System Identification Using Optimally Designed Functional Link Networks via a Fast Orthogonal Search Technique

Hazem M. Abbas
Mentor Graphics, Cairo, EGYPT
Email: h.abbas@ieee.org

***Abstract*— In this paper, a sparse nonlinear system identification method is proposed. Functional link neural nets (FLN) with their high orders polynomial basis functions are capable of performing complex nonlinear mapping. However, a large number of inputs and accurate modeling will require a huge number of basis functions that need to be explored. The Fast Orthogonal Search (FOS) introduced by Korenberg [1] is adopted here to detect the proper model and its associated parameters. The FOS algorithm is modified by first sorting all possible nonlinear functional expansion of the input pattern according to their correlation with the system output. The sorted functions are divided into equal size groups, pins, where functions with the highest correlation with the output are assigned to the first pin. Lower correlation members go the following pin and so forth. During the identification process, candidates in lower pins are tried first. If a solution is not found, next pins join the candidates pool for further modeling until the identification process completes within a prespecified accuracy. The proposed architecture is tested on noise-free and noisy nonlinear systems and shown to find sparse models that can approximate the experimented systems with acceptable accuracy.**

***Index Terms*— neural networks, functional link networks, orthogonal search, system identification, nonlinear mapping**

I. INTRODUCTION

Identification of nonlinear dynamic systems is of considerable importance in many engineering applications such as echo cancellation [2], device modeling [3], nonlinear filter design [4] among others. The use of neural networks emerges as a viable solution in the identification problem. The universal approximation properties of neural networks [5] make them a useful tool for modeling nonlinear systems. The problem of nonlinear modeling using static neural networks has been extensively researched [6] and many approaches have used multilayer perceptrons (MLP) and radial basis functions (RBF) [7]–[9]. Feedforward neural networks have been employed in system identification using a set of delayed inputs and outputs [7]. Multilayer perceptrons trained using the backpropagation (BP) algorithm for system modeling was presented in [6]. BP nets have shown to be robust and effective in modeling complex dynamic systems.

Other neural models also include radial basis function nets [9], [10] which can learn functions with local variations and can provide universal approximation capabilities [11]. Nonlinear systems identification using RBF net has been proposed in [7], [12]. Linear orthogonal neural network with Lagendre polynomials were also shown to approximate various complex functions [13].

Functional link networks (FLN) [14] replaces the hidden layer in MLP by providing nonlinear function expansion of the network input using functional links such as polynomial basis functions, Chebyshev polynomials or Hermite polynomials. The net output is composed of a linear sum of the the basis functions used. FLNs have proved capable of approximating nonlinear mappings and were shown to successfully model nonlinear systems [7]. Network parameters can be computed by directly solving a set of linear equations [14] or by employing other optimization techniques [15].

In system identification, the objective twofold: to select the model structure and to estimate the parameter values of the model. In some applications, the modeled nonlinear system involves large time delays and therefore most of the model terms (or kernels) are actually not contributing to the system output and thus resulting in a sparse model representation. When a FLN is trained to identify a nonlinear system, all candidate terms represented by the functional links are treated equally and the complete set of parameters are to be estimated. The value of all terms are estimated although only a few terms are needed to be identified and thus resulting in a considerable waste of computing power. In addition, the estimated model parameters will become inaccurate since a large number of insignificant terms are involved. The main purpose of this paper is to introduce a method that selects the proper FLN basis functional that will model a given system within an acceptable error.

Many methods have been proposed to address the model selection problem such as evolutionary algorithms [16], [17] and orthogonal least squares methods [1], [18]. Evolutionary techniques rely on applying methods borrowed from nature by applying genetic operators to find the correct model. Orthogonal techniques use the pool of candidate terms to calculate a new set of orthogonal

This paper is based on "A Novel Fast Orthogonal Search Method for Design of Functional Link Networks and their Use in System Identification," by H. Abbas, which appeared in the Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Montreal, Canada, 2007.

terms that reduce the squared error between the system and the model.

This paper presents a method to construct a minimum FLN network with polynomial functional links by employing a modified FOS [1] that sorts the candidate terms according with their correlation with the identified output so that terms with the highest correlation coefficients are used to form the orthogonal space. Analysis and simulation results of the proposed method demonstrate the efficacy of the method in finding the optimal design of FLN in a time much shorter than when the conventional FOS is used.

The paper is organized as follows. The characterization of the system identification problem is presented in section II. Section III presents a description of using FLN nets for system identification. Section IV reviews the FOS algorithm while section V introduces the sorted version of the FOS and its utilization to find the minimum polynomial expansions of the FLN. Analysis and experimental results of the proposed networks are presented in section VI.

