(\,\boldsymbol{\Theta}\,;\,\varphi(k)\,\right)}{\partial \theta_i} \tag{C.6}$$

In particular, when dealing with NIO model expressed as sigmoidal expansion (SBF), SLM method represents the typical scheme used for the estimation of static feedforward neural networks [42, 38].

The second way is referred to the Recurrent LM (RLM). This version computes the model response as

$$\begin{cases} y(k) &= F\left(\,\boldsymbol{\Theta}\,;\,\varphi(k)\,\right) \\ \varphi(k) &= \left[\,y(k-1)\,,\,\ldots\,,y(k-r)\,,\bar{u}(k)\,,\,\ldots\,,\bar{u}(k-r)\,\right]^{\mathrm{T}} \end{cases} \tag{C.7}$$

where the recurrent nature of the model is effectively accounted within $\varphi(k)$. Unlike the SLM method, the computation of each element of matrix (C.4) is evaluated by means of the following dynamic equation

$$\begin{aligned} \frac{\partial y(k)}{\partial \theta_i} &= \frac{\partial F\left(\,\boldsymbol{\Theta}\,;\,\varphi(k)\,\right)}{\partial \theta_i} \\ &+ \sum_{j=1}^{r} \frac{\partial F\left(\,\boldsymbol{\Theta}\,;\,\varphi(k)\,\right)}{\partial y(k-j)}\,\frac{\partial y(k-j)}{\partial \theta_i} \end{aligned} \tag{C.8}$$

When the NIO model is expressed as sigmoidal expansion (SBF), the RLM method belongs to the scheme adopted for the estimation of recurrent neural networks [43].

C.3 Nonlinear state-space model

According to Chapter 4, the response of a NSS model to the dataset input $\bar{u}(k)$ writes

$$\begin{cases} \mathbf{x}(k) &= \mathbf{F}_1\left(\,\boldsymbol{\Theta}_1\,;\,\mathbf{x}(k-1)\,,\,\bar{u}(k-1)\,\right) \\ y(k) &= F_2\left(\,\boldsymbol{\Theta}_2\,;\,\mathbf{x}(k)\,,\,\bar{u}(k)\,\right) \end{cases} \tag{C.9}$$

where $\mathbf{x}(k) = \left[x_1(k)\,,\,x_2(k)\,,\,\ldots\,,\,x_n(k)\right]^{\mathrm{T}}$. In this case we have $\boldsymbol{\Theta} = \left[\boldsymbol{\Theta}_1^{\mathrm{T}},\boldsymbol{\Theta}_2^{\mathrm{T}}\right]^{\mathrm{T}}$. The computation of each element of matrix (C.4) relies on the evaluation of

the following state equation

$$\begin{cases} \dfrac{\partial x_j(k)}{\partial \theta_i} &= \dfrac{\partial F_{1,j}\left(\, \boldsymbol{\Theta}_1 \,;\, \mathbf{x}(k-1)\,,\, \bar{u}(k-1)\,\right)}{\partial \theta_i} \\ &+ \displaystyle\sum_{l=1}^{n} \dfrac{\partial F_{1,j}\left(\, \boldsymbol{\Theta}_1 \,;\, \mathbf{x}(k-1)\,,\, \bar{u}(k-1)\,\right)}{\partial x_l(k-1)} \dfrac{\partial x_l(k-1)}{\partial \theta_i} \\ \dfrac{\partial y(k)}{\partial \theta_i} &= \dfrac{\partial F_2\left(\, \boldsymbol{\Theta}_2 \,;\, \mathbf{x}(k)\,,\, \bar{u}(k)\,\right)}{\partial \theta_i} \\ &+ \displaystyle\sum_{j=1}^{n} \dfrac{\partial F_2\left(\, \boldsymbol{\Theta}_2 \,;\, \mathbf{x}(k)\,,\, \bar{u}(k)\,\right)}{\partial x_j(k)} \dfrac{\partial x_j(k)}{\partial \theta_i} \end{cases} \tag{C.10}$$

Appendix D

Review of basic macromodels

D.1 Introduction

The parametric identification approach has been initially applied to the macromodeling of input and output ports of single-ended CMOS ICs. This Appendix shortly reviews the model representations used for these basic modeling problems. A complete discussion of input and output port modeling as well as several modeling examples involving commercial devices of different classes and starting from both transistor-level simulations and actual measurements are reported in [5, 6, 19].

