# Exact Search Directions for Optimization of Linear and Nonlinear Models Based on Generalized Orthonormal Functions

Alex da Rosa, Ricardo J. G. B. Campello, *Member, IEEE*, and Wagner C. Amaral, *Member, IEEE*

***Abstract*—A novel technique for selecting the poles of orthonormal basis functions (OBF) in Volterra models of any order is presented. It is well-known that the usual large number of parameters required to describe the Volterra kernels can be significantly reduced by representing each kernel using an appropriate basis of orthonormal functions. Such a representation results in the so-called OBF Volterra model, which has a Wiener structure consisting of a linear dynamic generated by the orthonormal basis followed by a nonlinear static mapping given by the Volterra polynomial series. Aiming at optimizing the poles that fully parameterize the orthonormal bases, the exact gradients of the outputs of the orthonormal filters with respect to their poles are computed analytically by using a back-propagation-through-time technique. The expressions relative to the Kautz basis and to generalized orthonormal bases of functions (GOBF) are addressed; the ones related to the Laguerre basis follow straightforwardly as a particular case. The main innovation here is that the dynamic nature of the OBF filters is fully considered in the gradient computations. These gradients provide exact search directions for optimizing the poles of a given orthonormal basis. Such search directions can, in turn, be used as part of an optimization procedure to locate the minimum of a cost-function that takes into account the error of estimation of the system output. The Levenberg-Marquardt algorithm is adopted here as the optimization procedure. Unlike previous related work, the proposed approach relies solely on input-output data measured from the system to be modeled, i.e., no information about the Volterra kernels is required. Examples are presented to illustrate the application of this approach to the modeling of dynamic systems, including a real magnetic levitation system with nonlinear oscillatory behavior.**

***Index Terms*—Back-propagation-through-time technique, generalized orthonormal bases of functions (GOBF), Kautz, Laguerre, linear and nonlinear systems identification, optimization, orthonormal basis functions (OBF), Volterra series.**

## I. Introduction

In recent years, there has been an increasing interest in the use of orthonormal basis functions (OBF) in studies involving the identification and control of dynamic processes [10], [15], [23], [24], [47], [48], [52]. The main reason for using OBF in such areas is that the corresponding approximate (modeling and control) problems usually have simpler solutions, since the orthonormality of these functions often yields simpler general models. One important issue involved in the use of an OBF-based model structure is the incorporation of approximate knowledge about the dominant dynamics of the system into the identification process [14], [21], [49]. This knowledge allows the number of free-design parameters of the model and, accordingly, the variance of their estimates, to be drastically reduced. The consequence is an increase in robustness and accuracy of the results. In what concerns robustness, another advantage of OBF models is that the OBF dynamics correspond to all-pass filters, which are robust for numerical implementation [13].

The Laguerre and Kautz bases [5], [19], [23] are the most commonly used OBF for the approximation of signals and systems. They are preferable for the modeling of systems having first- or second-order dominant dynamics, respectively. To model systems with more complex dominant dynamics, Generalized Orthonormal Basis Functions (GOBF)[1] [14], [15], [21], [22], [47] may be more appropriate, with the caveat that they involve more complex parameterizations.

Since the poles of the OBF are free-design parameters, their optimal selection constitutes an important stage of the model identification problem. When properly selected, an orthonormal series can increase the speed of convergence in problems of identification [14], [15]. In fact, if the parameterization of the basis is set close to the dominant modes of the system, then an accurate approximation can be obtained with fewer coefficients. For the Laguerre basis, analytical developments that lead to closed optimization solutions have been extensively addressed for both linear and nonlinear domains [6], [9], [11], [12], [17], [25], [27], [33], [34], [37], [40]. As observed in [31], a notorious advantage of the Laguerre basis is that its functions satisfy a suitable difference equation. Indeed, the Laguerre basis involves rational transfer functions with a simple recursive form and completely parameterized by a single real-valued pole (the Laguerre pole). For this reason, the Laguerre basis is preferable for representing well-damped dynamic systems.

Systems with poorly damped dynamics, however, typically cannot be accurately approximated with a small number of Laguerre functions. In other words, such functions are not very

Manuscript received April 17, 2008; revised September 23, 2008, February 10, 2009, and July 08, 2009. First published November 03, 2009; current version published December 09, 2009. This work was supported by the Brazilian National Council for Scientific and Technological Development (CNPq), under Grants 140706/2005-4, 306229/2006-4, and 301063/2007-9, and also by the Research Foundation of the State of São Paulo (Fapesp), under Grant 06/50231-5. Recommended by Associate Editor J.-F. Zhang.

A. da Rosa and W. C. Amaral are with the School of Electrical and Computer Engineering, University of Campinas (UNICAMP), Campinas-SP13083-852, Brazil (e-mails: alex@dca.fee.unicamp.br; wagner@dca.fee.unicamp.br).

R. J. G. B. Campello is with the Department of Computer Sciences, University of São Paulo (USP), São Carlos-SP 13560-970, Brazil (e-mail: campello@icmc.usp.br).

Digital Object Identifier 10.1109/TAC.2009.2031721

[1]Although the term Generalized Orthonormal Basis Functions is originally due to the formula by Heuberger and Van den Hof [14], [15], this term will be generically used hereafter to refer to orthonormal bases of functions with multiple modes.

well suited to approximate signals with strong oscillatory behavior [16], [26], [47]. This drawback has led to an increasing interest in the two-parameter Kautz functions, introduced in [3]. These functions can better approximate underdamped systems for being parameterized by a pair of resonant poles. Optimality conditions for the error between the impulse response of a given linear system and its Kautz approximation were derived in [16], [18]. In the context of pole location, a sub-optimal analytical choice of Kautz poles for discrete-time linear systems was proposed in [31], and the corresponding nonlinear counterpart was later addressed for Volterra models in [35], [41]. More recently [42], an analytical solution for one of the parameters related to the Kautz poles was derived when any-order Volterra kernels are decomposed into a set of independent orthonormal bases, each of which is parameterized by an individual pair of conjugate Kautz poles associated with the dominant dynamic of the kernel along a particular dimension. This is an extension of the findings in [35] and [41], in which the solution involves a single Kautz basis for expanding a given kernel along all its dimensions.

Despite the important theoretical results and advances achieved in regard to analytical approaches to the optimization of the poles of orthonormal bases, the solutions so far derived suffer from at least one of the following drawbacks: (i) a model of the system (e.g. FIR or Volterra) must be known in advance. For obvious reasons, this is often a hard requirement in practice, especially in the nonlinear case; (ii) the solution is only sub-optimal, for it minimizes an upper bound of the approximation error and/or for it optimizes only a subset of the basis parameters; and (iii) the solution is restricted to the Laguerre or Kautz bases. As far as the authors know, no analytical solution for optimization of GOBF has been derived so far. The reason is probably the structural complexity of the generalized bases, whose functions have independent parameterizations.

In contrast to the analytical methods, a different approach to the problem of pole location can be adopted that utilizes numerical procedures for optimization. For the reasons just described, numerical procedures are mandatory when optimizing GOBF or when no model of the system is known in advance. In this context, numerical procedures for selecting OBF poles in an iterative manner have been proposed [20], [28]–[30], [32], [36], [38], [39]. However, such iterative procedures cannot guarantee either optimality, convergence, or both. A general optimization formulation that conceptually embodies both the optimality and convergence requirements has been suggested in [39], but no strategy has been provided for determining the search directions to be followed by the optimization algorithm. The main difficulty comes from the fact that the relations between the model output and the OBF parameters (poles) are governed by dynamic equations. The use of the gradient descent technique was proposed in [28], but the gradient formulation and the corresponding search directions were roughly approximated by means of instantaneous (static) estimates of their partial derivatives. The present paper provides a method for the precise determination of search directions based on the analytical recursive computation of the derivatives of the output of the orthonormal basis filters with respect to their poles (back-propagation-through-time technique). Such derivatives can then be used as part of an optimization method to obtain exact search directions for the OBF poles that fully encompass the dynamic nature of these parameters. The Levenberg–Marquardt method [2], [4], [49] is adopted in this paper, since it can significantly outperform gradient descent and conjugate gradient methods for medium sized problems and is a usual choice in nonlinear optimization.

The remainder of the paper is organized as follows. In the next section, orthonormal basis functions are reviewed in the context of OBF Volterra models. In Section III, the problem of optimizing the poles of orthonormal bases in the description of linear systems (first-order Volterra models) is formulated and the main mathematical foundations for numerically solving it are furnished. In Section IV, the detailed formulae for the Kautz basis are provided. In Section V, the formulation for GOBF is investigated; Laguerre follows straightforwardly as a particular case. In Section VI, the formulations are extended to any-order Volterra models. In Section VII, a simulation example is presented and the results are compared with those from equivalent experiments reported in the literature. In Section VIII, an application to the modeling of a real magnetic levitation system with nonlinear oscillatory behavior is described. Finally, Section IX addresses the conclusions.

## II. Approximation of Volterra Models Using Orthonormal Functions

A Volterra model is essentially an input-output functional (polynomial) expansion of a nonlinear system whose structure is given by a straightforward generalization of the unit-impulse response model [46], [49]. The absence of output recursion in these models, their direct link to the nonlinear realization theory, and their ability to represent a wide class of nonlinear systems, among other desirable properties, make them very attractive. In the discrete-time domain, the mathematical description of a Volterra model relates the output $y(k)$ of a physical process to its input $u(k)$ as [51], [52]

$$y(k) = \sum_{\eta=1}^{\infty} \sum_{\tau_1=0}^{\infty} \cdots \sum_{\tau_\eta=0}^{\infty} h_\eta(\tau_1, \ldots, \tau_\eta) \prod_{l=1}^{\eta} u(k-\tau_l) \quad (1)$$

where the multidimensional functions $h_\eta(\tau_1, \ldots, \tau_\eta)$ are the $\eta$th-order Volterra kernels. Although these models can describe a wide class of nonlinear systems, their practical use is limited due to the usually large number of coefficients that are needed to be estimated, even for simple problems. Such a drawback can be avoided by expanding the Volterra kernels using orthonormal bases of functions. The number of parameters necessary to represent the models can thus be drastically reduced if properly designed bases of functions are adopted.

Approximating a given dynamic system by means of a truncated orthonormal series was first suggested by Wiener [53]. A representation of Wiener type [49], [51] consists of a linear dynamic, here composed of a set of orthonormal filters, followed by a nonlinear static mapping, here represented by the Volterra series. The basic idea of such OBF Volterra models is to describe the Volterra kernels $h_\eta$ by means of an expansion using OBF in such a way that one needs to determine the coefficients

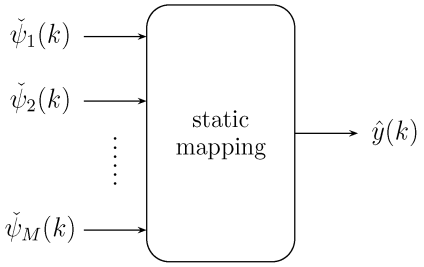



Fig. 1. General OBF model: static (linear or nonlinear) mapping over the outputs of a set of $M$ orthonormal filters.

of this expansion in lieu of the coefficients of the kernels. Formally, the kernels $h_\eta$ can be mathematically described using an orthonormal basis of functions $\{\psi_m\}$ as [52]

$$h_\eta(k_1,\ldots,k_\eta)=\sum_{i_1=1}^{\infty}\cdots\sum_{i_\eta=1}^{\infty}\alpha_{i_1,\ldots,i_\eta}\prod_{l=1}^{\eta}\psi_{i_l}(k_l) \quad (2)$$

which assumes that the kernels are absolutely summable on $[0,\infty)$. In practice, this condition can be assured by forcing the long memory terms of the kernels to be null, which is possible provided that the system to be modeled is stable. In other words, $h_\eta(k_1,\ldots,k_\eta)$ is assumed to be zero for $k_l>\epsilon,\forall l\in\{1,\ldots,\eta\}$. An appropriate value for $\epsilon<\infty$ can be set based on the saddle or rise time of the system.

If the input signal $u(k)$ in (1) is bounded so that $|u(k)|<1\,\forall k$, then the higher-order kernels can be ignored in such a way that the resulting Volterra model is truncated to a finite-order $N$ [46]. Furthermore, for computational reasons, (2) is, in practice, approximated with a finite number $M$ of functions, as follows:

$$\hat{h}_\eta(k_1,\ldots,k_\eta)=\sum_{i_1=1}^{M}\cdots\sum_{i_\eta=1}^{M}\alpha_{i_1,\ldots,i_\eta}\prod_{l=1}^{\eta}\psi_{i_l}(k_l). \quad (3)$$

Hence, if it is assumed that $u(k)=0$ for $k<0$, then (1) can be approximated as

$$\hat{y}(k)=\sum_{\eta=1}^{N}\left[\sum_{i_1=1}^{M}\cdots\sum_{i_\eta=1}^{M}\alpha_{i_1,\ldots,i_\eta}\prod_{l=1}^{\eta}\left(\sum_{\tau_l=0}^{k}\psi_{i_l}(\tau_l)u(k-\tau_l)\right)\right]. \quad (4)$$

In [8], it is shown that truncated Volterra models, such as those in (4), can approximate to desired accuracy any time-invariant analytical nonlinear system with fading memory and bounded input. Indeed, any prescribed approximation accuracy can be obtained by setting the number of kernels, $N$, and functions, $M$, to appropriate finite values.

A schematic description of OBF Volterra models is shown in Fig. 1, where $\breve{\psi}_m(k)$ denotes the response—to the input $u(k)$—of the filter with impulse response $\psi_m(k)$, i.e.

$$\breve{\psi}_m(k)=\sum_{\tau=0}^{k}\psi_m(\tau)u(k-\tau) \quad (5)$$

$$\breve{\Psi}_m(z)=\Psi_m(z)U(z). \quad (6)$$

Since model (4) is linear in the parameters $\alpha_{(.)}$, these parameters can be straightforwardly computed using a least-squares algorithm. Clearly, the overall number of parameters depends upon the number of functions, $M$, used in the kernels expansions. This quantity represents a trade-off between accuracy and parsimony of the model and can be significantly reduced by optimally designing the set of orthonormal filters $\{\psi_m\}$. However, the actual value for $M$ required to provide an accurate approximation also depends on the complexity of the specific problem in hand. Dynamic systems with multiple dominant modes, for example, typically require models with a larger number of functions.