II. The Nonlinear System Identification Problem

The nonlinear system identification problem is depicted in Fig. 1. The system has an input signal, $x(n)$, and produces an output, $y(n)$, $n = 1, \cdots, N$, where N is the record length. The identification model should

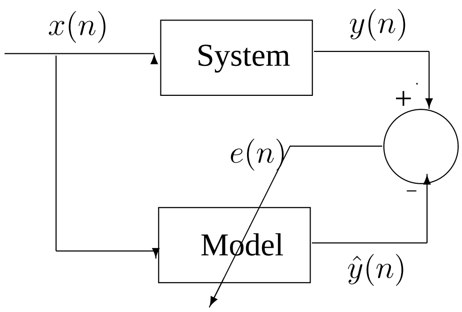

Fig. 1. Identification System

approximate the system output when excited with the same input by minimizing the cost function

$$J = \frac{1}{N} \sum_{i=1}^{N} e(i)^2 \quad (1)$$

where

$$e(i) = y(i) - \hat{y}(i)$$

is the estimation error between the observed output, $y(n)$, and the modeled output, $\hat{y}(n)$, within an acceptable accuracy. A model that is described by nonlinear autoregressive with exogenous inputs (NARX) can be expressed as

$$\hat{y}(n) = F\Big(y(n-1), \dots y(n-K); x(n), \dots, x(n-L)\Big) \quad (2)$$

where $F(.)$ is a nonlinear function to be determined, and K and L are maximum lags in the input and output, respectively. The estimation error, $e(n)$, is considered to be independent and identically distributed (*iid*) noise. When neural nets are used as the model, the nonlinearity, F, is replaced by the output of the sigmoidal hidden layer(s) in MLP [7], Gaussian basis functions in RBF [19], Chebyshev basis functions [20], or higher-degree polynomial in FLN [21]. An expansion of (2) using a polynomial type nonlinearity will produce a huge number of linear and nonlinear terms that will be proportional with the values of the system lags and the oder of of nonlinearity. In practice, a lot of those terms may be redundant. Orthogonal search methods [1], [22] are found to be very effective in selecting the proper candidate terms and estimating their associated values. Expressing the model in (2) with the regression model

$$y(n) = \sum_{i=0}^{M} w_i \; p_i(n) + e(n) \quad (3)$$

where M is the number of unknown terms, $p_i(n)$ represents a selected term in the NARX model, and w_i is the parameter associated with this term. Orthogonal search techniques normally minimize the cost function (1) by a finding a proper candidate terms through an orthogonalization process.

III. Function Approximation Using FLNs

FLN nets approximate a desired single output of a sparse system using a small set of basis functions.

Definition 1: The output of a multiple-input single-output FLN (Figure 2) network can be expressed as

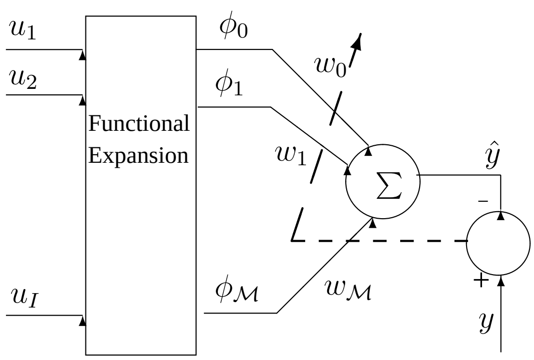

Fig. 2. Functional Link Network Structure

$$\hat{y}(n) = \sum_{i=0}^{\mathcal{M}} w_i \; \phi_i(n)$$

where $\mathbf{\Phi} = \{\phi_i\}_{i=0}^{\mathcal{M}}$ is a set of linearly independent basis functions, $\phi_i = F_i\Big(u(0), \dots u(n)\Big)$ is the ith functional expansion of the set of inputs, $\phi_0 = 1$, $U = \{u_i\}_{i=0}^{I}$, $\Omega = \{w_i\}_{i=0}^{\mathcal{M}}$ is the set of associated weights with the functional link set, Φ, and $\mathcal{M}$ is the total number of functional links that can be generated by the network.

The design of the FLN requires finding an optimal set of weight, Ω, that will result in a minimum modeling error, J. This can be achieved by solving a set of simultaneous linear equations or using a gradient descent method that will iteratively find the needed weight values.

In this work, we will be dealing with multiple-input single-output FLNs. Multidimensional function approximation using FLN nets have been introduced in [23] and analyzed in more details in [21].