D.2 Receivers

This Section discusses the macromodeling of the input port of receivers, like the one shown in Figure D.1. The modeling of an uncoupled input port is conceptually simple, since it amounts to looking for an approximation of the constitutive relation between the port voltage and current variables.

As the behavior of the port in the range of the power supply voltage is mainly linear and dynamic, and since it becomes strongly nonlinear for voltages outside the power supply voltage range, we use a model representation defined by

$$i(k) = i_l(\boldsymbol{\Theta}_l\,;\, v(k)) + i_{nl}(\boldsymbol{\Theta}_{nl}\,;\, v(k)) \tag{D.1}$$

where i and v denote the port current and voltage variables, respectively, defined by associate reference directions (see Figure D.1). Here we assume

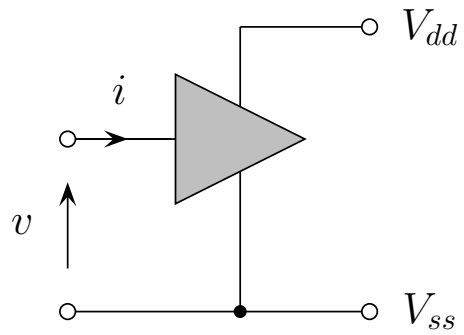

Figure D.1. Generic structure of an input port and its relevant electric variables. V_{dd} and V_{ss} indicate the power supply voltages.

that the current i splits into the sum of two contributions, of which i_l is a linear parametric model defined by an Auto Regressive with eXtra input model (ARX) scheme [4], and accounting for the linear behavior of the port; i_{nl} is a nonlinear parametric model accounting for the port behavior in the voltage range where the effects of protection circuits take place.

It is ought to remark that, in spite of its simplicity, the modeling of input ports may be a challenging problem, because the linear region may have high dynamic order and the onset of the nonlinear regime my be abrupt and may introduce additional and slow time constants. In spite of the above critical points, model representation (D.1) and a careful tuning of the modeling process allow to obtain good models for most cases of practical interest.

D.3 Drivers

Figure D.2 shows the typical structure of a digital driver made of cascaded stages. The electrical variables relevant for the model, *i.e.,* the voltage and current of the output and of the power supply ports (denoted by v, i, v_{dd} and i_{dd}) are defined in Fig D.2, as well.

Parametric macromodels of digital driver exploit the so-called two-piece representation. As an example, for the output port the model writes

$$\begin{aligned} i(k) &= w_H F_H(\boldsymbol{\Theta}_H\,;\, v(k)\,, v_{dd}(k)) \\ &+ w_L F_L(\boldsymbol{\Theta}_L\,;\, v(k)\,, v_{dd}(k)) \end{aligned} \tag{D.2}$$

where w_H and w_L are switching signals accounting for the device state transitions and playing the same role of the signal v_i in Figure D.2, F_H and F_L

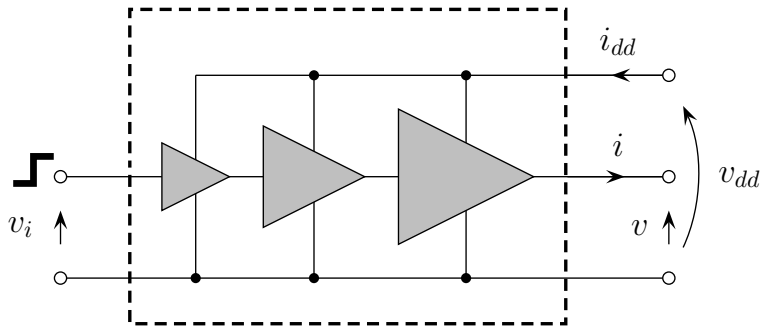

Figure D.2. Typical structure of a driver circuit and its relevant (output and power supply) port electrical variables.

are nonlinear parametric models accounting for the device behaviour in fixed logic HIGH and LOW states, respectively [5, 7]. The same model structure is assumed also for modeling the power supply port. Piecewise models similar to (D.2) occur also in conventional modeling approaches (e.g., IBIS [12]). The representation (D.2) yields models that are inherently accurate for operation in fixed logic states, because in that condition only one of the two submodels is active. In particular, the use of parametric nonlinear submodels allows to reproduce also complicated device behaviors [7]. Nevertheless, possible inaccuracies arise during state transitions, where (D.2) must mimic the behavior of the modeled device by using F_H and F_L, that contain information on fixed logic state operation only. The study of model descriptions that can overcome the previous limitation represents an interesting research area. Recently, some preliminary results on this topic have been already published in [51].

