

Research Article

Nonlinear System Identification Using Quasi-ARX RBFN Models with a Parameter-Classified Scheme

Lan Wang,¹ Yu Cheng,² Jinglu Hu,³ Jinling Liang,⁴ and Abdullah M. Dobaie⁵

¹Wuxi Institute of Technology, Wuxi 214121, China

²Ant Financial Services Group, Hangzhou 310099, China

³Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan

⁴School of Mathematics, Southeast University, Nanjing 210096, China

⁵Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Correspondence should be addressed to Jinling Liang; jinlliang@gmail.com

Received 15 June 2017; Accepted 6 November 2017; Published 11 December 2017

Academic Editor: Dimitri Volchenkov

Copyright © 2017 Lan Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Quasi-linear autoregressive with exogenous inputs (Quasi-ARX) models have received considerable attention for their usefulness in nonlinear system identification and control. In this paper, identification methods of quasi-ARX type models are reviewed and categorized in three main groups, and a two-step learning approach is proposed as an extension of the parameter-classified methods to identify the quasi-ARX radial basis function network (RBFN) model. Firstly, a clustering method is utilized to provide statistical properties of the dataset for determining the parameters nonlinear to the model, which are interpreted meaningfully in the sense of interpolation parameters of a local linear model. Secondly, support vector regression is used to estimate the parameters linear to the model; meanwhile, an explicit kernel mapping is given in terms of the nonlinear parameter identification procedure, in which the model is transformed from the nonlinear-in-nature to the linear-in-parameter. Numerical and real cases are carried out finally to demonstrate the effectiveness and generalization ability of the proposed method.

1. Introduction

Many real-world systems exhibit complex nonlinear characteristics and hence cannot be identified directly by linear methods. In the last two decades, nonlinear models such as neural networks (NNs), radial basis function networks (RBFNs), neurofuzzy networks (NFNs), and multiagent networks have received considerable research attention for nonlinear system identification [1–4]. However, from a user's point of view, the conventional nonlinear black-box models have been criticized mostly for not being user-friendly: (1) they neglect some good properties of the successful linear black-box modeling, such as the linear structure and simplicity [5, 6]; (2) an easy-to-use model is to interpret properties of nonlinear dynamics rather than being treated as vehicles for adjusting fit to the data [7]. Therefore, careful modeling is needed for a model structure favorable to certain applications.

To obtain the nonlinear models favorable to applications, a quasi-linear autoregressive with exogenous inputs (quasi-ARX) modeling scheme has been proposed with two parts included: a macro-part and a core-part [14]. As shown in Figure 1, the macro-part is a user-friendly interface favorable to specific applications, and the core-part is used to represent the complicated coefficients of the macro-part. To this end, by using Taylor expansion or other mathematical transformation techniques, a class of ARX-like interfaces is constructed as macro-parts, in which useful properties of linear models can be introduced, while their coefficients are represented by some nonlinear models such as RBFNs. In this way, a quasi-ARX predictor linear with input variable $u(t)$ can be further designed, where $u(t)$ in the core-part is replaced skillfully by an extra variable. Thereafter, a nonlinear controller can be generated directly from the quasi-ARX predictor, which is similar to the simple linear control method [15, 16]. In contrast, complex nonlinear controller design should

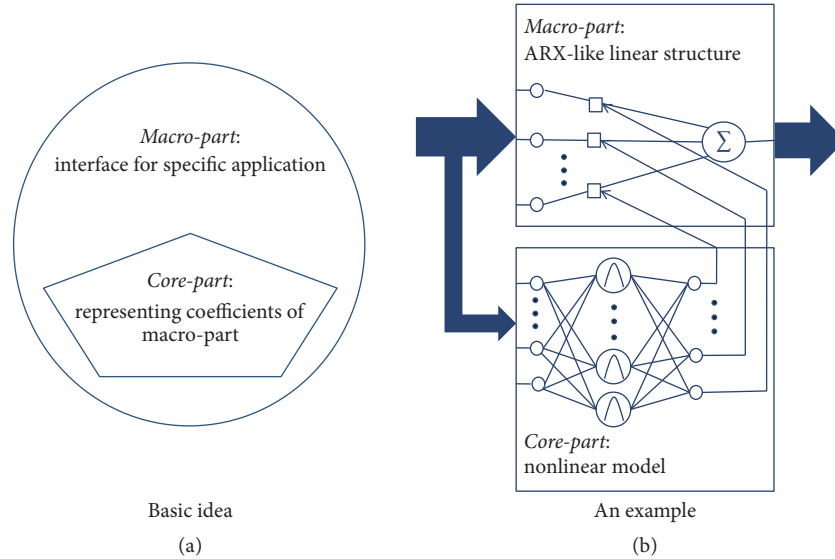


FIGURE 1: Quasi-ARX modeling. Basic idea of the quasi-ARX modeling is shown in (a), where a macro-part and a core-part are included in the constructed model. An example is illustrated in (b). An ARX-like linear structure works as macro-part for specific application, whose coefficients are parameterized by a flexible RBFN model.

be considered in NN based control methods, where two independent NNs are often contained: the one used for predictor and the other used for controller [17].