A. Linear System Solution Approach

Assume that there is an N number of input–output pattern pairs to be learned by the FLN, and the the input vector, $U \in R^d$, is composed of all possible d lags in the input and output, i.e., $d = L + K$, while the resultant output, $\hat{y}$ is a scalar value. Each of the input pattern is passed through a functional expansion block producing a corresponding $\mathcal{M}+1$-dimensional $(\mathcal{M} \geq d)$ expanded vector. Considering all N patterns, input–output relationship may be expressed as

$$\hat{\mathbf{y}} = \mathbf{\Phi}\ \Omega^T \tag{4}$$

where $\mathbf{\Phi} = [\phi_0\ \ \phi_1 \cdots \phi_{\mathcal{M}+1}]$ is a $N \times (\mathcal{M}+1)$ dimensional matrix, ϕ_i is an N-dimensional vector representing the ith basis function, and $\hat{\mathbf{y}}$ is an N-dimensional estimated output. In order to find the weight vector, Ω, a number N of simultaneous equations need to be solved. If the basis functions were chosen to be formed as nonlinear polynomials of the lagged input and output samples, the weights can be calculated using the least square error method based on the vector of the output signals, y, and the regressor data represented by $\mathbf{\Phi}$, i.e.,

$$\Omega = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}$$

Since the least square methods consider each polynomial term equally important and estimate all associated weights even when only a small subset of those polynomials, i.e., M, $M << \mathcal{M}$, need to be identified (which is the case with in sparse systems). Hence, this amounts to a huge waste of computation resources. Additionally, estimating a large number of insignificant terms will introduce inaccuracies in the estimated terms.

B. Gradient Descent Iterative Approach

The gradient descent algorithm can be applied to find the optimal set of weights, Ω, that makes the estimated error, J in (1). Starting with a set of random values, $\Omega(0)$, the weight set is updated using the rule

$$\Delta\Omega(i+1) = \Omega(i) + \Delta\Omega(i+1)$$

where $\Delta\Omega(i+1) = -\eta\frac{\partial J}{\partial\Omega(i)}$ is the incremental update to the vector Ω at iteration $i+1$ and η ia a suitable learning rate. Since the system in (3) is totally linear, the derivative in the updating rule will become zero when the weights reach their optimal values. Although the iterative approach is guananteed to reach an optimal solution, it will consume extensive computing power to find a large number of redundant weights, the ones with near zero value.

C. Orthogonal Least Squares (OLS) Approach

In order to alleviate the problems encountered by the previous two approaches, orthogonal least squares (OLS) search techniques [18], [22], [24] apply forward stepwise model selection techniques to find the most significant terms that can be used to fairly represent the system under study.

It works by applying an orthogonal decomposition of $\mathbf{\Phi}$ using a Gram-Schmidt orthogonalisation process which ensures that each new column added to a new matrix, $\mathbf{S}$, is orthogonal to all previously selected columns and $\mathbf{\Phi} = \mathbf{S}\Gamma$ where Γ is an $\mathcal{M} \times \mathcal{M}$ upper triangular matrix. Since the space spanned by the set of orthogonal basis matrix, $\mathbf{S}$, is the same as the one spanned by the original matrix,$\mathbf{\Phi}$, the model in (4) can be expressed as

$$\hat{\mathbf{y}} = \mathbf{S}\ \mathbf{g}^T$$

where $\mathbf{S}$ is the orthogonal transformation of $\mathbf{\Phi}$ and the new paramter set, $\mathbf{g}$, is computed in such a way that the error in (1) will be minimum. All terms are examined to determine the amount of contribution of each term in modeling the desired system and the terms with the largest contribution in the error reduction process is selected first. The orthogonalization process is repeated to select next terms.

Several methods have been proposed to improve the OLS techniques such as Korenberg's Fast Orthogonal Search (FOS) [1], Chng et. al. [25], McGaughey et el. [26], Chen and Wigger [27], Adeney and Korenberg [28], Tsang and Chan [29], and Billings and Wei [30].

In this work, a variation in Korenberg's FOS is proposed to make it more computationally efficient in finding the exact model. In the following, the FOS algorithm is reviewed.