Publications

During the PhD activity, the author has contributed to the publication of the following papers, whose title and abstract are listed below

Journals papers

I.S. Stievano, I.A. Maio, F.G. Canavero, C. Siviero, "Parametric Macromodels of Differential Drivers and Receivers," *IEEE Transactions on Advanced Packaging*, pp. 189–196, Vol. 28, No. 2, May, 2005.

Abstract – This paper addresses the modeling of differential drivers and receivers for the analog simulation of high-speed interconnection systems. The proposed models are based on mathematical expressions, whose parameters can be estimated from the transient responses of the modeled devices. The advantages of this macromodeling approach are: improved accuracy with respect to models based on simplified equivalent circuits of devices; improved numerical efficiency with respect to detailed transistor-level models of devices; hiding of the internal structure of devices; straightforward circuit interpretation; or implementations in analog mixed-signal simulators. The proposed methodology is demonstrated on example devices and is applied to the prediction of transient waveforms and eye diagrams of a typical low-voltage differential signaling (LVDS) data link.

I.S. Stievano, I.A. Maio, F.G. Canavero, C. Siviero, "Reliable Eye-Diagram Analysis of Data Links via Device Macromodels," *IEEE Transactions on Advanced Packaging*, pp. 31–38, Vol. 29, No. 1, February, 2006.

Abstract – This paper addresses the impact of device macromodels on the accuracy of signal integrity and performance predictions for critical digital interconnecting systems. It exploits nonlinear parametric models for both single-ended and differential devices, including the effects of power supply fluctuations and receiver bit detection. The analysis demonstrates that the use of well-designed macromodels dramatically speeds up the simulation as well it preserves timing accuracy even for long bit sequences.

I.S. Stievano, I.A. Maio, F.G. Canavero, C. Siviero, "Parametric Macromodels of Differential Drivers with Pre-Emphasis," *IEEE Transactions on Advanced Packaging*, accepted, to be published.

Abstract – This paper discusses the extraction of behavioral models of differential drivers with pre-emphasis for the assessment of signal integrity and electromagnetic compatibility effects in multi-gigabit data transmission systems. A suitable model structure is derived and the procedure for its estimation from port transient waveforms is illustrated. The proposed methodology is an extension of the macromodeling based on parametric relations applied to plain differential drivers. The obtained models preserve the accuracy and efficiency strengths of behavioral parametric macromodels for conventional devices. A realistic application example involving a high-speed communication path and a 3.125 Gb/s commercial driver model with pre-emphasis is presented.

International conferences

I.S. Stievano, C. Siviero, I.A. Maio, F.G. Canavero, "Modeling of the Static and Dynamic Behavior of Differential Drivers," in Proc. of the *4th International Workshop on Electromagnetic Compatibility of Integrated Circuits (EMC-Compo)*, Angers (France), Mar. 31 – Apr. 1, 2004.

Abstract – The development of behavioral models of differential drivers for the simulation of signal integrity and electromagnetic compatibility problems is addressed. The obtained macromodels are readily included in any circuit simulation environment. A complete modeling example is given.

I.S. Stievano, C. Siviero, I.A. Maio, F.G. Canavero, "Behavioral Macromodels of Differential Drivers," in Proc. of the *8th IEEE Workshop on Signal Propagation on Interconnects (SPI)*, Heidelberg (Germany), pp. 131–134, May 9–12, 2004.

Abstract – This paper addresses the development of behavioral macromodels of differential drivers for the assessment of signal integrity and electromagnetic compatibility effects in high-speed digital systems. The obtained

macromodels are readily implemented as SPICE-like subcircuts to be included in any circuit simulation environment. Accuracy and efficiency of macromodels are assessed by applying the proposed methodology to actual differential devices.