*Remark 1:* Note that the OBF Volterra model in (4) implicitly assumes that all its kernels are expanded using the same orthonormal basis $\{\psi_m\}$. Actually, each $\eta$th order kernel can be expanded using an independent basis of functions $\{\psi_{\eta,m}\}$ (e.g. see [33], [52]), as will be shown in the experiments reported in Sections VII and VIII. For the sake of simplicity and without any loss of generality, however, most of the theoretical developments in this paper will be presented following the same representation adopted here in this section, i.e., assuming that all Volterra kernels are expanded using the same orthonormal basis $\{\psi_m\}$.

### A. Orthonormal Basis Functions

The use of orthonormal filters for representing signals and systems has a long history, since the pioneering proposals by Takenaka [1], Wiener [53], and others. The problem of building a set of orthonormal continuous functions was presented in [3], whereas the corresponding discrete case was solved in [5]. Discrete-time orthonormal basis functions can be generated by cascading different all-pass filters of order one or two, as follows [21], [47]:

$$\Psi_m(z)=\frac{z\sqrt{1-|\beta_m|^2}}{z-\beta_m}\prod_{j=1}^{m-1}\left(\frac{1-\bar{\beta}_j z}{z-\beta_j}\right)\quad m=1,2,\ldots \quad (7)$$

where $\beta_m$ are the stable poles of the orthonormal basis ($\beta_m\in\mathbb{C}:|\beta_m|<1$) and $\bar{\beta}_m$ denotes the complex conjugate of $\beta_m$. The functions in (7) are the so-called Takenaka-Malmquist functions [47]. The corresponding realizations in time-domain, $\psi_m(k)$, are given by the inverse $Z$-transform of (7) and satisfy the orthonormality property. The set $\{\psi_m\}$ is complete on $\ell^2[0,\infty)$ if and only if $\sum_{m=1}^{\infty}(1-|\beta_m|)=\infty$ [21], [47], so any finite energy signal (including absolutely summable kernels) can be approximated with any prescribed accuracy by linearly combining a certain finite number of such functions. In general, functions $\psi_m(k)$ are complex-valued, although this is physically unrealistic in system identification problems. It is shown in [21] that this drawback can be circumvented by constructing a modified basis of functions with real-valued impulse responses, consisting of a linear combination of the complex-valued functions generated by (7).

When all the poles of (7) are real-valued and equal to each other, i.e. $\beta_m=\bar{\beta}_m=c\,(\forall m)$, one gets the Laguerre basis, which can be written in the $z$-domain as [11], [12], [17]

$$\Psi_m(z)=\frac{z\sqrt{1-c^2}}{z-c}\left(\frac{1-cz}{z-c}\right)^{m-1} \quad (8)$$

with $c$ denoting the Laguerre pole. By setting $c = 0$, the Laguerre functions simplify to an ordinary Pulse Basis $\Psi_m(z) = z^{-(m-1)}$ and model (4) reduces to the ordinary Nonlinear Finite Impulse Response (NFIR) Volterra model, i.e., a truncated version of (1).

Another important OBF realization, which has also been shown to be a particular case of a unifying construction for (7) [21], is obtained by cascading an all-pass filter with pole at $\beta$ and an all-pass filter with pole at $\bar{\beta}$, in such a way that the pairs of conjugate poles are equal to each other for any value of $m$, i.e. $\{\beta, \bar{\beta}, \beta, \bar{\beta}, \ldots\}$. The result is the well-known Kautz basis, whose functions are defined in the $z$-domain as [13], [31], [47]

$$\begin{aligned}\Psi_{2m}(z) &= \frac{z\sqrt{(1-c^2)(1-b^2)}}{z^2+b(c-1)z-c} \\ &\quad\times\left[\frac{-cz^2+b(c-1)z+1}{z^2+b(c-1)z-c}\right]^{m-1} \\ \Psi_{2m-1}(z) &= \frac{z(z-b)\sqrt{1-c^2}}{z^2+b(c-1)z-c} \\ &\quad\times\left[\frac{-cz^2+b(c-1)z+1}{z^2+b(c-1)z-c}\right]^{m-1}\end{aligned} \tag{9}$$

with $\Psi_{2m}(z)$ and $\Psi_{2m-1}(z)$ denoting the even and odd Kautz functions, respectively. Scalars $b$ and $c$ are real-valued parameters satisfying $|b| < 1$ and $|c| < 1$. These parameters are related to the pair of Kautz poles $(\beta, \bar{\beta})$ as

$$b = (\beta + \bar{\beta})/(1 + \beta\bar{\beta}) \tag{10}$$

$$c = -\beta\bar{\beta}. \tag{11}$$

Expressions analogous to (9) can be found, e.g., in [16], [18].

## III. Problem Statement

This section elaborates on the problem of adjusting those parameters of OBF models that relate dynamically to the model output. First, the back-propagation-through-time technique is briefly reviewed [49], whose basic idea is to unfold the dynamics of a given model in a recursive way until the model is only described by static relations. In order to make the basic idea easier to understand, the technique is initially presented in the context of linear systems. The optimization of the parameters of OBF models based on this technique is also initially discussed for the linear case in this section. The generalization to any-order Volterra models is addressed in Section VI.

### A. Back-Propagation-Through-Time Technique

Consider a linear dynamic system represented by a static mapping $\mathcal{H}$ over a set of input and output regression terms

$$\hat{y}(k) = \mathcal{H}\left[u(k-1), \ldots, u(k-n_u), \hat{y}(k-1), \ldots, \hat{y}(k-n_y)\right] \tag{12}$$

where $\hat{y}(k)$ denotes the model output. In this sort of model, so-called Output-Error (OE) model [49], the static mapping $\mathcal{H}$ relates dynamically to the model output, since it is implicitly and recursively included into terms $\hat{y}(k-i)$. Hence, the parameters of mapping $\mathcal{H}$ cannot be directly estimated from data using a standard estimation approach, like a least squares algorithm. Instead, the correct approach consists of representing the dynamics of the system by means of several static models unfolded through explicit time-recursions. In order to illustrate this, let's consider, for instance, model (12) with $n_u = n_y = 1$, i.e.

$$\hat{y}(k) = \mathcal{H}\left[u(k-1), \hat{y}(k-1)\right]. \tag{13}$$

Then, let the static mapping $\mathcal{H}$ be such that it can be parameterized by a set of free-design parameters, namely, the parameter vector $\boldsymbol{\theta}$ of the model. This vector can be optimized if the gradient of a cost-function with respect to it is available. Such a gradient depends on the derivatives of the model output with respect to the elements of $\boldsymbol{\theta}$. Particularly, provided that the model is linear, one can infer from (13) that

$$\frac{\partial\hat{y}(k)}{\partial\theta_i} = \frac{\partial\mathcal{H}_1[\cdot]}{\partial\theta_i} + \frac{\partial\mathcal{H}_2[\cdot]}{\partial\hat{y}(k-1)} \cdot \frac{\partial\hat{y}(k-1)}{\partial\theta_i} \tag{14}$$

where $\mathcal{H}_2$ and $\mathcal{H}_1$ are the complementary portions of $\mathcal{H}$ that depends and does not depend on $\hat{y}(k-1)$, respectively, and $\theta_i$ is the $i$th element of the parameter vector $\boldsymbol{\theta}$. The first term of (14) represents the static aspect of that equation. The second term arises from the recursive component and represents the dynamic aspect of the equation. Note that, if $\hat{y}(k-1)$ was an exogenous signal, such as the measured process output $y(k-1)$, there would be no dependence of this signal on the model parameters and the second term of (14) would be null. In OE models, however, $\hat{y}(k-1)$ is the output of the model itself in a previous time instant, which does depend on $\boldsymbol{\theta}$. Thus, the second term of (14) requires the derivatives of the model output at the previous time instant with respect to the model parameters. These derivatives can be calculated from $\hat{y}(k-1) = \mathcal{H}[u(k-2), \hat{y}(k-2)]$ as

$$\frac{\partial\hat{y}(k-1)}{\partial\theta_i} = \frac{\partial\mathcal{H}_1[\cdot]}{\partial\theta_i} + \frac{\partial\mathcal{H}_2[\cdot]}{\partial\hat{y}(k-2)} \cdot \frac{\partial\hat{y}(k-2)}{\partial\theta_i}. \tag{15}$$

Again, the second term in (15) requires the derivatives of the output at the preceding time instant. This procedure continues until the initial conditions at $k = 0$ are reached; at this time, $\hat{y}(0) = y(0)$ and, accordingly, $\partial\hat{y}(0)/\partial\theta_i = 0$.

In summary, the back-propagation-through-time technique decomposes the dynamics of a system into a series of static representations. This approach allows describing the derivatives of the model output in terms of the initial conditions and the input signal only, by backtracking $k$ steps through time. This technique is illustrated in Fig. 2.

### B. Optimization of OBF Poles

Consider initially a first-order OBF Volterra model (linear OBF model). In this case, the static mapping $\mathcal{H}$ in Fig. 1 is a linear function, and model (4) can be rewritten as

$$\hat{y}(k) = \sum_{m=1}^{M} \alpha_m \breve{\psi}_m(k) \tag{16}$$

where $\breve{\psi}_m(k)$ is defined as (5). The strategy adopted here consists essentially of the optimization of both the pole vector $\mathbf{p}$

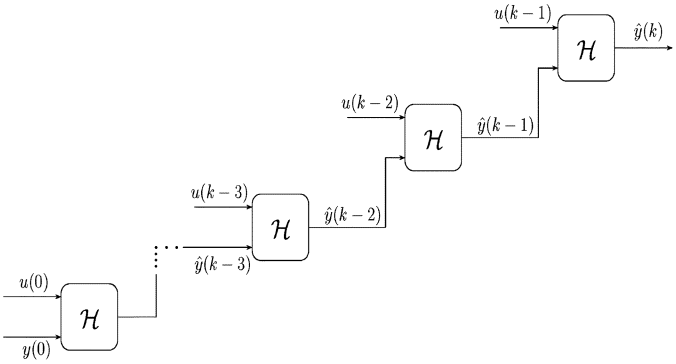



Fig. 2. Back-propagation-through-time technique.

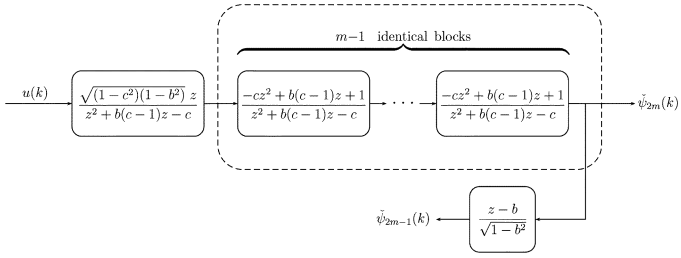



Fig. 3. Diagram of the dynamic portion of a Kautz model: OBF filters for $m = 1, 2, \ldots$.

that parameterizes the orthonormal basis $\{\psi_m\}$ and the corresponding coefficient vector $\boldsymbol{\alpha} \triangleq [\alpha_1\ \alpha_2\ \cdots\ \alpha_M]^T$. To do so, one defines the following optimization problem:

$$\min_{\boldsymbol{\theta}}\ J \triangleq \frac{1}{2}\sum_{k=1}^{N_d}[y(k) - \hat{y}(k)]^2 \tag{17}$$

where $\boldsymbol{\theta} \triangleq [\mathbf{p}^T\ \boldsymbol{\alpha}^T]^T$ and $N_d$ is the number of input-output (I/O) data samples to be used in the optimization procedure. The gradients of functional $J$ in (17) with respect to vectors $\mathbf{p}$ and $\boldsymbol{\alpha}$ are readily obtained from (16) as

$$\begin{aligned}\nabla_{\mathbf{p}}J &= \sum_{k=1}^{N_d}[\hat{y}(k) - y(k)]\,\nabla_{\mathbf{p}}\hat{y}(k)\\ &= \sum_{k=1}^{N_d}\sum_{m=1}^{M}[\hat{y}(k) - y(k)]\,\alpha_m\nabla_{\mathbf{p}}\check{\psi}_m(k) \qquad (18)\\ \nabla_{\boldsymbol{\alpha}}J &= \sum_{k=1}^{N_d}[\hat{y}(k) - y(k)]\,\nabla_{\boldsymbol{\alpha}}\hat{y}(k)\\ &= \sum_{k=1}^{N_d}[\hat{y}(k) - y(k)]\,\check{\boldsymbol{\psi}}(k) \qquad (19)\end{aligned}$$

where $\check{\boldsymbol{\psi}}(k) \triangleq [\check{\psi}_1(k)\ \check{\psi}_2(k)\ \cdots\ \check{\psi}_M(k)]^T$.

The central contribution of the present paper is the derivation of detailed analytical formulae for computing $\nabla_{\mathbf{p}}\check{\psi}_m(k)$ and their application to optimizing the poles of model (16) using a gradient-based optimization algorithm. In particular, the well-known Levenberg-Marquardt algorithm [2], [4], [49] is suggested and adopted here. The basis of this algorithm is a quadratic approximation of $J$ within a small neighborhood of its arguments (model parameters). The algorithm approximates the Hessian (i.e. matrix of mixed partials) of $J$ with respect to the parameters using only the information of the corresponding first-order derivatives (gradients). Specifically, it estimates the Hessian matrix using the sum of outer products of the gradients in (18) and (19). Denoting $F(\boldsymbol{\theta})$ as the Jacobian of the cost-function $J(\boldsymbol{\theta})$, the algorithm updates the parameter vector $\boldsymbol{\theta}$ according to the following equation [49]:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \left[F_{t-1}(\boldsymbol{\theta})F_{t-1}^T(\boldsymbol{\theta}) + \sigma_{t-1}I\right]^{-1}F_{t-1}(\boldsymbol{\theta})J_{t-1}(\boldsymbol{\theta}) \tag{20}$$

where $\sigma_{t-1}$ is a regularization factor and $I$ is the identity matrix.