IV. Fast Orthogonal Search (FOS) [1]

Assuming that the functional expansion of the FLN is provided by polynomial basis functions of the set of lagged inputs and outputs, the FLN output can be expressed as a time series

$$y(n) = \sum_{i=0}^{M} w_i\ \phi_i(n) + e(n) \tag{5}$$

where $\phi_0(n) = 1$ and for $m \geq 1$,

$$\phi_m(n) = x(n-l_1)\cdots x(n-l_j)y(n-k_1)\cdots y(n-k_i),$$

and

$$j \geq 0, 0 \leq l_1 \leq L, \cdots, 0 \leq l_j \leq L,$$

$$i \geq 0, 0 \leq k_1 \leq K, \cdots, 0 \leq k_i \leq K.$$

Using the orthogonal search method, the model in (5) is expressed as

$$y(n) = \sum_{i=0}^{M} g_i\ s_i(n) + e(n) \tag{6}$$

The new orthogonal basis, $s_i(n)$, are constructed from the $\phi_i(n)$ using the Gram-Schmidt procedure

$$s_0(n) = 1,$$

$$s_m(n) = \phi_m(n) - \sum_{r=0}^{m-1} \alpha_{mr}\ s_r(n), m = 1, \cdots, M \quad (7)$$

so that they are orthogonal over the observation period of the output where

$$\alpha_{mr} = \frac{\overline{\phi_m(n) s_r(n)}}{\overline{s_r^2(n)}}, \quad r = 0, \cdots, m-1 \quad (8)$$

and the overbar denotes the time average. The parameters, g_i, are selected to minimize the mean–squared error over the interval, i.e.,

$$J = \overline{e^2(n)} = \overline{\left(y(n) - \sum_{i=0}^{M} g_i\ s_i(n)\right)^2} \quad (9)$$

Exploiting the orthogonality property

$$\overline{s_i(n) s_j(n)} = 0, i \neq j$$

and setting $\frac{\partial J}{\partial g_i}$ to zero, the optimal value for g_r is

$$g_r = \frac{\overline{y(n) s_r(n)}}{\overline{s_r^2(n)}}. \quad (10)$$

In addition, the error can be reduced to the form

$$J = \overline{y^2(n)} - \overline{\sum_{i=0}^{M} g_i^2\ s_i^2(n)} \quad (11)$$

which clearly shows that the addition of a new term, $g_r\ s_r(n)$, will reduce the error by the amount

$$Q(r) = \overline{g_r^2\ s_r^2(n)}. \quad (12)$$

The M selected basis functions are those that produce the largest $Q(m)$ in (12). Finally, the weight values associated with the selected functional links, w_i, can be calculated using the following [31]:

$$\begin{aligned} w_m &= \sum_{i=m}^{M} g_i v_i && (13) \\ v_m &= 1, \\ v_i &= -\sum_{r=m}^{i-1} \alpha_{ir} v_r, \quad i = m+1, \cdots, M \end{aligned}$$

This completes the identification of the FLN basis functions needed to represent the sparse system and their associated weight values. In order to expand the model, it is required to calculate the quantity Q for all candidates and choose the one for which Q is the greatest. Note also that the construction of the orthogonal functions, $s_i(n)$, is computationally intensive as it should be done for each candidate term [24]. Also, the need to create the orthogonal series, $s_i(n)$, is time consuming and wastes memory storage.

The Fast Orthogonal search (FOS) provides a faster mechanism by avoiding creating neither the basis functions, $\mathbf{\Phi}$, nor the orthogonal ones, $\mathbf{S}$. It achieves this objective by providing a set of recursive difference equations that eventually produce the required parameters, α_{ij} and w_i. It also avoids creating the basis functions by carrying out a common core of calculations to obtain input mean and autocorrelations and then doing small corrections to yield indiviual time-averages. The FOS starts by finding a recursive equation for (8). It defines as the numerator in (8) as

$$D(m, r) = \overline{\phi_m(n) s_r(n)}$$

and using (7),

$$\begin{aligned} D(m, r) &= \overline{\phi_m(n)\phi_r(n)} - \sum_{i=0}^{r-1} \alpha_{ri}\ D(m, i), && (14) \\ D(m, 0) &= \overline{\phi_m(n)}, \\ & \quad r = 1, \cdots, M; m = 1, \cdots, M && (15) \end{aligned}$$

Similarly,

$$\overline{s_m^2(n)} = \overline{\phi_m^2(n)} - \sum_{r=0}^{m-1} \alpha_{mr}^2\ \overline{s_r^2(n)}, m = 1, \cdots, M \quad (16)$$

and

$$E(m) = \overline{s_m^2(n)}, \quad E(0) = 1, m = 0, \cdots, M \quad (17)$$

This results in

$$\alpha_{mr} = \frac{D(m, r)}{E(r)}, r = 0, \cdots, m-1; m = 1, \cdots, M \quad (18)$$