Actually, similar block-type models have been extensively studied and named in several forms according to their features, such as the state-dependent parameter models [18–20] and local linear models [10, 21]. Basically, identification methods can be categorized into three schemes:

- (1) Hierarchical identification scheme: quasi-ARX model structure can be considered as an “ARX submodel + NN” when NNs are utilized in the core-part [15, 16], and a hierarchical method has been proposed to identify the ARX submodel and the NN by a dual-loop scheme, where parameters in the ARX submodel are fixed and treated as constants in one loop, with the NN trained by a back propagation (BP) algorithm (only a small number of epochs are implemented); then the resultant NN is fixed to estimate the parameters of the ARX submodel in another loop. The two loops are executed alternatively to achieve a great approximation ability for nonlinear systems.
- (2) Parameter-classified identification scheme: when the nonlinear basis function models are embedded in the core-part of the quasi-ARX models, all the parameters can be classified as nonlinear (e.g., the center and width parameters in the embedded RBFNs) and linear (e.g., the linear weights in the embedded RBFNs) to the model. A structured nonlinear parameter optimization method (SNPOM) has been presented in [9] to optimize both the nonlinear and the linear parameters simultaneously for a RBF-type state-dependent parameter model, and improvement has been further given in [19, 22]. On the other hand, by using heuristic prior knowledge, the authors in [14, 23] estimate the

nonlinear parameters of a quasi-ARX NFN model, and the least square algorithm is used to estimate the linear parameters. Similarly, a prior knowledge has been used for nonlinear parameters in a quasi-ARX wavelet network (WN) model, where identification can be explained in an integrated approach [24, 25].

- (3) Global identification scheme: in this category, all the parameters in the quasi-ARX models are optimized regardless of the parameter features and model structure. For instance, a hybrid algorithm of particle swarm optimization (PSO) with diversity learning and gradient descent method has been proposed in [10] to identify the WN-type quasi-ARX model, which is always used in time series prediction. Moreover, NN [26] and support vector regression (SVR) [13] are applied, respectively, to identify all the quasi-ARX model parameters.

In this paper, specific efforts are made to extend the second identification scheme based on classifying the model parameters. Compared with the other schemes, this one explores the model properties deeply and provides a promising solution to a wide range of basis function embedded quasi-ARX models. It is known that SNPOM is an efficient optimization method fallen into this category, which makes good use of the model parameters feature and gives impressive performance in time series prediction and nonlinear control. However, this technique is still considered as a “non-transparent” approach since it is aimed at data-fitting only, and model parameters are difficult to be interpreted along with physical explanation of real world or nonlinear dynamics of systems [7]. Therefore, it may constrain further development of the model. In contrast, a prior knowledge based nonlinear parameter estimation makes sense to interpret system properties meaningfully, especially with respect to the

quasi-ARX RBFN model as discussed later in Section 3. The useful prior knowledge can evolve a quasi-ARX model from a “black-box” tool into a “semianalytical” one [27], which makes some parameters interpretable by our intuition, just following the principle of application favorable in quasi-ARX modeling. Owing to this fact, nonlinear parameters are determined in terms of prior interpretable knowledge, and linear parameters are adjusted to fit the data. It may contribute to low computational cost and high generalization of the model as parallel computation. Nevertheless, the problem is how to generate useful prior knowledge for an accurate nonlinear parameter estimation.