The Levenberg-Marquardt algorithm is a pseudo-second order method, which means that it relies on function evaluations and gradient information only. This algorithm is globally convergent, that is, it converges from any starting point $\boldsymbol{\theta}^0$ towards a point $\boldsymbol{\theta}_{\text{opt}}$ satisfying necessary optimality conditions for a local minimizer of $J(\boldsymbol{\theta})$, i.e. $\nabla_{\boldsymbol{\theta}}J = \mathbf{0}$. Of course, it is not possible to ensure that $\boldsymbol{\theta}_{\text{opt}}$ is a global minimizer of $J(\boldsymbol{\theta})$ unless $J(\boldsymbol{\theta})$ is a convex function [50]. The Levenberg-Marquardt algorithm can be thought of as a combination of the steepest descent and Gauss-Newton methods [49]: when the current solution is far away from the optimum, the algorithm behaves like a steepest descent method (slow, but guaranteed to converge); close to the optimum, it becomes a Gauss-Newton method, thus converging faster.

Section IV presents a study when the Kautz basis $\{\psi_m\}$ is used to implement the OBF model in (5) and (16).

## IV. Two-Parameter Kautz Formulation

In this section, a scheme like that in Fig. 2 is used to derive expressions that describe the outputs of the OBF filters in terms of both their initial conditions and the input signal. These expressions are then used as a basis for deriving the gradients of the filter outputs with respect to the OBF parameters, i.e., $\nabla_{\mathbf{p}}\check{\psi}_m(k)$ in (18). Initially, the two-parameter Kautz functions are investigated. The generalization to GOBF is addressed in Section V.

### A. The Even Kautz Functions

According to the definition of the Kautz functions in (9), a block diagram of the filters that compose the dynamics of a Kautz model can be constructed, as shown in Fig. 3.

Note that the output of the first even Kautz filter $\check{\Psi}_2(z)(m = 1)$ can be written in the $z$-domain as follows:

$$\check{\Psi}_2(z) = \frac{\sqrt{(1-c^2)(1-b^2)}z}{z^2 + b(c-1)z - c}U(z). \tag{21}$$

In the time-domain, one gets

$$\begin{aligned}\check{\psi}_2(k+2) + b(c-1)\check{\psi}_2(k+1) - c\check{\psi}_2(k)\\ = \sqrt{(1-c^2)(1-b^2)}u(k+1). \qquad (22)\end{aligned}$$

An approach to obtain an expression for $\check{\psi}_2(k)$ in terms of both the initial conditions ($\check{\psi}_2(0)$ and $\check{\psi}_2(1)$) and the input signal $u(k)$ consists of representing (22) in a state-space form. By defining $x_a(k) \triangleq \check{\psi}_2(k)$ and $x_b(k) \triangleq x_a(k+1) = \check{\psi}_2(k+1)$,

the difference (22) can be rewritten as the following state-space realization:

$$\begin{cases} x_a(k+1) = x_b(k) \\ x_b(k+1) = -b(c-1)x_b(k) + cx_a(k) \\ \quad +\sqrt{(1-c^2)(1-b^2)}u(k+1) \end{cases} \tag{23}$$

or equivalently, in the matricial form, as

$$\begin{cases} \begin{bmatrix} x_a(k+1) \\ x_b(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ c & -b(c-1) \end{bmatrix} \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix} \\ \quad + \begin{bmatrix} 0 \\ \sqrt{(1-c^2)(1-b^2)} \end{bmatrix} u(k+1) \\ \check{\psi}_2(k) = [1 \quad 0] \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix}. \end{cases} \tag{24}$$

This model corresponds to the following representation:

$$\begin{cases} \mathbf{x}_2(k+1) = \mathcal{A}\mathbf{x}_2(k) + \mathcal{B}u(k+1) \\ \check{\psi}_2(k) = \mathcal{C}\mathbf{x}_2(k) \end{cases} \tag{25}$$

whose solution for $\check{\psi}_2(k)$ is readily given by

$$\check{\psi}_2(k) = \mathcal{C}\mathcal{A}^k\mathbf{x}_2(0) + \sum_{i=0}^{k-1} \mathcal{C}\mathcal{A}^i\mathcal{B}u(k-i) \tag{26}$$

where $\mathbf{x}_2(0) = [\check{\psi}_2(0)\ \check{\psi}_2(1)]^T$ and the state matrices are given by

$$\mathcal{A} = \begin{bmatrix} 0 & 1 \\ c & -b(c-1) \end{bmatrix} \tag{27}$$

$$\mathcal{B} = \begin{bmatrix} 0 \\ \sqrt{(1-c^2)(1-b^2)} \end{bmatrix} \tag{28}$$

$$\mathcal{C} = [1 \quad 0]. \tag{29}$$

The gradient $\nabla_{\mathbf{p}}\check{\psi}_m(k)$ in (18) is then computed for $m = 1$ in terms of the derivatives of (26) with respect to parameters $b$ and $c$ that fully parameterize the Kautz basis, i.e. $\nabla_{\mathbf{p}}\check{\psi}_2(k) = [\partial\check{\psi}_2(k)/\partial b\ \partial\check{\psi}_2(k)/\partial c]^T$. These derivatives are given by

$$\begin{aligned} \frac{\partial\check{\psi}_2(k)}{\partial b} &= \mathcal{C}\frac{\partial(\mathcal{A}^k)}{\partial b}\mathbf{x}_2(0) \\ &\quad + \mathcal{C}\sum_{i=0}^{k-1}\left[\frac{\partial(\mathcal{A}^i)}{\partial b}\mathcal{B} + \mathcal{A}^i\frac{\partial\mathcal{B}}{\partial b}\right]u(k-i) \\ \frac{\partial\check{\psi}_2(k)}{\partial c} &= \mathcal{C}\frac{\partial(\mathcal{A}^k)}{\partial c}\mathbf{x}_2(0) \\ &\quad + \mathcal{C}\sum_{i=0}^{k-1}\left[\frac{\partial(\mathcal{A}^i)}{\partial c}\mathcal{B} + \mathcal{A}^i\frac{\partial\mathcal{B}}{\partial c}\right]u(k-i) \end{aligned} \tag{30}$$

where matrices $\partial(\mathcal{A}^k)/\partial b$ and $\partial(\mathcal{A}^k)/\partial c$ can be computed as

$$\frac{\partial(\mathcal{A}^k)}{\partial b} = \sum_{j=1}^{k}\mathcal{A}^{j-1}\frac{\partial\mathcal{A}}{\partial b}\mathcal{A}^{k-j} \tag{31}$$

$$\frac{\partial(\mathcal{A}^k)}{\partial c} = \sum_{j=1}^{k}\mathcal{A}^{j-1}\frac{\partial\mathcal{A}}{\partial c}\mathcal{A}^{k-j}. \tag{32}$$

So, in order to compute the gradient composed of the derivatives in (30), it is necessary to compute matrix $\mathcal{A}^k$. This computation can be much more efficiently performed by representing $\mathcal{A}$ in its eigenvectors basis. In this context, it is not difficult do deduce from (27), (10), and (11) that matrix $\mathcal{A}$ has eigenvectors $[1\ \beta]^T$ and $[1\ \bar{\beta}]^T$, with the corresponding eigenvalues equal to $\beta$ and $\bar{\beta}$, respectively. Hence, $\mathcal{A}$ can be rewritten as $\mathcal{A} = Q\Lambda Q^{-1}$, where $Q$ and $\Lambda$ are given by

$$Q = \begin{bmatrix} 1 & 1 \\ \beta & \bar{\beta} \end{bmatrix} \qquad \Lambda = \begin{bmatrix} \beta & 0 \\ 0 & \bar{\beta} \end{bmatrix}. \tag{33}$$

Based on these observations, it follows that:

$$\mathcal{A}^k = \frac{1}{\bar{\beta}-\beta}\begin{bmatrix} \bar{\beta}\beta^k - \beta\bar{\beta}^k & \bar{\beta}^k - \beta^k \\ \bar{\beta}\beta^{k+1} - \beta\bar{\beta}^{k+1} & \bar{\beta}^{k+1} - \beta^{k+1} \end{bmatrix} \tag{34}$$

when $\bar{\beta} \neq \beta$. Contrarily, if $\bar{\beta} = \beta$, then $\mathcal{A}^k$ becomes

$$\mathcal{A}^k = \begin{bmatrix} (1-k)\beta^k & k\beta^{k-1} \\ -k\beta^{k+1} & (k+1)\beta^k \end{bmatrix}. \tag{35}$$

Matrix $\mathcal{A}^k$, however, is not the only one that is needed for computing the derivatives in (30). Instead, matrices $\partial\mathcal{A}/\partial b$, $\partial\mathcal{A}/\partial c$, $\partial\mathcal{B}/\partial b$, and $\partial\mathcal{B}/\partial c$ are also required. These matrices are derived from (27) and (28) as

$$\frac{\partial\mathcal{A}}{\partial b} = \begin{bmatrix} 0 & 0 \\ 0 & 1-c \end{bmatrix} \qquad \frac{\partial\mathcal{A}}{\partial c} = \begin{bmatrix} 0 & 0 \\ 1 & -b \end{bmatrix}$$

$$\frac{\partial\mathcal{B}}{\partial b} = \begin{bmatrix} 0 \\ \frac{-b\sqrt{1-c^2}}{\sqrt{1-b^2}} \end{bmatrix} \qquad \frac{\partial\mathcal{B}}{\partial c} = \begin{bmatrix} 0 \\ \frac{-c\sqrt{1-b^2}}{\sqrt{1-c^2}} \end{bmatrix}.$$

Then, the gradient $\nabla_{\mathbf{p}}\check{\psi}_2(k) = [\partial\check{\psi}_2(k)/\partial b\ \partial\check{\psi}_2(k)/\partial c]^T$ that is composed of the derivatives in (30) is now completely determined.

Equation (30) represents the derivatives of the first even Kautz function $\check{\psi}_2(k)$ with respect to the Kautz parameters $b$ and $c$, where $\check{\psi}_2(k)$ is the even Kautz function in (9) for $m = 1$. Expressions for the Kautz filters with $m \geq 2$ can equivalently be obtained. According to the block diagram shown in Fig. 3, one can write

$$\check{\Psi}_{2m}(z) = \frac{-cz^2 + b(c-1)z + 1}{z^2 + b(c-1)z - c}\check{\Psi}_{2(m-1)}(z) \tag{36}$$

or in the time-domain

$$\begin{aligned} \check{\psi}_{2m}(k+2) + b(c-1)\check{\psi}_{2m}(k+1) - c\check{\psi}_{2m}(k) &= -c\check{\psi}_{2(m-1)}(k+2) \\ &\quad + b(c-1)\check{\psi}_{2(m-1)}(k+1) + \check{\psi}_{2(m-1)}(k). \end{aligned} \tag{37}$$

By using a reasoning analogous to that above for $\check{\psi}_2(k)$, it is possible to rewrite model (37) as the following state-space realization:

$$\begin{cases} \mathbf{x}_{2m}(k+1) = \mathcal{A}\mathbf{x}_{2m}(k) + \mathcal{B}_1\check{\psi}_{2(m-1)}(k+2) \\ \quad + \mathcal{B}_2\check{\psi}_{2(m-1)}(k+1) + \mathcal{B}_3\check{\psi}_{2(m-1)}(k) \\ \check{\psi}_{2m}(k) = \mathcal{C}\mathbf{x}_{2m}(k) \end{cases} \tag{38}$$

with $\mathbf{x}_{2m}(0) = [\check{\psi}_{2m}(0)\ \check{\psi}_{2m}(1)]^T$, $\mathcal{A}$ and $\mathcal{C}$ as in (27) and (29), respectively, and $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ given by

$$\mathcal{B}_1 = \begin{bmatrix} 0 \\ -c \end{bmatrix} \qquad \mathcal{B}_2 = \begin{bmatrix} 0 \\ b(c-1) \end{bmatrix} \qquad \mathcal{B}_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{39}$$

The solution of (38) for $\check{\psi}_{2m}(k)$ is computed as

$$\check{\psi}_{2m}(k) = \mathcal{C}\mathcal{A}^k \mathbf{x}_{2m}(0) + \mathcal{C}\sum_{i=0}^{k-1} \mathcal{A}^i \left[\mathcal{B}_1 \check{\psi}_{2(m-1)}(k+1-i) + \mathcal{B}_2 \check{\psi}_{2(m-1)}(k-i) + \mathcal{B}_3 \check{\psi}_{2(m-1)}(k-1-i)\right]. \quad (40)$$