This provides two recursive equations

$$\begin{aligned} D(m, r) &= \overline{\phi_m(n)\phi_r(n)} - \sum_{i=0}^{r-1} \frac{D(r, i)\ D(m, i)}{E(i)}, && (19) \\ & \quad r = 1, \cdots, m-1; m = 2, \cdots, M \\ E(m) &= \overline{p_m^2(n)} - \sum_{r=0}^{m-1} \frac{D^2(m, r)}{E(r)}, && (20) \\ & \quad m = 1, \cdots, M \end{aligned}$$

By repeating a similar procedure with the output, y, one will obtain another recursive equation,

$$C(m) = \overline{y(n)\phi_m(n)} - \sum_{r=0}^{m-1} \alpha_{mr}\ C(r), m = 1 \cdots, M \quad (21)$$

and $C(0) = 1$. Using (21) and (20),

$$\begin{aligned} g_m &= \frac{C(m)}{E(m)}, m = 0 \cdots, M && (22) \\ Q(m) &= g_m^2\ E(m) && (23) \\ \overline{e^2(n)} &= \overline{y^2(n)} - \sum_{m=0}^{M} g_m^2\ E(m) && (24) \end{aligned}$$

Finally, the weight values associated with the M selected functional links, w_i, can be calculated using (13). This completes the identification of the required FLN basis functions and their associated weight values.

The speed up offered by this algorithm is achieved by exploiting the lagged nature of the difference equation in the $\phi_i(n)$ terms that makes it possible to accelerate the calculations of the different time averages in the FOS algorithm. This is done by relating the time averages to input and output means and correlations and then by making small corrections for the finite record length. More details can be found in [32]. The method has shown to save a lot of computation time and memory storage. Also, the recursive equations in (20) and (21) requires calculations to be performed for the current candidate while values for previous candidates are reused.

V. SORTED FAST ORTHOGONAL SEARCH (SFOS)

The FOS can be enhanced further through different variations. This can be done by arranging the basis functions terms in such a way that most *probable* terms are selected first. A method was proposed in [33] where the terms are grouped into disjoints subsets that are searched sequentially. The set with linear x terms are searched first, followed by the y terms, and then by the nonlinear xx terms, the yy terms, and finally by the xy terms. A similar approach was proposed in [15] in their genetic evolution of a FLN by favoring simple models first. More complex nonlinear individuals are searched when the simpler linear terms become unable to fit the required mapping.

This work presents another modification to the FOS algorithm. The main idea of the proposed modification is to exploit the fact that basis functions with the highest correlation with output are more likely to be principal system terms [17]. Moving those terms that are highly correlated with the output higher up in the candidate list, will result in a much faster convergence to the minimum basis function architecture.

The algorithm starts by sorting the $\mathcal{M}$ candidate basis functions in $\mathbf{\Phi}$, in descending order according to their correlation with the output, $R_{\phi_i\, y}$, with ϕ_i being the ith functional link candidate and y is the system ouput. The sorted candidates are then grouped into a number of pins, V. Each pin is assigned an equal number of candidate functions, $P = \frac{\mathcal{M}}{V}$, where candidates with highest correlation with the output go the first pin and lower correlation terms are assigned to the following one and so forth. Then the conventional FOS is applied to operate on the candidates in the first pin. Obviously, the majority of the candidates needed to fully represent the system will be picked there and a great reduction in the representation error is expected. If a solution within acceptable accuracy cannot be attained after testing all first pin candidates, the members of the second pin are added to the candidate pool and the FOS algorithm continues until a final acceptable solution is reached.

A pseudo code of the algorithm could be as follows.

1. Start *SFOS Algorithm*
2. Given are: input, u, output, y,
   a number of sampled points, N
   input, L, output lags, K,
   and SNR if there is a meausurement noise
3. Calculate all members $\mathcal{M}$ candidates in $\mathbf{\Phi}$,
   and their correlation, $\{R_{\phi_i\, y}\}_{i=1}^{\mathcal{M}}$, with y
4. Sort candidates in descending order according to R,
5. Assign a number $P = \frac{\mathcal{M}}{V}$ of candidates to each of V pins
6. Add candidates in pin #1 to the candidates pool
7. Apply the FOS to the candidate pool
8. IF an accurate model has been constructed THEN exit
   ELSE add the candidates of next pin to the candidate pool,
   Repeat Step 7
9. end *Algorithm*

Theorem 1: The candidate with the highest contribution in the the model error reduction, $Q(i)$, is the one with highest correlation, $\rho^2(y, \phi_i)$, with the system output, y.