In the current study, a two-step approach is proposed to identify the quasi-ARX RBFN model for the nonlinear systems. Firstly, a clustering method is applied to generate the data distribution information for the system, whereby center parameters of the embedded RBFN are determined as cluster centers, and the width parameter of each RBF is set in terms of distance from other nearby centers. Then, it is straightforward to utilize the linear SVR for linear parameter estimation. The main purpose of this work is to provide an interpretable identification approach for the quasi-ARX models, which can be regarded as complementary to the identification procedures [6, 9, 13]. Compared with the heuristic prior knowledge used in quasi-ARX NFN model identification, the clustering based method gives an alternative approach to prior knowledge for nonlinear parameter estimation, and the quasi-ARX RBFN model is interpreted as a local linear model with interpolation. Moreover, when linear SVR is applied for linear parameter estimation, identification of the quasi-ARX RBFN model can be treated as an SVR with novel kernel mapping and associated feature space, and the kernel mapping is equivalent to the nonlinear parameter estimation procedure, which is transformed from a nonlinear-in-nature model to the linear-in-parameter one. Unlike the SVR-based method [9], the kernel function proposed in this study takes an explicit mapping, which is effective in coping with potential overfitting for some complex and noisy learning tasks [28]. Finally, in the proposed method, nonlinear parameters are estimated directly based on the prior knowledge; to some extent, it can be considered as an algorithmic approach for initialization of SNPOM.

The remainder of the paper is organized as follows. Section 2 introduces a quasi-ARX RBFN modeling scheme. Section 3 proposes the identification method of the quasi-ARX RBFN model. Section 4 investigates two numerical examples and a real case. Finally, some discussions and conclusions are made in Section 5.

2. Quasi-ARX RBFN Modeling

Let us consider a single-input-single-output (SISO) nonlinear time-invariant system whose input-output dynamics is described as

$$y(t) = g(\varphi(t)) + e(t), \quad (1)$$

where $\varphi(t) = [y(t-1), \dots, y(t-n_y), u(t-1), \dots, u(t-n_u)]^T$; $u(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}$, and $e(t) \in \mathbb{R}$ are the system input, output,

and a stochastic noise of zero-mean at time t , respectively; n_u and n_y are the unknown maximum delays of the input and output, respectively. $\varphi(t) \in \mathbb{R}^n$ with $n = n_y + n_u$ is the regression vector composed of the delayed input-output data. $g(\cdot)$ is an unknown function (black-box) describing the dynamics of system under study, which is assumed to be continuously differentiable and satisfies $g(0) = 0$.

Performing the Taylor expansion to $g(\varphi(t))$ at $\varphi(t) = 0$, one has

$$y(t) = e(t) + g(0) + (g'(0))^T \varphi(t) + \frac{1}{2} \varphi^T(t) g''(0) \varphi(t) + \dots \quad (2)$$

Then (1) is reformalized with an ARX-like linear structure:

$$y(t) = \varphi^T(t) \theta(\varphi(t)) + e(t), \quad (3)$$

where

$$\begin{aligned} \theta(\varphi(t)) &= g'(0) + \frac{1}{2} g''(0) \varphi(t) + \dots \\ &= [a_{1,t} \dots a_{n_y,t} \quad b_{0,t} \dots b_{n_u-1,t}]^T. \end{aligned} \quad (4)$$

In (4), coefficients $a_{i,t} = a_i(\varphi(t))$ and $b_{j,t} = b_j(\varphi(t))$ are nonlinear functions of $\varphi(t)$ for $i = 1, 2, \dots, n_y$ and $j = 0, 1, \dots, n_u - 1$; thus it can be represented by RBFN as

$$\theta(\varphi(t)) = \Omega_0 + \sum_{j=1}^M \Omega_j \mathcal{N}(p_j, \varphi(t)), \quad (5)$$

where p_j includes the center parameter vector μ_j and the width parameter σ_j of the j th RBF $\mathcal{N}(p_j, \varphi(t))$, M denotes the number of basis functions utilized, and $\Omega_j = [\omega_{1j}, \dots, \omega_{nj}]^T$ is a connection matrix between the input variables and the associated basis functions. According to (3) and (5), a compact representation of *quasi-ARX RBFN model* is given as

$$y(t) = \sum_{j=0}^M \varphi^T(t) \Omega_j \mathcal{N}(p_j, \varphi(t)) + e(t), \quad (6)$$

in which the set of RBFs with scaling parameter λ (the default value of λ is 1) is

$$\mathcal{N}(p_j, \varphi(t)) = \begin{cases} \exp\left(-\frac{\|\varphi(t) - \mu_j\|^2}{\lambda \sigma_j^2}\right) & j \neq 0 \\ 1 & j = 0. \end{cases} \quad (7)$$

3. Parameter Estimation of Quasi-ARX RBFN Model

From (6) and (7), it is known that p_j (i.e., μ_j, σ_j) for $j = 1, \dots, M$ and are M nonlinear parameters for the model, whereas Ω_j ($j = 0, \dots, M$) become linear when all the nonlinear parameters are determined/fixed. In the following, the clustering method and SVR are, respectively, applied to estimate those two types of parameters.