Taking the derivatives of (40) with respect to parameters $b$ and $c$ yields

$$\begin{aligned}
\frac{\partial \check{\psi}_{2m}(k)}{\partial b} &= \mathcal{C}\frac{\partial(\mathcal{A}^k)}{\partial b}\mathbf{x}_{2m}(0) \\
&\quad + \mathcal{C}\sum_{i=0}^{k-1}\Bigg[\mathcal{A}^i\Bigg(\mathcal{B}_1\frac{\partial}{\partial b}\check{\psi}_{2(m-1)}(k+1-i) + \frac{\partial \mathcal{B}_2}{\partial b}\check{\psi}_{2(m-1)}(k-i) + \mathcal{B}_2\frac{\partial}{\partial b}\check{\psi}_{2(m-1)}(k-i) + \mathcal{B}_3\frac{\partial}{\partial b}\check{\psi}_{2(m-1)}(k-1-i)\Bigg) \\
&\quad + \frac{\partial(\mathcal{A}^i)}{\partial b}\left(\mathcal{B}_1\check{\psi}_{2(m-1)}(k+1-i) + \mathcal{B}_2\check{\psi}_{2(m-1)}(k-i) + \mathcal{B}_3\check{\psi}_{2(m-1)}(k-1-i)\right)\Bigg] \\
\frac{\partial \check{\psi}_{2m}(k)}{\partial c} &= \mathcal{C}\frac{\partial(\mathcal{A}^k)}{\partial c}\mathbf{x}_{2m}(0) \\
&\quad + \mathcal{C}\sum_{i=0}^{k-1}\Bigg[\mathcal{A}^i\Bigg(\frac{\partial \mathcal{B}_1}{\partial c}\check{\psi}_{2(m-1)}(k+1-i) + \mathcal{B}_1\frac{\partial}{\partial c}\check{\psi}_{2(m-1)}(k+1-i) + \frac{\partial \mathcal{B}_2}{\partial c}\check{\psi}_{2(m-1)}(k-i) + \mathcal{B}_2\frac{\partial}{\partial c}\check{\psi}_{2(m-1)}(k-i) + \mathcal{B}_3\frac{\partial}{\partial c}\check{\psi}_{2(m-1)}(k-1-i)\Bigg) \\
&\quad + \frac{\partial(\mathcal{A}^i)}{\partial c}\left(\mathcal{B}_1\check{\psi}_{2(m-1)}(k+1-i) + \mathcal{B}_2\check{\psi}_{2(m-1)}(k-i) + \mathcal{B}_3\check{\psi}_{2(m-1)}(k-1-i)\right)\Bigg]. \quad (41)
\end{aligned}$$

The terms $\partial(\mathcal{A}^i)/\partial b$, $\partial(\mathcal{A}^i)/\partial c$, and $\mathcal{A}^i$ in (41) are given by (31), (32), and (34), (35), respectively, and the derivatives $\partial\mathcal{B}_1/\partial c$, $\partial\mathcal{B}_2/\partial b$, and $\partial\mathcal{B}_2/\partial c$ can be readily computed from (39) as $\partial\mathcal{B}_1/\partial c = [0 \ \ -1]^T$, $\partial\mathcal{B}_2/\partial b = [0 \ (c-1)]^T$, and $\partial\mathcal{B}_2/\partial c = [0 \ b]^T$.

In summary, analytical expressions for the gradients of the outputs of the even Kautz filters with respect to the Kautz parameters $b$ and $c$ have been derived in this section. These gradients, given by $\nabla_{\mathbf{p}}\check{\psi}_{2m}(k) = [\partial\check{\psi}_{2m}(k)/\partial b \ \partial\check{\psi}_{2m}(k)/\partial c]^T$ (for $m = 1, 2, \ldots$), are described by (30) and (41) for $m = 1$ and $m \geq 2$, respectively.

### B. The Odd Kautz Functions

The goal of this section is to present results analogous to those obtained in Section IV-A, but with respect to the odd Kautz functions. Inasmuch as the formulation related to these functions can be derived following precisely the same steps detailed above for the even functions, the intermediate mathematical developments are omitted here for the sake of compactness. The final analytical expressions for the derivatives that compose the gradients of the outputs of the odd Kautz filters with respect to the Kautz parameters $b$ and $c$ for $m = 1$ are given by

$$\begin{aligned}
\frac{\partial \check{\psi}_1(k)}{\partial b} &= \mathcal{C}\frac{\partial(\mathcal{A}^k)}{\partial b}\mathbf{x}_1(0) \\
&\quad + \mathcal{C}\sum_{i=0}^{k-1}\left[\frac{\partial(\mathcal{A}^i)}{\partial b}\left(\mathcal{B}_4 u(k+1-i) + \mathcal{B}_5 u(k-i)\right) + \mathcal{A}^i\frac{\partial \mathcal{B}_5}{\partial b}u(k-i)\right] \\
\frac{\partial \check{\psi}_1(k)}{\partial c} &= \mathcal{C}\frac{\partial(\mathcal{A}^k)}{\partial c}\mathbf{x}_1(0) \\
&\quad + \mathcal{C}\sum_{i=0}^{k-1}\left[\frac{\partial(\mathcal{A}^i)}{\partial c}\left(\mathcal{B}_4 u(k+1-i) + \mathcal{B}_5 u(k-i)\right) + \mathcal{A}^i\frac{\partial \mathcal{B}_4}{\partial c}u(k+1-i) + \mathcal{A}^i\frac{\partial \mathcal{B}_5}{\partial c}u(k-i)\right]
\end{aligned} \quad (42)$$

with matrices $\mathcal{B}_4$ and $\mathcal{B}_5$ defined as

$$\mathcal{B}_4 = \begin{bmatrix} 0 \\ \sqrt{1-c^2} \end{bmatrix} \qquad \mathcal{B}_5 = \begin{bmatrix} 0 \\ -b\sqrt{1-c^2} \end{bmatrix}$$

and derivatives straightforwardly computed as $\partial\mathcal{B}_4/\partial c = [0 \ -c/\sqrt{1-c^2}]^T$, $\partial\mathcal{B}_5/\partial b = [0 \ -\sqrt{1-c^2}]^T$, and $\partial\mathcal{B}_5/\partial c = [0 \ bc/\sqrt{1-c^2}]^T$.

For $m \geq 2$, the analytical expressions for the derivatives are obtained by replacing $2m$ with $2m-1$ in (41), as presented in [43].

### C. Algorithm

The method proposed here can thus be summarized by the following steps. Starting from an initial parameter vector $\boldsymbol{\theta}^0$, do:

1) Compute the gradient $\nabla_{\boldsymbol{\theta}}^T J = [\nabla_{\mathbf{p}}^T J \ \nabla_{\boldsymbol{\alpha}}^T J]$ using (18) and (19). Note: use (30) and (41), as well as their counterparts for the odd Kautz functions, to compute the term $\nabla_{\mathbf{p}}\check{\psi}_m(k) = [\partial\check{\psi}_m(k)/\partial b \ \partial\check{\psi}_m(k)/\partial c]^T$ in (18);
2) Use $\nabla_{\boldsymbol{\theta}} J$ to update the parameter vector $\boldsymbol{\theta}$ according to the updating policy of a given optimization algorithm (e.g. Levenberg-Marquardt). Note: include the feasibility intervals of the Kautz parameters $b$ and $c$ as constraints into the optimization model so as to prevent the optimization algorithm from producing instable Kautz poles and/or poles that are not complex conjugate;
3) Go back to step 1 until a stopping criterion has been achieved.

## V. Generalization to the GOBF

Laguerre and Kautz bases are preferable for modeling systems with first- and second-order dominant dynamics, respectively. Systems having more complex dominant dynamics are better represented using models based on generalized orthonormal bases, because the mathematical description of such bases involves multiple poles. For this reason, this section elaborates on how to extend the results discussed so far in this paper to the GOBF.

Initially, consider the basis functions in (7) parameterized by real-valued poles only, i.e. $\beta_m = \bar{\beta}_m \triangleq p_m$ for $m = 1, \ldots, M$. In this case, one has

$$\Psi_m(z) = \frac{z\sqrt{1-p_m^2}}{z-p_m} \prod_{j=1}^{m-1} \left(\frac{1-p_j z}{z-p_j}\right) \tag{43}$$

and the whole collection of free-design parameters (to be optimized) of the GOBF can thus be arranged in a single real-valued vector encoding all the poles of the basis, i.e. $\mathbf{p} = [p_1 \ \cdots \ p_M]^T$, in such a way that

$$\begin{aligned}\nabla_{\mathbf{p}}\check{\psi}_m(k) &= \left[\partial\check{\psi}_m(k)/\partial p_1 \ \cdots \ \partial\check{\psi}_m(k)/\partial p_M\right]^T \\ &= \Big[\partial\check{\psi}_m(k)/\partial p_1 \ \cdots \ \partial\check{\psi}_m(k)/\partial p_m \\ &\qquad \underbrace{0 \ \cdots \ 0}_{(M-m) \text{ times}}\Big]^T. \end{aligned} \tag{44}$$

From (43) and (6) it follows that the output of the first GOBF filter is given by:

$$\check{\Psi}_1(z) = \frac{z\sqrt{1-p_1^2}}{z-p_1} U(z)$$

or equivalently, in the time-domain

$$\check{\psi}_1(k+1) - p_1\check{\psi}_1(k) = \sqrt{1-p_1^2}\,u(k+1). \tag{45}$$

The solution of (45) for $\check{\psi}_1(k)$ is readily obtained as

$$\check{\psi}_1(k) = p_1^k\check{\psi}_1(0) + \sqrt{1-p_1^2}\sum_{i=0}^{k-1} p_1^i u(k-i) \tag{46}$$

and the derivative of the output of the first GOBF filter with respect to its pole $p_1$ can thus be computed as

$$\begin{aligned}\frac{\partial\check{\psi}_1(k)}{\partial p_1} &= kp_1^{k-1}\check{\psi}_1(0) \\ &\quad + \frac{1}{\sqrt{1-p_1^2}}\sum_{i=0}^{k-1}\left[ip_1^{i-1} - (i+1)p_1^{i+1}\right]u(k-i). \end{aligned} \tag{47}$$

For $m \geq 2$, one can infer the following from (43) and (6):

$$\check{\Psi}_m(z) = \sqrt{\frac{1-p_m^2}{1-p_{m-1}^2}}\left(\frac{1-p_{m-1}z}{z-p_m}\right)\check{\Psi}_{m-1}(z).$$

Equivalently, in the time-domain one has the following difference equation:

$$\begin{aligned}&\check{\psi}_m(k+1) - p_m\check{\psi}_m(k) \\ &\quad = \sqrt{\frac{1-p_m^2}{1-p_{m-1}^2}}\left[\check{\psi}_{m-1}(k) - p_{m-1}\check{\psi}_{m-1}(k+1)\right] \end{aligned} \tag{48}$$

whose solution for $\check{\psi}_m(k)$ is given by

$$\begin{aligned}\check{\psi}_m(k) &= p_m^k\check{\psi}_m(0) + \sqrt{\frac{1-p_m^2}{1-p_{m-1}^2}} \\ &\quad \times \sum_{i=0}^{k-1} p_m^i\left[\check{\psi}_{m-1}(k-1-i) - p_{m-1}\check{\psi}_{m-1}(k-i)\right]. \end{aligned} \tag{49}$$

Now, it is necessary to compute the derivatives of $\check{\psi}_m(k)$ in (49) with respect to $p_l$ for $l = 1, \ldots, m$ in order to obtain the gradient $\nabla_{\mathbf{p}}\check{\psi}_m(k)$ in (44). Given that $\partial\check{\psi}_{m-1}(k)/\partial p_m = 0$, the derivatives of (49) with respect to $p_m$, $p_{m-1}$, and $p_l (l = 1, \ldots, m-2)$ can be computed using

$$\begin{aligned}\frac{\partial\check{\psi}_m(k)}{\partial p_m} &= kp_m^{k-1}\check{\psi}_m(0) + \frac{1}{\sqrt{(1-p_m^2)(1-p_{m-1}^2)}} \\ &\quad \times \sum_{i=0}^{k-1}\left[ip_m^{i-1} - (i+1)p_m^{i+1}\right] \\ &\quad \cdot\left[\check{\psi}_{m-1}(k-1-i) - p_{m-1}\check{\psi}_{m-1}(k-i)\right] \end{aligned} \tag{50}$$

$$\begin{aligned}\frac{\partial\check{\psi}_m(k)}{\partial p_{m-1}} &= \sum_{i=0}^{k-1}\Bigg[\sqrt{\frac{1-p_m^2}{(1-p_{m-1}^2)^3}}p_{m-1}p_m^i \\ &\quad \times\left(\check{\psi}_{m-1}(k-1-i) - p_{m-1}\check{\psi}_{m-1}(k-i)\right) \\ &\quad + \sqrt{\frac{1-p_m^2}{1-p_{m-1}^2}}p_m^i \\ &\quad \times\Bigg(\frac{\partial}{\partial p_{m-1}}\check{\psi}_{m-1}(k-1-i) - \check{\psi}_{m-1}(k-1-i) \\ &\qquad - p_{m-1}\frac{\partial}{\partial p_{m-1}}\check{\psi}_{m-1}(k-1-i)\Bigg)\Bigg] \end{aligned} \tag{51}$$

$$\begin{aligned}\frac{\partial\check{\psi}_m(k)}{\partial p_l} &= \sqrt{\frac{1-p_m^2}{1-p_{m-1}^2}}\sum_{i=0}^{k-1} p_m^i\Bigg[\frac{\partial}{\partial p_l}\check{\psi}_{m-1}(k-1-i) \\ &\qquad - p_{m-1}\frac{\partial}{\partial p_l}\check{\psi}_{m-1}(k-i)\Bigg] \\ &\quad l = 1, \ldots, m-2. \end{aligned} \tag{52}$$

The steps of the proposed method when the GOBF basis is adopted in model (16) are essentially the same as those described in Section IV-C. The only difference concerns the gradients $\nabla_{\mathbf{p}}\check{\psi}_m(k)$, which are now given by (44) and computed using (47) and (50)–(52) for $m = 1$ and $m \geq 2$, respectively. Recall, however, that these equations were derived by assuming that the generalized basis in (43) is parameterized with real-valued poles only. One of the reasons is that, if one or more poles are complex, then the GOBF in (7) has complex-valued inverse $Z$-transforms. Fortunately, this drawback can be circumvented by using an alternative generalized orthonormal basis of functions with real-valued impulse responses, as described in [21]. The gradients $\nabla_{\mathbf{p}}\check{\psi}_m(k)$ can be derived for this alternative basis in an analogous fashion. The basic idea is outlined in Appendix A.

Finally, it is worth remarking that the formulation presented in this section has been derived by considering that every GOBF pole is independent of the others. This means that the number of poles to be optimized coincides with the number of basis functions $(M)$. In practice, however, one may require only a limited number of independent poles that can be repeated multiple times, in such a way that the number of functions $M$ becomes a multiple of the number of poles to be optimized. The formulation regarding this particular case can be derived in a way analogous to that described above in this section.