I.S. Stievano, I.A. Maio, F.G. Canavero, C. Siviero, "Device Macromodel Impact on Data Link Performance Assessment," in Proc. of the *IEEE 13th Topical Meeting on Electrical Performance of Electronic Packaging (EPEP)*, Portland, OR (USA), pp. 239–242, Oct. 25–27, 2004.

Abstarct – The use of simulation and macromodels to assess the performance of a typical interconnecting system is addressed. A receiver macromodel including threshold decision of received signals is proposed, and an efficiency analysis demontrates that the use of well-designed macromodels for all parts of the transmission chain dramatically speeds up the simulation. Also, a careful discussion of simulated eye diagrams shows that macromodels guarantee timing accuracy even in very long bit sequences.

I.S. Stievano, I.A. Maio, F.G. Canavero, C. Siviero, "Behavioral Macromodels of Differential Drivers with Pre-Emphasis," in Proc. of the *9th IEEE Workshop on Signal Propagation on Interconnects (SPI)*, Garmisch-Partenkirchen (Germany), pp. 129–132, May 10–13, 2005.

Abstract – This paper addresses the extension of the behavioral modeling via parametric relations to differential drivers with pre-emphasis. These devices are of paramount importance to enable multi-gigabit data transmission over conventional copper interconnects. The proposed models preserve the accuracy and efficiency strenghts of behavioral parametric macromodels for conventional devices. Their operation is demonstrated in a realistic simulation example involving a 3.125 Gb/s commercial driver with pre-emphasis.

C. Siviero, I.S. Stievano, I.A. Maio, "Behavioral Modeling of IC Output Buffers: a Case Study," in Proc of the *2005 IEEE PhD Research in Microelectronics and Electronics (PRIME)*, Lausanne (Switzerland), pp. 366–369, Jul. 25–28, 2005.

Abstract – This paper addresses the behavioral modeling of IC output ports

by means of nonlinear parametric relations and system identification methods. The approach is applied to a commercial device and a systematic discussion of the impact of the modeling setup on the model performance is presented.

C. Siviero, P.M. Lavrador, J.C. Pedro, "A Frequency Domain Extraction Procedure of Low-Pass Equivalent Behavioral Models of Microwave PAs," in Proc. of the *2006 European Microwave Integrated Circuits Conference (EuMIC2006)*, Manchester (UK), pp. 253–256, Sep. 10–13, 2006.

Abstract – This paper reports on a novel extraction procedure of low-pass equivalent behavioral models of microwave power amplifiers. The proposed approach is based on the use of the discrete-time Volterra Series model and on the complex envelope of a multisine as stimulus. The model extraction procedure is formulated using an alternative orthogonal model representation in the frequency domain, providing a methodology which successfully leads to an optimal model identification not affected by the ill-conditioning of typical learning methods.

I.S. Stievano, C. Siviero, I.A. Maio, F.G. Canavero, "Guaranteed Locally-Stable Macromodels of Digital Devices via Echo State Networks," in Proc. of the *IEEE 15th Topical Meeting on Electrical Performance of Electronic Packaging (EPEP)*, Scottsdale, AR (USA), Oct. 23–25, 2006.

Abstract – The assessment of signal integrity effects in high-speed digital systems requires accurate and efficient IC macromodels. The proposed methodology is based on parametric relations that are expressed in terms of discrete-time Echo State Networks. This approach overcomes the stability limitations of traditional parametric macromodels used so far. Applications of Echo State Networks to the modeling of real devices exhibiting a complex dynamical behavior are discussed.

I. S. Stievano, C. Siviero, F. G. Canavero, "Composite Local-Linear State-Space Models for the Behavioral Modeling of Digital Devices," accepted for the *IEEE Instrumentation and Measurement Technology Conference*, Varsaw (Poland), May 1–3, 2007.

Abstract – This paper addresses the generation of IC macromodels for the assessment of signal integrity effects in high-speed digital systems. The proposed methodology is based on the estimation of composite local linear state-space models from device port responses. These models help to overcome some limitations of traditional parametric macromodels and are readily implemented in any simulation environment like SPICE or VHDL-AMS. The advocated approach is applied with success to the modeling of a real device exhibiting a strong nonlinear behavior and high order dynamical effects.