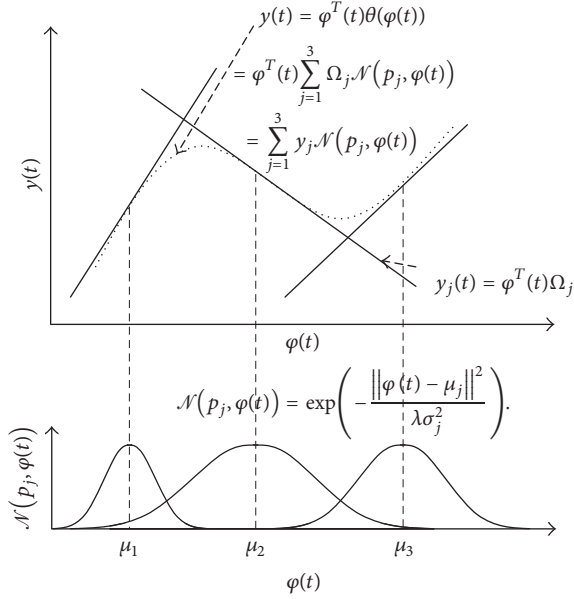


FIGURE 2: A local linear interpretation for the quasi-ARX RBFN model. A one-dimensional nonlinear system is approximated by three linear submodels, whose operating areas are decided by the associated RBFs; meanwhile, the RBFs also provide interpolations or weighs for all the linear submodels dependent on the operating points. A main interpolation is obtained from a linear submodel when the operating point is near to the center of the associated RBF, while only a minor one can be received when the operating point is far from the corresponding RBF.

3.1. Nonlinear Parameters Estimation. The choice of the center parameters plays an important role in performance of the RBF-type model [29]. In this paper, these parameters are estimated by means of prior knowledge from the clustering method rather than by minimizing the mean square of the training error. It should be mentioned that using the clustering method for initializing the center parameters is not a new idea in RBF-type models, and sophisticated clustering algorithms have been proposed in [30, 31]. In the present work, nonlinear parameters are estimated in a clustering way, which have meaningful interpretations. From this point of view, (6) is investigated as a local linear model with M submodels $y_j = \varphi^T(t)\Omega_j$ ($j = 1, \dots, M$), and the j th RBF $\mathcal{N}(p_j, \varphi(t))$ is regarded as a time-varying interpolation function for associated linear submodel to preserve the local property. Figure 2 gives a schematic diagram to illustrate the quasi-ARX RBFN model via a local linear mean.

In this way, the local linear information of the data can be generated by means of clustering algorithm, where the number of clusters (linear subspaces) is equivalent to the number of RBF neurons, and each cluster center is set as the center parameter of the associated RBF. In order to determine appropriately the operating area of each local linear submodel, width of each RBF is set to well cover the corresponding subspace. Generally speaking, we can set the width parameters of the RBF neurons according to the distances among those centers. For instance, a proper width

parameter σ_j of certain RBF can be obtained as a mean value of distances from its center μ_j to its nearest two others. From (7), one knows that an excessive small value of the width parameters may result in insufficient local linear operating areas for all data, while a wide-shape setting will make all the RBFs overlapped and hence the local property of each linear submodel is weakened.

Remark 1. Figure 2 only gives a meaningful interpretation of the model parameters. In real applications, since the data distribution is complex and the exact local linear subspaces may not exist, the clustering partition approach is used to provide several rational operating areas, and the scaling parameter λ can be set to adjust the width parameters for good weighting to each associated area.

3.2. Linear Parameters Estimation. After estimating and fixing the nonlinear parameters, (6) can be rewritten in a linear-in-parameter manner as

$$y(t) = \Phi^T(t)\Theta + e(t), \quad (8)$$

where $\Phi(t)$ is an abbreviation of $\Phi(\varphi(t))$ with

$$\Phi(t) = [\varphi^T(t), \mathcal{N}_1(t)\varphi^T(t), \dots, \mathcal{N}_M(t)\varphi^T(t)]^T, \quad (9)$$

$$\Theta = [\Omega_0^T, \Omega_1^T, \dots, \Omega_M^T]^T, \quad (10)$$

in which, since p_j in the j th RBF has already been estimated, we represent the j th RBF $\mathcal{N}(p_j, \varphi(t))$ by a shorten form as $\mathcal{N}_j(t)$ in (9). Therefore, the nonlinear system identification problem is reduced to a linear regression one with respect to $\Phi(t)$, and all the linear parameters are denoted by Θ .