## VI. Generalization to Higher-Order Volterra Models

As discussed in Section III-B, the gradients of the cost-function $J$ in (17), further derived in Sections IV and V, are valid only for first-order OBF Volterra models (linear OBF models). In this section we extend those results to any-order OBF Volterra models. This extension involves only the static portion of the models, that is, the Volterra polynomial in (4) for $N > 1$, since the dynamic portion does not depend on the Volterra polynomial order. So, let us consider the $\eta$th-order term of model (4), written individually as

$$\hat{y}_\eta(k) = \sum_{i_1=1}^{M} \cdots \sum_{i_\eta=1}^{M} \alpha_{i_1,\ldots,i_\eta} \prod_{l=1}^{\eta} \check{\psi}_{i_l}(k). \tag{53}$$

The gradient of (53) with respect to the parameter vector $\mathbf{p}$ of the orthonormal basis $\{\psi_m\}$ is given by

$$\begin{aligned}\nabla_{\mathbf{p}}\hat{y}_\eta(k) &= \sum_{i_1=1}^{M} \cdots \sum_{i_\eta=1}^{M} \alpha_{i_1,\ldots,i_\eta} \left[ \sum_{l=1}^{\eta} \nabla_{\mathbf{p}}\check{\psi}_{i_l}(k) \prod_{\substack{j=1 \\ j\neq l}}^{\eta} \check{\psi}_{i_j}(k) \right] \\ &= \sum_{l=1}^{\eta} \left[ \sum_{i_1=1}^{M} \cdots \sum_{i_\eta=1}^{M} \alpha_{i_1,\ldots,i_\eta} \nabla_{\mathbf{p}}\check{\psi}_{i_l}(k) \prod_{\substack{j=1 \\ j\neq l}}^{\eta} \check{\psi}_{i_j}(k) \right]\end{aligned} \tag{54}$$

where $\mathbf{p} \triangleq [b\ c]^T$ when $\{\psi_m\}$ is the Kautz basis (Section IV) and $\mathbf{p} \triangleq [p_1\ \cdots\ p_M]^T$ when GOBF are used (Section V). Note that the overall output of model (4) can be rewritten using (53) as $\hat{y}(k) = \sum_{\eta=1}^{N} \hat{y}_\eta(k)$, which implies $\nabla_{\mathbf{p}}\hat{y}(k) = \sum_{\eta=1}^{N} \nabla_{\mathbf{p}}\hat{y}_\eta(k)$. Hence, the gradient of the cost-function $J$ for the optimization problem (17) can now be computed as

$$\begin{aligned}\nabla_{\mathbf{p}} J &= \sum_{k=1}^{N_d} [\hat{y}(k) - y(k)]\, \nabla_{\mathbf{p}}\hat{y}(k) \\ &= \sum_{k=1}^{N_d} \sum_{\eta=1}^{N} [\hat{y}(k) - y(k)]\, \nabla_{\mathbf{p}}\hat{y}_\eta(k)\end{aligned} \tag{55}$$

where $\nabla_{\mathbf{p}}\hat{y}_\eta(k)$ is given by (54). From an analogous reasoning one has $\nabla_{\boldsymbol{\alpha}}\hat{y}(k) = \sum_{\eta=1}^{N} \nabla_{\boldsymbol{\alpha}}\hat{y}_\eta(k)$, where $\nabla_{\boldsymbol{\alpha}}\hat{y}_\eta(k)$ can be readily derived from (53), and the gradient of $J$ with respect to $\boldsymbol{\alpha}$ can thus be computed as

$$\begin{aligned}\nabla_{\boldsymbol{\alpha}} J &= \sum_{k=1}^{N_d} [\hat{y}(k) - y(k)]\, \nabla_{\boldsymbol{\alpha}}\hat{y}(k) \\ &= \sum_{k=1}^{N_d} \sum_{\eta=1}^{N} [\hat{y}(k) - y(k)]\, \nabla_{\boldsymbol{\alpha}}\hat{y}_\eta(k).\end{aligned} \tag{56}$$

The main steps of the algorithm described in Section IV-C keep unchanged. The only difference is that the gradients of the cost-funcion $J$ in (17) are now computed using (55) and (56) in place of (18) and (19), respectively. An asymptotic analysis of the computational complexity of the method is presented in Appendix B.

## VII. Illustrative Example

The proposed method is illustrated in this section by means of a simulated example and the results are compared with those from equivalent experiments reported in the literature. Specifically, the proposed method is used to select the poles for the dynamic system modeled in a contemporary paper [38], which consists of a second-order Volterra polynomial series with first- and second-order kernels given by

$$h_1(k_1) = \mathcal{Z}^{-1}\left[\frac{z+0.5}{(z-0.3)(z-0.2)}\right] \tag{57}$$

$$h_2(k_1,k_2) = 0.25 g(k_1) g(k_2) \tag{58}$$

where

$$g(k) = \mathcal{Z}^{-1}\left[\frac{z+1}{(z-0.8)(z+0.8)}\right]$$

in which $\mathcal{Z}^{-1}$ denotes the unilateral inverse $Z$-transform.

Much of the discussions presented in [38] is focused on comparing the modeling error when varying the number and the class of orthonormal functions used to expand the above kernels. Following the lines of [38], the model used here to approximate the system described above is also a second-order OBF Volterra representation with two independent generalized orthonormal bases $\{\psi_{1,m}\}$ and $\{\psi_{2,m}\}$ for expanding the first- and second-order kernels, respectively (see Remark 1 in Section II). The second-order kernel is considered to be symmetric, which means that it takes the same value for any permutation of its arguments [52].[2] In this case, one has $h_2(k_1,k_2) = h_2(k_2,k_1)$ and the coefficients of the orthonormal expansion thus satisfy $\alpha_{i_1,i_2} = \alpha_{i_2,i_1}$. Hence, the resulting model can be written as follows:

$$\hat{y}(k) = \sum_{i_1=1}^{M} \alpha_{i_1} \check{\psi}_{1,i_1}(k) + \sum_{i_1=1}^{M} \sum_{i_2=1}^{i_1} \alpha_{i_1,i_2} \check{\psi}_{2,i_1}(k) \check{\psi}_{2,i_2}(k) \tag{59}$$

with $\check{\psi}_{1,i}(k)$ and $\check{\psi}_{2,i}(k)$ denoting the result of the filtering of the input signal $u(k)$ by the generalized orthonormal functions $\psi_{1,i}(k)$ and $\psi_{2,i}(k)$, respectively. The bases $\{\psi_{1,m}\}$ and $\{\psi_{2,m}\}$ are designed to have $M$ real-valued poles each. Each basis is parameterized by an individual pole vector, i.e., the functions describing the first-order term of model (59) are parameterized by the pole vector $\mathbf{p}_1 \triangleq [p_{1,1}\ p_{1,2}\ \cdots\ p_{1,M}]^T$, whereas the functions describing the second-order term are parameterized by $\mathbf{p}_2 \triangleq [p_{2,1}\ p_{2,2}\ \cdots\ p_{2,M}]^T$, where $p_{\eta,m}$ is

[2]Note that, under the perspective of the output of the Volterra model, it is possible to replace any non-symmetric kernel with a symmetric equivalent by means of an ordinary symmetrization procedure [52].

TABLE I
ESTIMATED GOBF POLES OF MODEL (59) FOR $M = 4$

| Kernel order ($\eta$) | Final poles |
|---|---|
| 1 | $p_{1,1} = 0.2011$<br>$p_{1,2} = 0.2985$<br>$p_{1,3} = 0.0057$<br>$p_{1,4} = 0.0016$ |
| 2 | $p_{2,1} = 0.8027$<br>$p_{2,2} = -0.7982$<br>$p_{2,3} = 0.0047$<br>$p_{2,4} = 0.0025$ |

defined as the real-valued pole of the $m$th function of the $\eta$th basis. The calculation of the gradient of (59) with respect to $\mathbf{p} \triangleq [\mathbf{p}_1^T \ \mathbf{p}_2^T]^T$ is thus given by

$$\nabla_{\mathbf{p}}\hat{y}(k) = \sum_{i_1=1}^{M} \alpha_{i_1} \nabla_{\mathbf{p}} \breve{\psi}_{1,i_1}(k) + \sum_{i_1=1}^{M} \sum_{i_2=1}^{i_1} \alpha_{i_1,i_2} \left[ \breve{\psi}_{2,i_1}(k) \nabla_{\mathbf{p}} \breve{\psi}_{2,i_2}(k) + \breve{\psi}_{2,i_2}(k) \nabla_{\mathbf{p}} \breve{\psi}_{2,i_1}(k) \right]. \quad (60)$$

Note that the OBF in the first-order term of model (59) depend solely on their own pole vector $(\mathbf{p}_1)$, not on the pole vector of the second-order term $(\mathbf{p}_2)$, and vice-versa. Hence, the corresponding independent components of the gradients in (60) will be null.

The system under investigation is assumed to be excited by a random input signal with mean zero and variance one. Initially, no noise is added to the system output. An experiment with $M = 4$ is performed using the initialization pole vectors $\mathbf{p}_1^0 = \mathbf{p}_2^0 = [0\ 0\ 0\ 0]^T$. The method thus generates a series of values for the poles until the algorithm converges, after 18 iterations. The final poles are summarized in Table I. The corresponding expansion coefficients are given by

$$\boldsymbol{\alpha}_{1,\text{opt}} = [1.1954 \quad 0.8901 \quad 0.0006 \quad 0.0005]^T \quad (61)$$

$$\boldsymbol{\alpha}_{2,\text{opt}} = \begin{bmatrix} 0.8362 & * & * & * \\ 0.1833 & 0.0117 & * & * \\ -0.0083 & -0.0021 & -0.0001 & * \\ -0.0066 & -0.0005 & -0.0005 & 0.0003 \end{bmatrix}. \quad (62)$$

The original kernels in (57) and (58) are illustrated in Fig. 4(a) and (b) together with the corresponding kernels of the model, recovered from the expansion coefficients in (61) and (62), respectively. The output of the model is also compared to that of the original system with respect to another set of data (validation data), as illustrated in Fig. 5. The result can be quantified by the Normalized Quadratic Error (NQE) [38]

$$\text{NQE} \triangleq 10 \log \frac{\sum_{k=1}^{N_d} [y(k) - \hat{y}(k)]^2}{\sum_{k=1}^{N_d} [y(k)]^2} \quad (63)$$

which gives $\text{NQE} = -47.6$ dB for the noise-free simulation illustrated in Fig. 5.

For the sake of a fair comparison with [38], an additional experiment was performed under the same conditions adopted in that reference. Specifically, the model optimization procedure proposed here was repeated with Gaussian noise corresponding

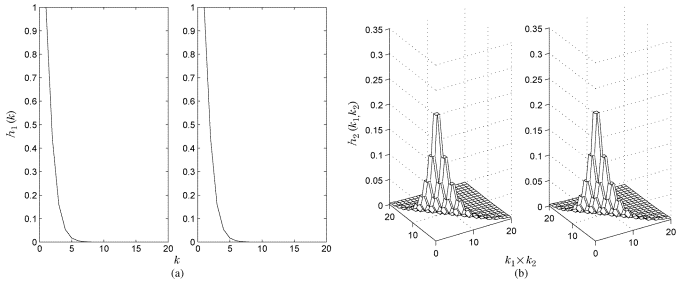



Fig. 4. Volterra kernels for the illustrative example: (a) original kernel in (57) (on the left) and its approximation with optimized GOBF (on the right); and (b) original kernel in (58) (on the left) and its approximation with optimized GOBF (on the right). (a) 1st-order kernel; (b) 2nd-order kernel.

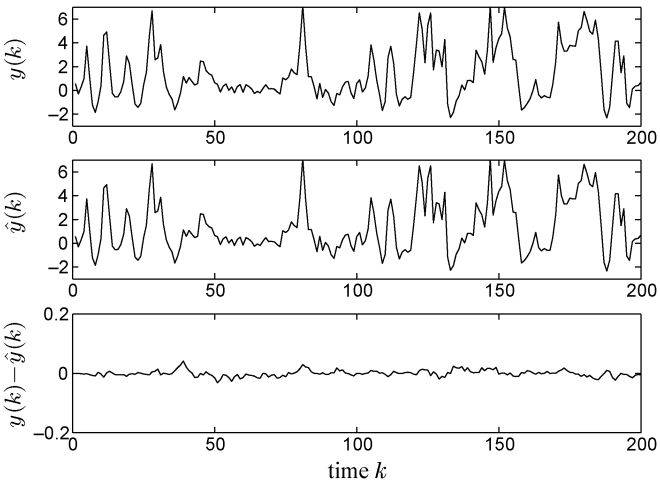


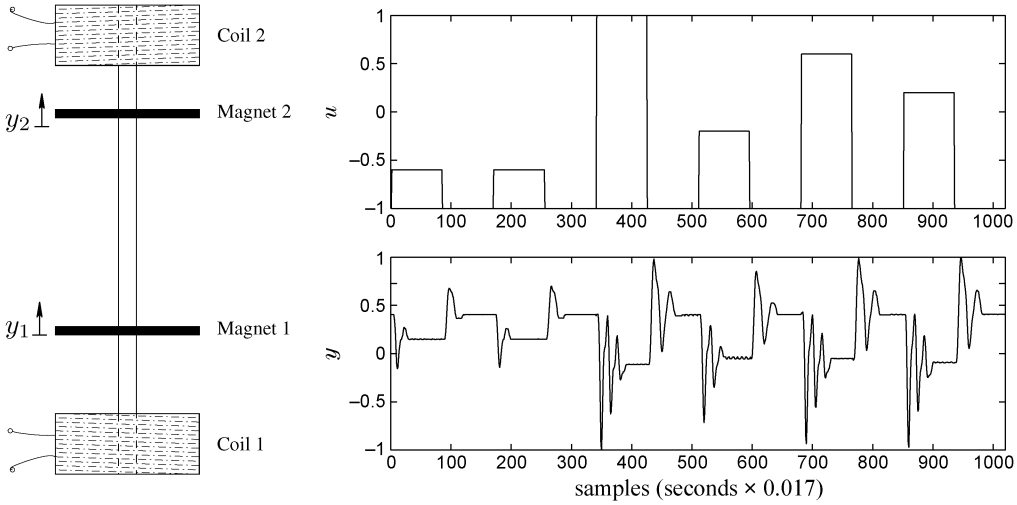

Fig. 5. Actual output (top), model output (middle) and error (bottom) for a set of noise-free validation data.