Proof: Since the term with the largest error reduction contribution, $Q(r)$, should be selected first, let us examine the procedure of selecting the first term. To calculate $Q(1)$, we start with the values of the first constant basis function,

$$m = 0,\ \ E(0) = 1,\ \ C(0) = \overline{y},\ \ D(1,0) = \overline{\phi_1(n)}$$

The reduction in error by the chosen term is,

$$Q(1) = \overline{g_1^2\ s_1^2(n)} = g_1^2\ E(1) = \frac{C^2(1)}{E(1)}$$

Using (21) and (20),

$$C(1) = \overline{y(n)\phi_1(n)} - \alpha_{10}C(0) = \overline{y(n)\phi_1(n)} - \overline{y(n)}\ \overline{\phi_1(n)}$$

$$E(1) = \overline{\phi_1^2(n)} - \alpha_{10}D(1,0) = \overline{\phi_1^2(n)} - [\overline{\phi_1(n)}]^2$$

This gives,

$$\begin{aligned} Q(1) &= \frac{\overline{y(n)\phi_1(n)} - \overline{y(n)}\ \overline{\phi_1(n)}}{\overline{\phi_1^2(n)} - [\overline{\phi_1(n)}]^2} \\ &= \frac{\overline{y(n)\phi_1(n)} - \overline{y(n)}\ \overline{\phi_1(n)}}{\sigma_{\phi_1}^2} \end{aligned} \quad (25)$$

where $\sigma_{\phi_1}^2$ is the variance of the term, $\phi_1(n)$. The correlation coefficient, $\rho(y, \phi_1)$, between the output and the tested term is defined as

$$\rho^2(y, \phi_1) = \frac{\overline{[(y(n) - \overline{y(n)})(\phi_1(n) - \overline{\phi_1(n)})]}^2}{\sigma_{\phi_1}^2\ \sigma_y^2}$$

It is straightforward to show that

$$\sigma_{\phi_1}^2\ Q(1) = \rho^2(y, \phi_1)\ \sigma_y^2\ \sigma_{\phi_1}^2$$

and hence

$$Q(1) \propto \rho^2(y, \phi_1) \quad (26)$$

Therefore, the candidate with the highest $Q(1)$, is the term which has the maximum $\rho^2(y, \phi_1)$ with the observed output. ■

This finding justifies the proposed ordering of the candidates. By removing the output contribution of this candidate from the original output and repeating the process with the remaining candidates, the above analysis will always select the candidate with the highest correlation

TABLE I
EXAMPLE 1: ORDERING OF BASIS FUNCTIONS BEFORE AND AFTER SORTING

No.	Candidate location	loc. after sorting
1	20	3
2	118	7
3	282	78
4	357	12
5	403	4
6	411	13
7	715	1
8	822	76
9	831	5
10	1495	2

with the new output. It should be noted that such ordering does not guarantee that all significant candidates will be placed high on the list. However, the CPU time needed by the FOS algorithm is the upper bound of the proposed SFOS. As it will be shown in the experiments, the SFOS will be shown to perform much faster than FOS.

VI. SIMULATION RESULTS AND ANALYSIS

The proposed FLN-SFOS identification algorithm has been tested using a set of experiments on both static and dynamic systems. We will describe two examples that demonstrate the performance of the algorithm in identifying a sparse 3-*rd* order dynamic system. The locations of the principal basis functions are chosen randomly. The corresponding weight values of the chosen functions are generated from a uniform distribution bounded by the upper and lower values of each kernel ($[-2.5, 2.5]$). The first example demonstrates the ability of the algorithm in correctly identifying the system driven by a white Gaussian input and noise-free measurement. The second example describes the algorithm performance when there is a white Gaussian measurement noise. In the experiments, the input sequence, $x(n)$, is drawn from a zero-mean unit-variance white Gaussian distribution and the output is observed for a record length of $500 + (K + L)$. The first $K + L$ samples are discarded to eliminate any error that might result in calculating the time averages.

A. Example 1: A third-order system with no measurement noise

The first example is of a 3–rd order system with time delays equal to $(L = 21; K = 0)$. This amounts to a total number of basis functions, $\mathcal{M} = 2024$. Only 10 candidates ($M = 10 \ll \mathcal{M}$) have been used to generate the output. A pin size of $P = 10$, is used which results in having 21 pins to try. Table I shows the ordering of the selected basis functions before and after the correlation-based sorting process. It is obvious that one basis function is linear and the rest are combinations of nonlinear terms in 2–*nd* and 3–*rd* order terms of x. The sorting process resulted in all significant terms to be grouped within the first pin (highest candidate is in the 78th position). If the conventional FOS is applied on this data, a number of 1495 terms has to be tested in order to completely reconstruct the original system. With the SFOS, the process is terminated after the 78th term is tested and even before trying all candidates in the pin. The order of selection of the basis functions by the SFOS is 1, 2, 3, 7, 5, 4, 12, 13, 78, and 76. Obviously, the terms with the highest $\rho^2(y, \phi_i)$ were selected first. When the FOS is applied on the system, it required 2.2969 sec of computing time. The proposed SFOS-FLN algorithm only took 0.093 sec. of computing time, a significant speed–up factor of 24. Expectedly, both algorithms reached the same representation error.