Remark 2. As a result of nonlinear parameter estimation, $\Phi(t)$ plays an important role in transforming the quasi-ARX RBFN models from nonlinear-in-nature to linear-in-parameter with respect to Θ . Accordingly, it also transforms the nonlinear mapping from the original input space of $g(\cdot)$ into a high feature space; that is, $\varphi(t) \rightarrow \Phi(t)$. This explicit mapping will be utilized for an inner-product kernel in the later part.

In the following, the linear parameters are estimated by a linear SVR, considering the structural risk minimization principal as

$$\min \mathcal{J} \triangleq \frac{1}{2}\Theta^T\Theta + C\sum_{t=1}^N(\xi_t + \xi_t^*) \quad (11)$$

subject to

$$\begin{aligned} y(t) - \Phi^T(t)\Theta &\leq \epsilon + \xi_t, \\ -y(t) + \Phi^T(t)\Theta &\leq \epsilon + \xi_t^*, \end{aligned} \quad (12)$$

where N is the number of observations, $\xi_t \geq 0$ and $\xi_t^* \geq 0$ are slack variables, C is a nonnegative weight determining how much the prediction errors are penalized, which exceeds the

threshold value ϵ . The solution can be transformed to find a saddle point of the associated Lagrange function:

$$\begin{aligned} \mathcal{L}(\Theta, \xi_t, \xi_t^*, \alpha_t, \alpha_t^*, \beta_t, \beta_t^*) \\ \triangleq \frac{1}{2} \Theta^T \Theta + C \sum_{t=1}^N (\xi_t + \xi_t^*) \\ + \sum_{t=1}^N \alpha_t (y(t) - \Phi^T(t) \Theta - \epsilon - \xi_t) \\ + \sum_{t=1}^N \alpha_t^* (-y(t) + \Phi^T(t) \Theta - \epsilon - \xi_t^*) \\ - \sum_{t=1}^N (\beta_t \xi_t + \beta_t^* \xi_t^*), \end{aligned} \quad (13)$$

where α_t , α_t^* , β_t , and β_t^* are nonnegative parameters to be designed later. The saddle point could be acquired by minimizing \mathcal{L} with respect to Θ , ξ_t , and ξ_t^* :

$$\frac{\partial \mathcal{L}}{\partial \Theta} = 0 \implies \Theta = \sum_{t=1}^N (\alpha_t - \alpha_t^*) \Phi(t), \quad (14a)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_t^*} = 0 \implies \beta_t^* = C - \alpha_t^*, \quad (14b)$$

$$\frac{\partial \mathcal{L}}{\partial \xi_t} = 0 \implies \beta_t = C - \alpha_t. \quad (14c)$$

Thus, one can convert the primal problem (11) into an equivalent dual problem as

$$\begin{aligned} \max \mathcal{W}(\alpha_t, \alpha_t^*) \\ \triangleq -\frac{1}{2} \sum_{t,k=1}^N (\alpha_t - \alpha_t^*) (\alpha_k - \alpha_k^*) \Phi^T(t) \Phi(k) \\ + \sum_{t=1}^N (\alpha_t - \alpha_t^*) y(t) - \epsilon \sum_{t=1}^N (\alpha_t + \alpha_t^*) \end{aligned} \quad (15)$$

subject to

$$\sum_{t=1}^N (\alpha_t - \alpha_t^*) = 0, \quad \alpha_t, \alpha_t^* \in [0, C]. \quad (16)$$

To do this, the training results $\hat{\alpha}_t$ and $\hat{\alpha}_t^*$ are obtained from (15), and the linear parameter vector Θ is then obtained by the training value:

$$\Theta = \sum_{t=1}^N (\hat{\alpha}_t - \hat{\alpha}_t^*) \Phi(t). \quad (17)$$

In the above way, contributions of the SVR-based linear parameter estimation method can be concluded as follows.

- (1) The robust performance for parameter estimation is introduced because of the structural risk minimization of SVR.