TABLE II
ERRORS OBTAINED USING DIFFERENT METHODS FOR APPROXIMATING THE SYSTEM WITH KERNELS (57) AND (58) USING $M = 4$ ORTHONORMAL FUNCTIONS (GAUSSIAN NOISE ADDED TO THE OUTPUT)

| Method | NQE (in dB) with SNR = 30dB |
|---|---|
| [38] (with Laguerre) | $-21.5$ |
| [38] (with GOBF) | $-27.0$ |
| Proposed here (with GOBF) | $-36.4$ |

to a signal-to-noise ratio (SNR) of 30 dB added to the output. A comparison of the errors obtained for different optimization methods and orthonormal bases is presented in Table II. The error obtained using the proposed methodology is computed by using an independent set of validation data. It is clear from Table II that, despite the noise added to the output, the modeling error resulting from the proposed method underwent a stronger attenuation than that reported in [38]. Table II also shows that, as expected, models based on GOBF provide better results than do those involving simpler bases (Laguerre in this case).

Another experiment, reported in a complementary material by the authors [43], compares the obtained results with expected results of a theoretical nature that can be derived analytically for a particular class of linear systems. In that example, the optimal Kautz poles have been computed by varying the speed (fast, medium and slow dynamics) of a system ruled by an impulse response purely characterized by a pair of complex conjugate




(a) Schematic sketch (b) Current through Coil 2 (above) and position of magnet 1 (below)

Fig. 6. Laboratory-scale magnetic levitation system: (a) schematic drawn; and (b) real I/O data for model estimation.

poles. According to the results obtained, the poles of the system have been almost perfectly recovered after a few iterations of the proposed methodology.

## VIII. Modeling of a Real Magnetic Levitation System

The magnetic levitation system considered here, schematically shown in Fig. 6(a), consists of upper and lower drive coils that produce a magnetic field in response to a dc current. Two magnets travel along a precision ground glass guide rod. By energizing the lower coil, one of the magnets is levitated by a repulsive magnetic force. As current in the coil increases, the field strength also increases and the height of the levitated magnet is increased too. Two laser-based sensors measure the positions of the magnets, which are designed to provide large levitated displacements [45].

Let $m_1$ be the mass of the lower magnet, $v_1$ the viscous friction coefficient between this magnet and air, and $g$ the acceleration of gravity. The movement of the lower magnet (Magnet 1) is governed by $m_1\ddot{y}_1 + v_1\dot{y}_1 + F_{m12} = F_{u11} - F_{u21} - m_1 g$, where $y_1$ is the position of Magnet 1, $F_{u11}$ is the magnetic force from Coil 1 interacting with Magnet 1, $F_{u21}$ is the magnetic force from Coil 2 interacting with Magnet 1, and $F_{m12}$ is the mutual magnetic force between the two magnets. These forces are described by the following nonlinear equations:

$$F_{m12} = \frac{c}{(y_c + y_2 - y_1 + d)^4} \qquad F_{u11} = \frac{i_1}{a(k_s y_1 + b)^4}$$
$$F_{u21} = \frac{i_2}{a(y_c + k_s y_1 + b)^4}$$

where $y_2$ is the position of Magnet 2, $y_c$ is the distance between Coils 1 and 2, $i_1$ and $i_2$ are the currents through Coils 1 and 2, respectively, and $a$, $b$, $c$, and $d$ are real-valued constants. In the sequel, some experimental results for modeling a laboratory-scale plant of the magnetic levitation system described above are presented.

Experimental data have been acquired by keeping constant the dc current applied to Coil 1 of the plant while varying current through Coil 2. The input signal $u$ (current through Coil 2) has been designed as a sequence of steps with different amplitudes so as to excite different modes of the system. The measured output signal $y$ has been taken as the position of Magnet 1 $(y_1)$. Fig. 6(b) shows the input and output data available for estimation of the model. Before estimation, these data were sampled with a period of 0.017 seconds and normalized within $[-1,1]$ in order to avoid numerical problems. Another similar yet independent set of data has also been acquired and reserved for further model validation.[3]

The model adopted here relates the input $u(k)$ and output $y(k)$ in Fig. 6(b) by means of a second-order $(N = 2)$ Volterra representation with a symmetric second-order kernel, as usual in the literature [7], [10]. By describing its first- and second-order terms as expansions on two independent Kautz bases, the model equation becomes the same as that in (59). The Kautz functions describing the first-order term of this model are parameterized by a pole vector $\mathbf{p}_1 \triangleq [b_1\ c_1]^T$, whereas those describing the second-order term are parameterized by $\mathbf{p}_2 \triangleq [b_2\ c_2]^T$. The gradient of the model output with respect to $\mathbf{p} \triangleq [\mathbf{p}_1^T\ \mathbf{p}_2^T]^T$ is thus given by (60).

In this experiment, the initial Kautz poles are chosen as $(\beta_1^0, \bar{\beta}_1^0) = (\beta_2^0, \bar{\beta}_2^0) = 0.5 \pm \mathrm{i}0.5$ for the first- and second-order terms of model (59). In practice, a better initial estimate of the poles could be obtained by observing the time response of the system (Fig. 6(b)). Instead, it is considered here that a naive user has blindly chosen the initial poles as $(\beta_1^0, \bar{\beta}_1^0) = (\beta_2^0, \bar{\beta}_2^0) = 0.5 \pm \mathrm{i}0.5$ for they are placed in the middle of the feasible search space for the real and imaginary components of the poles. These values result in the initial real-valued parameter vectors $\mathbf{p}_1^0 = \mathbf{p}_2^0 = [0.6667\ -0.5]^T$ from (10) and (11). The number of Kautz functions is chosen as $M = 8$. One thus applies the algorithm using (30), (41), and their counterparts for the odd Kautz functions to compute $\nabla_{\mathbf{p}}\breve{\psi}_m(k)$ and, consequently, to get the search directions for the poles. The evolution of the poles of the orthonormal

[3]The estimation/validation data sets are available at http://www.icmc.usp.br/~campello/Sub_Pages/IEEETAC_inpress.htm.

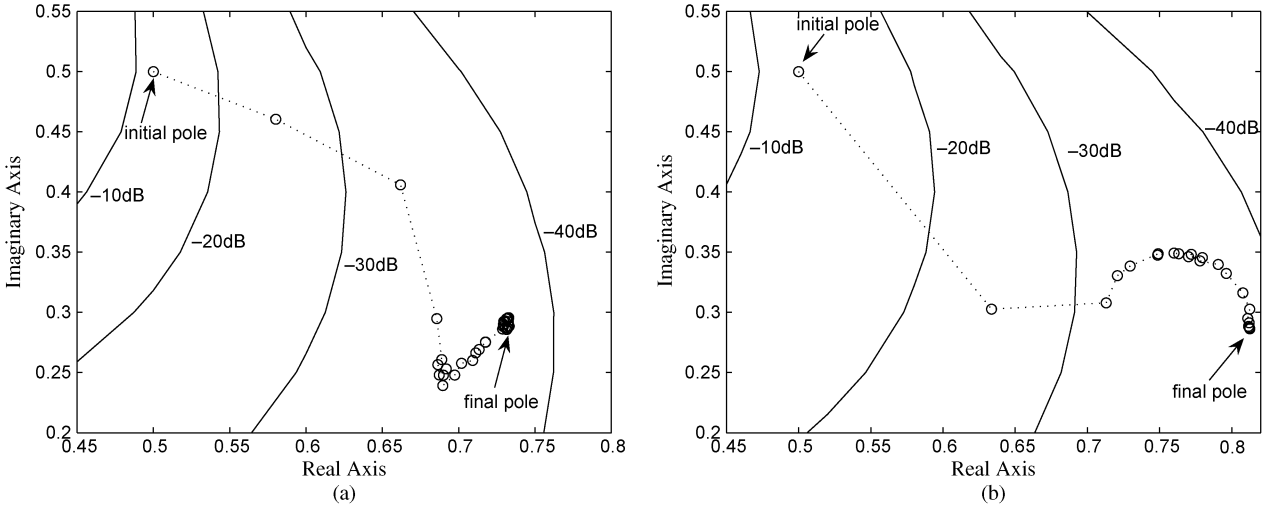



Fig. 7. Evolution of the Kautz poles for the magnetic levitation system throughout iterations (including some contour lines of NQE). The final poles are 0.7328 ± i0.2956 for the first-order term and 0.8125 ± i0.2863 for the second-order term of the Volterra model. (a) First-order term; (b) second-order term.

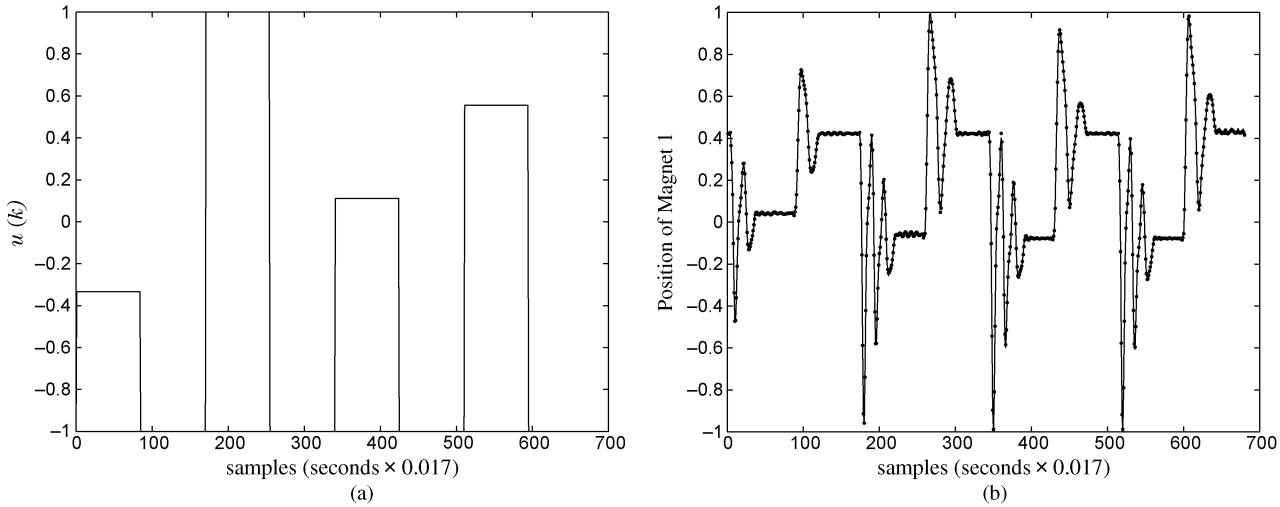



Fig. 8. Modeling of the magnetic levitation system: (a) Input signal used for model validation; and (b) actual system output ($y(k)$ in solid line) and predicted output of the model with optimized Kautz poles ($\hat{y}(k)$ in dotted line).

TABLE III
ESTIMATED KAUTZ POLES FOR THE MAGNETIC LEVITATION SYSTEM

| Kernel order ($\eta$) | $b_{\rm opt}$ | $c_{\rm opt}$ | Final poles $(\beta_{\rm opt}, \beta_{\rm opt})$ |
|---|---|---|---|
| 1 | 0.9023 | −0.6245 | $0.7328 \pm$ i0.2956 |
| 2 | 0.9328 | −0.7421 | $0.8125 \pm$ i0.2863 |

functions for the first- and second-order terms of the model of the magnetic levitation system is shown in Fig. 7(a) and (b), respectively. Some contour lines of NQE are also shown in these figures. Table III presents the values of the Kautz parameters and poles after the convergence of the algorithm, which took 39 iterations. The stopping criterion adopted in this experiment was the difference between the value of $J$ in (17) in successive iterations be less than $10^{-5}$.

The model of the magnetic levitation system can now be compared to the actual system output with regard to the data samples reserved for model validation. Fig. 8(a) shows the set of input data, designed as a sequence of steps with different amplitudes. Fig. 8(b) displays the corresponding model output, $\hat{y}(k)$, jointly with the actual output measured from the system, $y(k)$. It can be seen that a nearly exact approximation of this highly nonlinear system has been obtained.

## IX. CONCLUSION

A novel strategy for numerical optimization of orthonormal bases of functions used for approximation of dynamic systems has been proposed. This strategy, which is valid for linear models and also for Volterra models of any order, is based on the computation of analytical expressions for the derivatives of the output of the orthonormal filters with respect to the basis poles. Such expressions provide exact search directions to be used by a numerical method for optimizing the OBF poles in linear and nonlinear models. The solution for the underlying problem involves the minimization of a quadratic cost-function that takes into consideration the error of estimation of the system output. The expressions related to the Kautz and GOBF bases have been emphasized, but those related to the Laguerre basis can be derived straightforwardly as a particular case. The

main advantage of the proposed method is that it computes the gradients analytically using only I/O data measured from the system to be modeled. No previous information about the system impulse response or higher-order Volterra kernels is required. Contrarily, the kernels can be straightforwardly recovered from their optimized OBF expansions if desired. In particular, first- and second-order kernels recovered from the series expansion can be used for graphical visualization of the corresponding dynamics of the model. An illustrative example involving a nonlinear simulated system has shown that the proposed method performs very well and arises as a promising tool for practical OBF-model-based system identification and control. In this context, the proposed method has been applied to the optimization of a second-order Kautz-Volterra model of a real magnetic levitation system with strong nonlinear oscillatory behavior. Simulations suggest that the resulting model represents the dynamics of the system almost perfectly.

Extensions of the proposed ideas towards adaptive, stochastic, robust, and frequency-domain model formulations are still open problems that deserve further investigations.