B. Example 2: A third-order system with Gaussian noise added

The previous experiment was repeated after adding a white Gaussian noise to the output with a SNR equal to 10 dB. As in the first example, the input used is a 500 data points drawn from a white Gaussian noise with zero-mean and unit variance. The measurement noise is independent from the input signal. Due to noise, it is expected that extra basis functions are added in the final solution as an attempt to fit the noise. The FOS managed to identify the original basis functions. However, the corresponding weights, w_i were different from the original one. Moreover, nine extra basis functions were added to the model in order to achieve a final mse error equal to 1.528. After candidate sorting, once again all candidates were placed in the first pin. Naturally, the exact locations within the pin differed from the one in the noise–free case due to the new correlation values caused by the noisy output. The SFOS managed to identify the 10 original candidates with exceptional accuracy in the weight values and only extracted just one extra false candidate with a very small weight value. However, the final mse error was 1.7, an 11% increase than the mse obtained by the FOS. The CPU time needed for the noisy case was 0.1094 sec. for the SFOS and 2.2969 sec. for the FOS, a speed–up of about 22. It is worth mentioning that the speed–up factor is dependent on the number of candidate functionals, $\mathcal{M}$, and the pin size, V.

The algorithms has been also tested with other real data and the reported results were similar to the ones discussed above.

It is worth mentioning that the SFOS-FLN algorithm can be extended to be applied in different areas such as selecting the best features in classification problems [34] to test the capability of the algorithm in producing nonlinear discriminant analysis functions based on the higher polynomials. It can be also employed to find efficient Volterra system identification models [35].

VII. CONCLUSIONS

In this paper, an approach, SFOS, of constructing FLN for sparse polynomial systems identification is presented. The proposed algorithm exploits the fact that FLN basis functions with the highest correlation with output are more probable to be principal system terms. The algorithm sorts all available candidates in descending order according to their correlation with the output and assigns them to different pins of fixed size. It has been shown that such an ordering will guarantee that principal terms will get selected first. The FOS algorithm is applied to pick up the correct candidates and calculates their associated weight values using orthogonal search and Cholskey decomposition. The SFOS algorithm examines first the basis function candidates in the first pin. If the selected functions become unable to produce an acceptable solution, subsequent candidates in the following pins are added to the pool. The process ends when a solution is found and the identified system is successfully reproduced. In the absence of any measurement noise, the algorithm has produced exact results when applied to sparse second and third order systems. For noisy outputs, the algorithm managed to detect the correct terms with a small error in the kernel values. In either case, the SFOS has shown to provide a considerable speed–up factor when compared with the FOS.

REFERENCES

[1] M. J. Korenberg, "A robust orthogonal algorithm for system identification and time–series analysis," *Biological Cybernetics*, vol. 60, pp. 267–276, 1989.

[2] O. Agazzi and D. G. Messerschmitt, "Nonlinear echo cancellation of data signals," *IEEE Transcation on Communication*, vol. COMM, no. 30, pp. 2421–2433, 1982.

[3] G. Stegmayer, "Volterra series and neural networks to model an electronic device nonlinear behavior," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN)*, 2004, pp. 2907–2910.

[4] J. D. Taft, "Quadratic linear filters for signal detection," *IEEE Transcation on Signal Processing*, vol. SP, no. 39, pp. 2557–2559, 1991.

[5] K. Funahashi, "On the approximate realization of continuous-mappings by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989.

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

[7] S. Chen, S. Billings, and P. Grant, "Nonlinear system identification using neural networks," *International Journal of Control*, vol. 51, no. 6, pp. 1191–1214, 1990.

[8] S. H. Tan, J. B. Hao, and J. Vandewalle, "Efficient identification of rbf neural-net models for nonlinear discrete-time multivariable dynamicalsystems," *Neurocomputing*, vol. 9, no. 1, p. 1126, 1995.

[9] J. Y. Choi, H. F. VanLandingham, and S. Bingulac, "A constructive approach for nonlinear system identification using multilayer perceptrons," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 26, no. 2, p. 307312, 1996.