- (2) There is no need to calculate the linear parameter Θ directly. Instead, it becomes a dual form of the quadratic optimization, which is represented by utilizing α_t and α_t^* depending on the size of the training data. It is very useful to alleviate the computational cost especially when the model suffers from the curse-of-dimensionality.
- (3) Identification of quasi-ARX model is specified as an SVR with explicit kernel mapping $\Phi(t)$, which has been mentioned in Remark 2. To this end, the quasi-ARX RBFN model is reformalized as

$$\begin{aligned} y(t) &= \Phi^T(t) \sum_{t'=1}^N (\hat{\alpha}_{t'} - \hat{\alpha}_{t'}^*) \Phi(t') \\ &= \sum_{t'=1}^N (\hat{\alpha}_{t'} - \hat{\alpha}_{t'}^*) \mathcal{K}(t, t'), \end{aligned} \quad (18)$$

where t' is time of training data, and a quasi-linear kernel, which is explicitly explained in the following remark, is defined as an inner product of the explicit nonlinear mapping $\Phi(t)$:

$$\begin{aligned} \mathcal{K}(t, t') &= \Phi^T(t) \Phi(t') \\ &= \varphi^T(t) \varphi(t') \sum_{i=0}^M \mathcal{N}_i(t) \mathcal{N}_i(t'). \end{aligned} \quad (19)$$

Remark 3. The quasi-linear kernel name is twofold. Firstly, it is derived from the quasi-ARX modeling scheme. Secondly, from (19) it is known that when M is as small as zero, the kernel is reduced to a linear one, and nonlinearity of the kernel mapping is improved when increasing the value of M . Compared with conventional kernels and with implicit kernel mapping, the nonlinear mapping of the quasi-linear kernel is turnable by M , which also reflects the nonlinearity of the quasi-ARX RBFN models in the sense of the number of local linear subspaces utilized. A proper value of M is essentially helpful to cope with the potential overfitting which will be shown in the following simulations.

4. Experimental Studies

In this section, identification performance of the above proposed approach to quasi-ARX RBFN model is evaluated by three examples. The first one is an example to show the performance of quasi-ARX RBFN model for time series prediction. Second, a rational system generated from Narendra and Parthasarathy [17] is simulated with a small amount of training data, which is used to demonstrate the generalization of the proposed quasi-linear kernel. At last, an example modeling a hydraulic robot actuator is carried out for a general comparison.

In the nonlinear parameter estimation procedure, affinity propagation (AP) clustering algorithm [32] is utilized to partition the input space and automatically generate the size of clusters in terms of data distribution, where Euclidean

distance is evaluated as the similarity between exemplars. Then, centers of all clusters are selected as the RBF center parameters in the quasi-ARX model, and the width parameter of a certain RBF is decided as the mean value of distances from the associated center to the nearest two others. For the linear parameter estimation, LibSVM toolbox [33] is applied, where ν -SVR is used with default ν setting by Matlab 7.6. Finally, the model performance is evaluated by root mean square error (RMSE) as

$$\text{RMSE} = \frac{\sqrt{\sum_t (y(t) - \hat{y}(t))^2}}{K}, \quad (20)$$

where $\hat{y}(t)$ is the prediction value of the system output $y(t)$ and K is the number of regression vectors.

4.1. Modeling the Mackey-Glass Time Series. The time series prediction on the chaotic Mackey-Glass differential equation is one of the most famous benchmarks for comparing the learning and generalization abilities of different models. This time series is generated from the following equation:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t), \quad (21)$$

where $a = 0.2$, $b = 0.1$, and $\tau = 17$, which are the most often used values in the previous research, and the equation does show chaotic behavior with them. To make the comparisons fair with the earlier works, we will predict $x(t+6)$ using the input variables $x(t)$, $x(t-6)$, $x(t-12)$, and $x(t-18)$. Two thousand data points are generated with initial condition taken as $x(t) \equiv 1.2$ for $t \in [-17, 0]$ based on the fourth-order Runge-Kutta method with time step $\Delta t = 0.1$. Then, one thousand input-output data pairs are selected from $t = 201$ to $t = 1200$, which is shown in Figure 3. The first 500 data pairs are used as training data, while the remaining 500 are used to predict $x(t+6)$ followed by

$$x(t+6) = \varphi^T(t) \Omega_0 + \sum_{j=1}^{\bar{M}} \varphi^T(t) \Omega_j \mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j, \varphi(t)) \quad (22)$$

with

$$\mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j, \varphi(t)) = \exp\left(-\frac{\|\varphi(t) - \hat{\mu}_j\|^2}{\hat{\sigma}_j^2}\right), \quad (23)$$

where $\varphi(t) = [x(t-18), x(t-12), x(t-6), x(t)]^T$.

The prediction of the Mackey-Glass time series using a quasi-ARX RBFN model starts, where 20 clusters are obtained from the AP clustering algorithm, and thus 20 RBF neurons are correspondingly constructed. Thereafter, SVR is used for linear parameter estimation, in which the super-parameter C is set as 100. The predicted result is compared with the original time series of test data in Figure 4, which gives a RMSE of 0.0091.