## APPENDIX A
## GENERALIZATION TO GOBF WITH COMPLEX POLES

As discussed in Section II-A, the orthonormal basis functions defined in (7) have complex-valued inverse $Z$-transforms when they are designed with complex-valued poles. This drawback can be circumvented by constructing a modified basis of functions with complementary pairs of real-valued impulse responses, as shown in [21]. This modified, alternative representation will be used here to compute the gradients of the GOBF with complex conjugate poles.

Let us first consider an OBF-based model with the first $M-1$ basis functions $\{\Psi_1(z), \Psi_2(z), \ldots, \Psi_{M-1}(z)\}$ parameterized by their respective $M-1$ real-valued poles, given by the set $\{p_1, p_2, \ldots, p_{M-1}\}$, precisely as in (43). According to [21], if it is desired to include a complex-valued pole $\beta_M$ into this set of poles, then two modified functions $\Psi'_M(z)$ and $\Psi''_M(z)$ with real-valued impulse responses must be constructed as a linear combination of $\Psi_M(z)$ and $\Psi_{M+1}(z)$ in (7). In this case, the new set of functions will be $\{\Psi_1(z), \ldots, \Psi_{M-1}(z), \Psi'_M(z), \Psi''_M(z)\}$, with the corresponding set of poles $\{p_1, \ldots, p_{M-1}, \beta_M, \bar{\beta}_M\}$. The pair of modified functions is given by [21]

$$\Psi'_M(z) = \frac{z\sqrt{1-|\beta_M|^2}(\lambda' z + \gamma')}{z^2 - (\beta_M + \bar{\beta}_M)z + |\beta_M|^2} \prod_{j=1}^{M-1}\left(\frac{1-p_j z}{z-p_j}\right)$$

$$\Psi''_M(z) = \frac{z\sqrt{1-|\beta_M|^2}(\lambda'' z + \gamma'')}{z^2 - (\beta_M + \bar{\beta}_M)z + |\beta_M|^2} \prod_{j=1}^{M-1}\left(\frac{1-p_j z}{z-p_j}\right) \quad (64)$$

where $\beta_M \in \mathbb{C}$ is the complex-valued pole included into the GOBF and $\lambda', \gamma', \lambda'', \gamma''$ are real-valued parameters that relate to $\beta_M$ as follows:

$$[\lambda' \quad \gamma'] \begin{bmatrix} 1+|\beta_M|^2 & \beta_M + \bar{\beta}_M \\ \beta_M + \bar{\beta}_M & 1+|\beta_M|^2 \end{bmatrix} \begin{bmatrix} \lambda' \\ \gamma' \end{bmatrix} = \left|1-\beta_M^2\right|^2$$

$$\begin{bmatrix} \lambda'' \\ \gamma'' \end{bmatrix} = \frac{1}{\sqrt{1-\xi^2}} \begin{bmatrix} \xi & 1 \\ -1 & -\xi \end{bmatrix} \begin{bmatrix} \lambda' \\ \gamma' \end{bmatrix}$$

$$\xi = \frac{\beta_M + \bar{\beta}_M}{1+|\beta_M|^2}.$$

According to (6), (43) and (64), the following relationship can be derived:

$$\check{\Psi}'_M(z) = \sqrt{\frac{1-|\beta_M|^2}{1-p_{M-1}^2}} \cdot \frac{(\lambda' z + \gamma')(1-p_{M-1}z)}{z^2 - (\beta_M + \bar{\beta}_M)z + |\beta_M|^2} \check{\Psi}_{M-1}(z)$$

or equivalently, in the time-domain

$$\begin{aligned} &\check{\psi}'_M(k+2) - (\beta_M + \bar{\beta}_M)\check{\psi}'_M(k+1) + |\beta_M|^2 \check{\psi}'_M(k) \\ &\quad = \sqrt{\frac{1-|\beta_M|^2}{1-p_{M-1}^2}} \cdot \big[-\lambda' p_{M-1}\check{\psi}_{M-1}(k+2) \\ &\qquad + (\lambda' - \gamma' p_{M-1})\check{\psi}_{M-1}(k+1) \\ &\qquad + \gamma'\check{\psi}_{M-1}(k)\big]. \end{aligned} \quad (65)$$

In order to solve the difference equation in (65), it is helpful adopting a state-space representation. In this case, (65) can be rewritten as

$$\begin{cases} \mathbf{x}'_M(k+1) = \mathcal{A}_1 \mathbf{x}'_M(k) + \mathcal{B}_6\check{\psi}_{M-1}(k+2) \\ \quad + \mathcal{B}_7\check{\psi}_{M-1}(k+1) + \mathcal{B}_8\check{\psi}_{M-1}(k) \\ \check{\psi}'_M(k) = \mathcal{C}\mathbf{x}'_M(k) \end{cases}$$

whose solution for $\check{\psi}'_M(k)$ is given by

$$\begin{aligned} \check{\psi}'_M(k) = \mathcal{C}\mathcal{A}_1^k \mathbf{x}'_M(0) + \mathcal{C}\sum_{i=0}^{k-1} \mathcal{A}_1^i \big[&\mathcal{B}_6\check{\psi}_{M-1}(k+1-i) \\ &+ \mathcal{B}_7\check{\psi}_{M-1}(k-i) + \mathcal{B}_8\check{\psi}_{M-1}(k-1-i)\big] \end{aligned} \quad (66)$$

where $\mathbf{x}'_M(0) = [\check{\psi}'_M(0)\ \check{\psi}'_M(1)]^T$, matrix $\mathcal{C}$ is as in (29), and $\mathcal{A}_1, \mathcal{B}_6, \mathcal{B}_7, \mathcal{B}_8$ are as follows:

$$\mathcal{A}_1 = \begin{bmatrix} 0 & 1 \\ -|\beta_M|^2 & \beta_M + \bar{\beta}_M \end{bmatrix} \quad \mathcal{B}_6 = K_M \begin{bmatrix} 0 \\ -\lambda' p_{M-1} \end{bmatrix}$$

$$\mathcal{B}_7 = K_M \begin{bmatrix} 0 \\ \lambda' - \gamma' p_{M-1} \end{bmatrix} \quad \mathcal{B}_8 = K_M \begin{bmatrix} 0 \\ \gamma' \end{bmatrix}$$

$$K_M = \sqrt{\frac{1-|\beta_M|^2}{1-p_{M-1}^2}}.$$

In order to compute the gradient of (66) with respect to $\beta_M$, this pole can be merely considered as a real-valued vector, i.e. $\beta_M \triangleq [\mathbf{Re}(\beta_M)\ \mathbf{Im}(\beta_M)]^T$. In this case, the gradient is defined as $\nabla_{\beta_M}\check{\psi}'_M(k) \triangleq [\partial\check{\psi}'_M(k)/\partial\mathbf{Re}(\beta_M)\ \partial\check{\psi}'_M(k)/\partial\mathbf{Im}(\beta_M)]^T$. The first element of this gradient is given by

$$\begin{aligned} &\frac{\partial\check{\psi}'_M(k)}{\partial\mathbf{Re}(\beta_M)} \\ &\quad = \mathcal{C}\frac{\partial\mathcal{A}_1^k}{\partial\mathbf{Re}(\beta_M)}\mathbf{x}'_M(0) + \mathcal{C}\sum_{i=0}^{k-1}\frac{\partial\mathcal{A}_1^i}{\partial\mathbf{Re}(\beta_M)} \\ &\qquad \times \big[\mathcal{B}_6\check{\psi}_{M-1}(k+1-i) + \mathcal{B}_7\check{\psi}_{M-1}(k-i) \\ &\qquad\quad + \mathcal{B}_8\check{\psi}_{M-1}(k-1-i)\big] \\ &\qquad + \mathcal{A}_1^i\left[\frac{\partial\mathcal{B}_6}{\partial\mathbf{Re}(\beta_M)}\check{\psi}_{M-1}(k+1-i)\right. \\ &\qquad\quad \left. + \frac{\partial\mathcal{B}_7}{\partial\mathbf{Re}(\beta_M)}\check{\psi}_{M-1}(k-i)\right. \end{aligned}$$

$$+\frac{\partial \mathcal{B}_8}{\partial \mathbf{Re}(\beta_M)}\check{\psi}_{M-1}(k-1-i)\Big] \qquad (67)$$

whereas the second one can be obtained by replacing $\mathbf{Re}(\beta_M)$ with $\mathbf{Im}(\beta_M)$ in (67).

Finally, the derivatives of (66) with respect to $p_l (l = 1,\ldots,M-1)$ can be computed using the following equation:

$$\begin{aligned}\frac{\partial \check{\psi}'_M(k)}{\partial p_l} = \mathcal{C}\sum_{i=0}^{k-1}\mathcal{A}_1^i\Bigg[&\mathcal{B}_6\frac{\partial \check{\psi}_{M-1}(k+1-i)}{\partial p_l}\\ &+\frac{\partial \mathcal{B}_6}{\partial p_l}\check{\psi}_{M-1}(k+1-i)\\ &+\mathcal{B}_7\frac{\partial \check{\psi}_{M-1}(k-i)}{\partial p_l}\\ &+\frac{\partial \mathcal{B}_7}{\partial p_l}\check{\psi}_{M-1}(k-i)\\ &+\mathcal{B}_8\frac{\partial \check{\psi}_{M-1}(k-1-i)}{\partial p_l}\\ &+\frac{\partial \mathcal{B}_8}{\partial p_l}\check{\psi}_{M-1}(k-1-i)\Bigg]. \qquad (68)\end{aligned}$$

The derivations for the complementary basis function $\check{\Psi}''_M(z)$ in (64) are analogous (*mutatis mutandis*) and are omitted here.

If it is desired to include an additional pair of complex conjugate poles into the GOBF, then two more functions with real-valued impulse responses must be constructed as a linear combination of the complex-valued functions in (7), as suggested in [21]. The corresponding gradients are presented in a complementary material by the authors [43].

## APPENDIX B
## ANALYSIS OF THE COMPUTATIONAL COST OF THE PROPOSED METHOD

This appendix presents an asymptotic complexity analysis of the proposed method in terms of computing time. For the reader not familiar with asymptotic analysis, please refer to [44] for an introduction. Before proceeding with the discussions, it is important to remark that the search directions generated by the proposed method can be used jointly with any gradient-based optimization algorithm, that is to say, their use is not necessarily hooked on the particular optimization algorithm suggested and adopted in the experiments of this paper (Levenberg-Marquardt). For this reason, the following discussions are restricted to the computational complexity of the procedures for calculation of these directions only.

### A. Linear Kautz Model

In the case of linear OBF models with Kautz functions, the possibly critical variables of the proposed method in terms of computing time are the number of I/O samples, $N_d$, and the number of functions in the orthonormal basis, $M$. The complexity analysis considered here consists of evaluating the asymptotic computational cost of the steps of the proposed method with regard to these variables. The steps are those necessary to calculate (18), (19), (30) and (41)[4] (please, see the algorithm in Section IV-C).

[4]The computational complexity of (42) and its counterpart for $m \geq 2$ is the same as that of (30) and (41).

(I) According to the respective definitions of matrices $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\partial\mathcal{A}/\partial b$, $\partial\mathcal{A}/\partial c$, $\partial\mathcal{B}/\partial b$, $\partial\mathcal{B}/\partial c$, $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{B}_4$, $\mathcal{B}_5$, $\partial\mathcal{B}_1/\partial c$, $\partial\mathcal{B}_2/\partial b$, $\partial\mathcal{B}_2/\partial c$, $\partial\mathcal{B}_4/\partial c$, $\partial\mathcal{B}_5/\partial b$, and $\partial\mathcal{B}_5/\partial c$ in Sections IV-A and IV-B, it is clear that none of them depends on any of the above-mentioned potentially critical variables of the problem ($N_d$ and $M$). Hence, their calculation is made in constant time with regard to these variables, that is, in $O(1)$ time.

(II) The cost of computing $\mathcal{A}^k$, either from (34) or (35), is determined by the cost of computing term $\beta^k$. As such a term can be computed incrementally for $k = 1,2,\ldots$, the overall cost of computing $\mathcal{A}^k$ for all $k = 1,\ldots,N_d$ is of orders of magnitude $O(N_d)$.

(III) Each matrix multiplication in (31) and (32) is performed in constant time, but the summation makes the calculation of $\partial\mathcal{A}^k/\partial b$ and $\partial\mathcal{A}^k/\partial c$, for $k = 1,\ldots,N_d$, demand $O(1+2+\cdots+N_d)$ time, that is, $O(N_d\cdot(N_d+1)/2)$, which is $O(N_d^2)$.

(IV) It is trivial to show that, by representing the Kautz dynamics in state-space form, the calculation of the outputs of all the $M$ Kautz filters in a given time instant $k$, $\check{\psi}_i(k)(i = 1,\ldots,M)$, can be made in $O(M^2)$ time. Since it is necessary to calculate all the filter outputs for $k = 1,\ldots,N_d$, one gets $O(M^2N_d)$ time. Once the outputs of the Kautz filters have been computed, it is possible to compute the output of the model, $\hat{y}(k)$, for all $k$, using (16). Such a computation demands additional $O(MN_d)$ time. Hence, the calculation of the outputs of the filters and the output of the model for all sampling instants demand $O(M^2N_d+MN_d)$ time, or, equivalently, $O(M^2N_d)$.

(V) Once all the terms mentioned in the previous items are available, (30) and (41) can be computed. Each of them demands, for $m$ and $k$ given (fixed), a number proportional to $k$ algebraic operations, each of which to be computed in constant time, i.e., in $O(k)$ time altogether. However, for a given $m$, it is necessary to calculate such equations for $k = 1,\ldots,N_d$, which results in $O(1+2+\cdots+N_d)$ time, that is, $O(N_d^2)$. Finally, it is necessary to carry through these calculations for each one of the Kautz filters, that is, $m = 1,\ldots,M$, which implies a total time of orders of magnitude $O(MN_d^2)$.