[10] S. Elanyar and Y. Shin, "Radial basis fnction neural network for approximation and estimation of nonlinear stochastic dynamic systems," *IEEE Transactions on Neural Networks*, vol. 5, pp. 594–603, 1994.

[11] E. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered neural networks with gaussian hidden units as universal approximations," *Neural Comput.*, vol. 2, no. 2, pp. 210–215, 1990.

[12] V. Sunil Elanayar and Y. C. Shin, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems," vol. 5, no. 4, pp. 594–603, July 1994.

[13] S. Yang and C. Tseng, "An orthogonal neural network for function approximation," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 26, pp. 779–785, 1996.

[14] Y.-H. Pao, S. Phillips, and D. Sobajic, "Neural-net computing and intelligent control systems," *International Journal of Control*, vol. 56, no. 2, pp. 263–289, 1992.

[15] A. Sierra, J. A. Macias, and F. Corbacho, "Evolution of functional link networks," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 54–65, 2001.

[16] L. Yao, "Genetic algorithm based identification of nonlinear systems by sparse volterra filters," *IEEE Transactions on Signal Processing*, vol. 47, no. 12, pp. 3433–3435, 1999.

[17] H. Abbas and M. Bayoumi, "Volterra-system identification using adaptive real-coded genetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 36, no. 4, pp. 671–684, 2006.

[18] S. Chen, C. Cowan, and P. Grant, "Orthogonal least squares learning for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[19] S. Chen, S. Ching, and K. Alkadhimi, "Regularized orthogonal least squares algorithm for constructing radial basis function networks," *International Journal of Control*, vol. 64, no. 5, pp. 829–837, 1996.

[20] S. Purwara, I. Karb, and A. Jha, "On-line system identification of complex systems using chebyshev neural networks," *Applied Soft Computing*, vol. 7, no. 1, pp. 364–372, 2007.

[21] J. Patra, R. Pal, B. Chatterji, and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 29, pp. 254–262, 1999.

[22] S. Chen, S. Billings, and W. Luo, "Orthogonal least squares methods and their application to non linear system identification," *International Journal of Control*, vol. 50, pp. 1873–1896, 1989.

[23] T. Yamada and T. Yabuta, "Dynamic system identification using neural networks," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, pp. 204 – 211, 1993.

[24] M. J. Korenberg and L. D. Paarmann, "Orthogonal approaches to time–series analysis and system identification," *IEEE Signal Processing Magazine*, pp. 29–43, July 1991.

[25] E. Chng, S. Chen, and B. Mulgrew, "Efficient computational schemes for the orthogonal least squares algorithm," *IEEE Transactions on Signal Processing*, vol. 43, pp. 373 – 376, 1995.

[26] M. D.R., "Using the fast orthogonal search with first term reselection to find subharmonic terms in spectral analysis," *Annals of Biomedical Engineering*, vol. 31, pp. 741–751(11), June 2003.

[27] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Trans. Signal Processing*, vol. 43, pp. 1713–1715, July 1995.

[28] K. M. Adeney and M. J. Korenberg, "Iterative fast orthogonal search algorithm for mdl-based training of generalized single-layer networks," *Neural Netw.*, vol. 13, no. 7, pp. 787–799, 2000.

[29] K. M. Tsang and W. L. Chan, "A search algorithm for the identification of multiple inputs nonlinear systems using orthogonal least squares estimator," *Electrical Engineering*, vol. 88, pp. 357 – 365, 2006.

[30] S. A. Billings and H.-L. Wei, "Sparse model identification using a forward orthogonal regression algorithm aided by mutual information," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 306–310, 2007.

[31] M. J. Korenberg, S. Bruder, and P. McIlroy, "Exact orthogonal kernel estimation from finite data records: Extending wiener's identification of nonlinear systems," *Annals of Biomedical Engineering*, vol. 16, pp. 201–214, 1988.

[32] M. J. Korenberg, "Identifying nonlinear difference equation and functional expansion representations: The fast orthogonal algorithm," *Annals of Biomedical Engineering*, vol. 16, pp. 123–142, 1988.

[33] P. McIlroy, "Applications of nonlinear system identification," Master's thesis, Queen's University at Kingston, Ontario, Canada, 1986.

[34] K. Mao, "Fast orthogonal forward selection algorithm for feature subset selection," *IEEE Transactions on Neural Networks*, vol. 13, no. 5, pp. 1218–1224, Sep 2002.

[35] G.-O. Glentis, P. Koukoulas, and N. Kalouptsidis, "Efficient algorithms for volterra system identification," *IEEE Transactions on Signal Processing*, vol. 47, no. 11, pp. 3042–3057, Nov 1999.