In Figure 4, the predicted result fits the original data very well; however, it is still not as good as the results from some famous models/methods listed in Table 1. Since

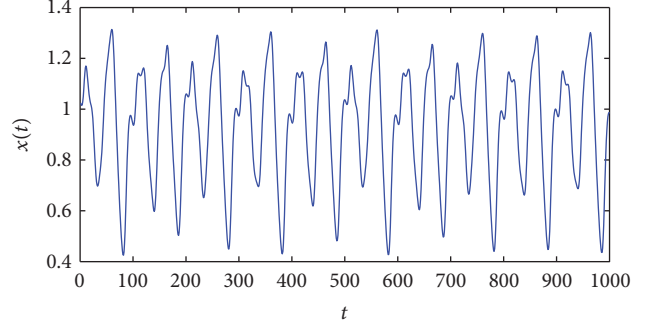


FIGURE 3: Time series generated from the Mackey-Glass equation.

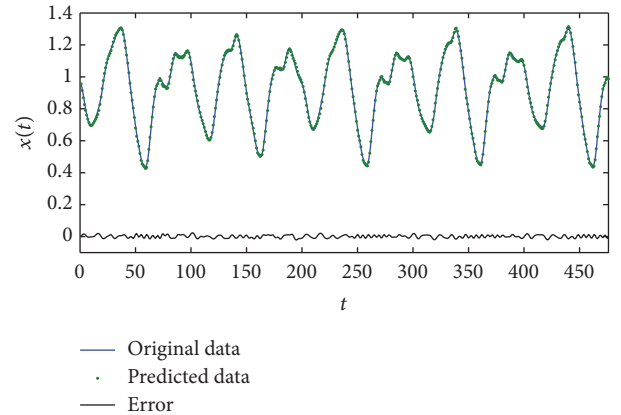


FIGURE 4: Prediction result with the quasi-ARX RBFN model.

no disturbance is contained in this example, it is found that the prediction performance can be easily improved by minimizing the training prediction error. In the comparison list, SNPOM for RBF-AR model, hybrid learning method for local linear wavelet neural network (LLWNN), and genetic algorithm (GA) for RBFN are all optimization-based identification methods, and it is relatively easy for them to achieve small RMSEs of the prediction by iterative training. However, these methods are much more time-costing in comparison with only 6 seconds by the proposed method for the quasi-ARX RBFN model. In addition, although the k -means clustering method for RBFN is implemented in a deterministic way and shows efficient result, the number of RBF neurons used is as big as 238. In fact, a small prediction RMSE obtained from these methods does not mean good identification of the models, since overtraining may happen some times.

In the present example, we confirm the effectiveness of the optimization-based method given above and propose a hybrid approach for identification of the quasi-ARX RBFN model, where prediction result from the proposed method can be further improved by SNPOM (the function “lsqnonlin” in the Matlab Optimization Toolbox is used [9]). It is seen that the prediction RMSE can be improved to 2.1×10^{-3} by only 15 iterations of implementation in SNPOM, and the result becomes compatible with others. However, such optimization is not always effective, especially in model

TABLE 1: Results of different models for Mackey-Glass time series prediction.

| Model | Method | Number of neurons | RMSE |
|----------------------|---------------------------------|-------------------|----------------------|
| Autoregressive model | Least square | 5 | 0.19 |
| FNT [8] | PIPE | Not provided | 7.1×10^{-3} |
| RBF-AR [9] | SNPOM | 25 | 5.8×10^{-4} |
| LLWNN [10] | PSO + gradient decent algorithm | 10 | 3.6×10^{-3} |
| RBF [11] | k -means clustering | 238 | 1.3×10^{-3} |
| RBF [12] | GA | 98 | 1.5×10^{-3} |
| Quasi-ARX RBFN model | Proposed | 20 | 9.1×10^{-3} |
| Quasi-ARX RBFN model | Proposed + SNPOM | 20 | 2.1×10^{-3} |

simulations on testing data, such as in the model $x(t) = f(\hat{x}(t-1), \hat{x}(t-2))$, where $\hat{x}(t-1)$ is the prediction value of $x(t-1)$. In the following, a rational system is evaluated by simulated quasi-ARX RBFN models to show advantages of the proposed method.

4.2. Modeling a Rational System. Accurate identification of nonlinear systems usually requires quite long training sequences which contain a sufficient amount of data from the whole operating region. However, as the amount of data is often limited in practice, it is important to study the identification performance for shorter training sequences with a limited amount of data. The system under study is a nonlinear rational model described as

$$y(t) = f(y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)) + e(t), \quad (24)$$

where

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2} \quad (25)$$

and $e(t) \in (0, 0.01)$ is the white noise.