(VI) Let $n_p$ be the dimension of the pole vector $\mathbf{p}$ in (18). The computational time needed to calculate all the $n_p$ scalar components of $\nabla_{\mathbf{p}}J$ in (18) is of orders of magnitude $O(n_pMN_d)$. For the Kautz basis it follows that $n_p = 2$ ($\mathbf{p} = [b\ c]^T$), which implies $O(MN_d)$.

(VII) The time required for calculation of each one of the $M$ scalar components of the gradient vector $\nabla_{\boldsymbol{\alpha}}J$ in (19) is of orders of magnitude $O(N_d)$, which implies $O(MN_d)$ time for computing the whole gradient.

From items (I) through (VII) listed above, it follows that the time orders of magnitude for computing the directions for optimization of linear Kautz models according to the method proposed in this paper is $O(M^2N_d+N_d^2M)$. In practice, however, one typically has $N_d \gg M$, which makes it possible to assert that the orders of magnitude is simply $O(N_d^2)$.

### *B. Extensions to GOBF and Volterra*

The only difference when GOBF are used instead of Kautz is that the dimension of the pole vector $\mathbf{p}$ in (18) is no longer two, but it is such that $n_p \propto M$. This change affects only item (VI) of the previous analysis for the Kautz case, that comes to be $O(M^2 N_d)$ when GOBF is used. Such a change, however, does not affect the overall complexity of the method, which is still $O(M^2 N_d + N_d^2 M)$, or simply $O(N_d^2)$ given the practical hypothesis that the number of I/O samples used for optimization is much greater than the number of orthonormal functions used in the model ($N_d \gg M$). It is not difficult to show that this result also holds for Volterra models, provided that the Volterra polynomial order, $N$, is considered to be a predefined constant rather than a critical variable. Such an assumption is quite realistic because, in practice, $N$ is typically equal to 2 [7], [10] and rarely greater than or equal to 3.

## REFERENCES


[1] S. Takenaka, "On the orthogonal functions and a new formula of interpolation," *Japanese J. Math.*, vol. II, pp. 129–145, 1925.

[2] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *Quart. Appl. Math.*, vol. 2, no. 2, pp. 164–168, 1944.

[3] W. H. Kautz, "Transient synthesis in time domains," *IRE Trans. Circuit Theory*, vol. 1, no. 3, pp. 29–39, 1954.

[4] D. W. Marquardt, "An algorithm for the least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, vol. 11, no. 2, pp. 431–441, 1963.

[5] P. W. Broome, "Discrete orthonormal sequences," *J. Assoc. Comput. Mach.*, vol. 12, no. 2, pp. 151–168, 1965.

[6] G. J. Clowes, "Choice of time-scaling factor for linear system approximation using orthonormal Laguerre functions," *IEEE Trans. Autom. Control*, vol. AC-10, no. 4, pp. 487–489, Oct. 1965.

[7] S. A. Billings, "Identification of nonlinear systems—A survey," *Proc. Inst. Elect. Eng. D*, vol. 127, no. 6, pp. 272–285, 1980.

[8] S. Boyd and L. O. Chua, "Fading memory and the problem of approximation nonlinear operators with Volterra series," *IEEE Trans. Circuits Syst.*, vol. CS-32, no. 11, pp. 1150–1161, Nov. 1985.

[9] M. A. Masnadi-Shirazi and N. Ahmed, "Optimum Laguerre networks for a class of discrete-time systems," *IEEE Trans. Signal Processing*, vol. 39, no. 9, pp. 2104–2108, Sep. 1991.

[10] G. A. Dumont and Y. Fu, "Non-linear adaptive control via Laguerre expansion of Volterra kernels," *Int. J. Adaptive Control Signal Processing*, vol. 7, no. 5, pp. 367–382, 1993.

[11] Y. Fu and G. A. Dumont, "An optimum time scale for discrete Laguerre network," *IEEE Trans. Autom. Control*, vol. 38, no. 6, pp. 934–938, Jun. 1993.

[12] T. A. M. O. Silva, "Optimality conditions for truncated Laguerre networks," *IEEE Trans. Signal Processing*, vol. 42, no. 9, pp. 2528–2530, Sep. 1994.

[13] B. Wahlberg, "System identification using Kautz models," *IEEE Trans. Autom. Control*, vol. 39, no. 6, pp. 1276–1282, Jun. 1994.

[14] P. M. J. Van den Hof, P. S. C. Heuberger, and J. Bokor, "System identification with generalized orthonormal basis functions," *Automatica*, vol. 31, no. 12, pp. 1821–1834, 1995.

[15] P. S. C. Heuberger, P. M. J. Van den Hof, and O. H. Bosgra, "A generalized orthonormal basis for linear dynamical systems," *IEEE Trans. Autom. Control*, vol. 40, no. 3, pp. 451–465, Mar. 1995.

[16] T. A. M. O. Silva, "Optimality conditions for truncated Kautz networks with two periodically repeating complex conjugate poles," *IEEE Trans. Autom. Control*, vol. 40, no. 2, pp. 342–346, Feb. 1995.

[17] N. Tanguy, R. Morvan, P. Vilbé, and L. C. Calvez, "Optimum choice of free parameter in orthonormal approximations," *IEEE Trans. Autom. Control*, vol. 40, no. 10, pp. 1811–1813, Oct. 1995.

[18] A. C. den Brinker, F. P. A. Benders, and T. A. M. O. Silva, "Optimality conditions for truncated Kautz series," *IEEE Trans. Circuits Syst. II*, vol. 43, no. 2, pp. 117–122, Feb. 1996.

[19] B. Wahlberg and P. M. Mäkilä, "Approximation of stable linear dynamical systems using Laguerre and Kautz functions," *Automatica*, vol. 32, no. 5, pp. 693–708, 1996.

[20] P. Bodin, L. F. Villemoes, and B. Wahlberg, "An algorithm for selection of best orthonormal rational basis," in *Proc. 36th Conf. Decision Control*, San Diego, CA, 1997, pp. 1277–1282.

[21] B. Ninness and F. Gustafsson, "A unifying construction of orthonormal bases for system identification," *IEEE Trans. Autom. Control*, vol. 42, no. 4, pp. 515–521, Apr. 1997.

[22] T. A. M. O. Silva, "Stationary conditions for the $L^2$ error surface of the generalized orthonormal basis functions lattice filter," *Signal Processing*, vol. 56, no. 3, pp. 233–253, 1997.

[23] J. Bokor and F. Schipp, "Approximate identification in Laguerre and Kautz bases," *Automatica*, vol. 34, no. 4, pp. 463–468, 1998.

[24] G. H. C. Oliveira, W. C. Amaral, G. Favier, and G. A. Dumont, "Constrained robust predictive controller for uncertain processes modeled by orthonormal series functions," *Automatica*, vol. 36, no. 4, pp. 563–571, 2000.

[25] N. Tanguy, R. Morvan, P. Vilbé, and L. C. Calvez, "Online optimization of the time scale in adaptive Laguerre-based filters," *IEEE Trans. Signal Processing*, vol. 48, no. 4, pp. 1184–1187, Apr. 2000.

[26] N. Tanguy, R. Morvan, P. Vilbé, and L. C. Calvez, "Pertinent parameters for Kautz approximation," *Eletron. Lett.*, vol. 36, no. 8, pp. 769–771, 2000.

[27] R. J. G. B. Campello, W. C. Amaral, and G. Favier, "Optimal Laguerre series expansion of discrete Volterra models," in *Proc. Eur. Control Conf.*, Porto, Portugal, 2001, pp. 372–377.

[28] R. Hacioğlu and G. A. Williamson, "Reduced complexity Volterra models for nonlinear system identification," *EURASIP J. Appl. Signal Processing*, vol. 2001, no. 4, pp. 257–265, 2001.

[29] L. S. H. Ngia, "Separable nonlinear least-squares methods for efficient off-line and on-line modeling of systems using Kautz and Laguerre filters," *IEEE Trans. Circuits Syst. II*, vol. 48, no. 6, pp. 562–579, Jun. 2001.

[30] B. E. Sarroukh, S. J. L. van Eijndhoven, and A. C. den Brinker, "An iterative solution for the optimal poles in a Kautz series," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing*, Salt Lake City, UT, 2001, pp. 3949–3952.

[31] N. Tanguy, R. Morvan, P. Vilbé, and L. C. Calvez, "Pertinent choice of parameters for discrete Kautz approximation," *IEEE Trans. Autom. Control*, vol. 47, no. 5, pp. 783–787, May 2002.

[32] A. Y. Kibangou, G. Favier, and M. Hassani, "Generalized orthonormal basis selection for expanding quadratic Volterra filters," in *Proc. 13th IFAC Symp. Syst. Ident.*, Rotterdam, The Netherlands, 2003, pp. 1119–1124.

[33] R. J. G. B. Campello, G. Favier, and W. C. Amaral, "Optimal expansions of discrete-time Volterra models using Laguerre functions," *Automatica*, vol. 40, no. 5, pp. 815–822, 2004.

[34] A. C. den Brinker and B. E. Sarroukh, "Pole optimisation in adaptive Laguerre filtering," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Processing*, Montreal, QC, Canada, 2004, pp. 649–652.

[35] A. da Rosa, W. C. Amaral, and R. J. G. B. Campello, "Choice of free parameters in expansions of discrete-time Volterra models using Kautz functions," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005, [CD ROM].

[36] A. Y. Kibangou, G. Favier, and M. Hassani, "Iterative optimization method of GOB-Volterra filters," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005, [CD ROM].

[37] A. Y. Kibangou, G. Favier, and M. Hassani, "Laguerre-Volterra filters optimization based on Laguerre spectra," *EURASIP J. Appl. Signal Processing*, vol. 2005, no. 17, pp. 2874–2887, 2005.

[38] A. Y. Kibangou, G. Favier, and M. Hassani, "Selection of generalized orthonormal bases for second-order Volterra filters," *Signal Processing*, vol. 85, no. 12, pp. 2371–2385, 2005.

[39] S. C. Patwardhan and S. L. Shah, "From data to diagnosis and control using generalized orthonormal basis filters. Part I: Development of state observers," *J. Process Control*, vol. 15, no. 7, pp. 819–835, 2005.

[40] R. J. G. B. Campello, W. C. Amaral, and G. Favier, "A note on the optimal expansion of Volterra models using Laguerre functions," *Automatica*, vol. 42, no. 4, pp. 689–693, 2006.

[41] A. da Rosa, R. J. G. B. Campello, and W. C. Amaral, "Choice of free parameters in expansions of discrete-time Volterra models using Kautz functions," *Automatica*, vol. 43, no. 6, pp. 1084–1091, 2007.

[42] A. da Rosa, R. J. G. B. Campello, and W. C. Amaral, "An optimal expansion of Volterra models using independent Kautz bases for each kernel dimension," *Int. J. Control*, vol. 81, no. 6, pp. 962–975, 2008.

[43] A. da Rosa, R. J. G. B. Campello, and W. C. Amaral, Exact Search Directions for Optimization of Linear and Nonlinear Models Based on Generalized Orthonormal Functions Dept. Comp. Sci., Univ. São Paulo (USP), São Paulo, Brazil, Tech. Rep., 2008 [Online]. Available: http://www.icmc.usp.br/~campello/Sub_Pages/Selected_Publications.htm

[44] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2002.
[45] *"Manual for Model 730—Magnetic Levitation System,"* Educational Control Products, 1999.
[46] P. Eykhoff, *System Identification: Parameter and State Estimation*. New York: Wiley, 1974.
[47] P. S. C. Heuberger, P. M. J. Van den Hof, and B. Wahlberg, *Modelling and Identification With Rational Orthogonal Basis Functions*. New York: Springer, 2005.
[48] F. J. Doyle, III, R. K. Pearson, and B. A. Ogunnaike, *Identification and Control Using Volterra Models*. New York: Springer-Verlag, 2002.
[49] O. Nelles, *Nonlinear System Identification*. New York: Springer-Verlag, 2001.
[50] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer-Verlag, 1999.
[51] W. J. Rugh, *Nonlinear System Theory—The Volterra/Wiener Approach*. Baltimore, MA: Johns Hopkins University Press, 1981.
[52] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. Melbourne, FL: Robert Krieger Publishing Company, 1980.
[53] N. Wiener, *Nonlinear Problems in Random Theory*. Cambridge, MA: The MIT Press, 1966.

**Alex da Rosa** was born in Goiânia-GO, Brazil, in 1980. He received the B.Sc. degree in electrical engineering from the Federal University of Goiás (UFG), Goiás, Brazil, in 2002, and the M.Sc. and Ph.D. degrees in electrical engineering from the School of Electrical and Computer Engineering, State University of Campinas (UNICAMP), Campinas, Brazil, in 2005 and 2009, respectively.

His research interests include the areas of modeling and identification of nonlinear dynamic systems.

**Ricardo J. G. B. Campello** (S'98–M'02) was born in Recife-PE, Brazil, in 1972. He received the B.Sc. degree in electronics engineering from the State University of São Paulo (Unesp), São Paulo, Brazil, in 1994, and the M.Sc. and Ph.D. degrees in electrical engineering from the School of Electrical and Computer Engineering, State University of Campinas (Unicamp), Campinas-SP, Brazil, in 1997 and 2002, respectively.

In 2002, he was a Visiting Scholar at the Laboratoire D'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S), Université de Nice-Sophia Antipolis (UNSA), France. Since 2007, he is with the Department of Computer Sciences, University of São Paulo (USP) at São Carlos. His current research interests include dynamic systems identification, computational intelligence, data mining, and machine learning.

**Wagner C. Amaral** (M'75) was born in Campinas, SP, Brazil, in 1952. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from the State University of Campinas, UNICAMP, in 1974, 1976, and 1981, respectively.

Since 1991, he has been a Full Professor with the Department of Computer Engineering and Industrial Automation, UNICAMP. From 1995 to 1999 he was the Director of the School of Electrical and Computer Engineering at UNICAMP. His research interests are in the areas of modeling, identification and predictive control.