Difficulty of this example lies in the fact that only 100 samples are provided for training, which is created by 100 random sequences distributed uniformly in the interval $[-1, 1]$, while 800 testing data samples are generated from the system with input:

$$u(t) = \begin{cases} \sin\left(\frac{2\pi t}{250}\right) & \text{if } t \leq 500 \\ 0.8 \sin\left(\frac{2\pi t}{250}\right) + 0.2 \sin\left(\frac{2\pi t}{25}\right) & \text{otherwise.} \end{cases} \quad (26)$$

The excited training signal $u(t)$ and system output $y(t)$ are illustrated in Figure 5.

In this case, 9 clusters are automatically obtained from the AP clustering algorithm; then the nonlinear parameters μ_j and σ_j ($j = 1, \dots, 9$) of the quasi-ARX RBFN model are estimated as Section 3 described. SVR is utilized thereafter for linear parameter estimation, where super-parameters are

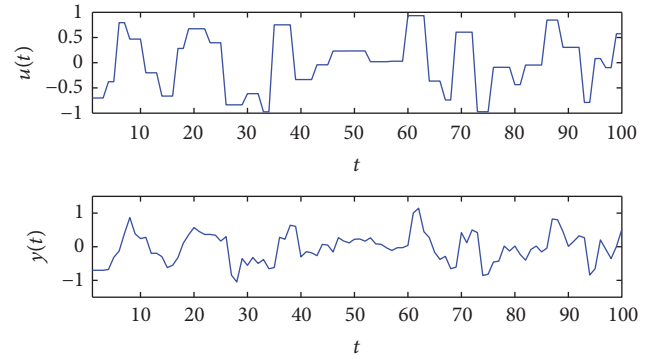


FIGURE 5: Training data for rational system identification.

set with different values for testing. Following the training, the simulated model is

$$y(t) = \varphi^T(t) \Omega_0 + \sum_{j=1}^{\bar{M}} \varphi^T(t) \Omega_j \mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j, \varphi(t)) \quad (27)$$

with

$$\mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j, \varphi(t)) = \exp\left(-\frac{\|\varphi(t) - \hat{\mu}_j\|^2}{\hat{\sigma}_j^2}\right), \quad (28)$$

where $\varphi(t) = [\hat{y}(t-1), \hat{y}(t-2), \hat{y}(t-3), u(t-1), u(t-2)]^T$ and $\hat{y}(t-n)$ denotes the simulated result in the previous n step. Figure 6 simulates the quasi-ARX RBFN model on the testing data, which gives a RMSE of 0.0379 under the super-parameter $C = 10$.

Due to the fact that identification of the quasi-ARX RBFN model can be regarded as an SVR with quasi-linear kernel, a general comparison is given to show advantages of the quasi-ARX RBFN model from SVR-based identification. Not only the short training sequence but also a long sequence with 1000 pairs of samples, which is generated and implemented in the same manner as the short one, is applied for comparing. Table 2 presents the comparison results of the proposed method (i.e., SVR with quasi-linear kernel), SVR with linear kernel, SVR with Gaussian kernel, and quasi-ARX model identified directly by an SVR (Q-ARX SVR), where various choices of SVR super-parameters C and γ for Gaussian kernel

TABLE 2: Simulated results of the SVR-based methods for rational system.

| Method | Super-parameters | | RMSE | |
|-----------------------|------------------|---------------------|-------------------------|------------------------|
| | C | γ (Gaussian) | Short training sequence | Long training sequence |
| Proposed | 1 | - | 0.0546 | 0.0287 |
| | 10 | - | 0.0379 | 0.0216 |
| | 100 | - | 0.0423 | 0.0216 |
| SVR + linear kernel | 1 | - | 0.0760 | 0.0710 |
| | 10 | - | 0.0764 | 0.0708 |
| | 100 | - | 0.0763 | 0.0708 |
| SVR + Gaussian kernel | 1 | 0.01 | 0.1465 | 0.0560 |
| | | 0.05 | 0.0790 | 0.0426 |
| | | 0.1 | 0.0808 | 0.0421 |
| | | 0.5 | 0.0895 | 0.0279 |
| | | 0.01 | 0.0782 | 0.0376 |
| | 10 | 0.05 | 0.0722 | 0.0409 |
| | | 0.1 | 0.0866 | 0.0365 |
| | | 0.5 | 0.0699 | 0.0138 |
| | | 0.01 | 0.0722 | 0.0352 |
| | | 0.05 | 0.0859 | 0.0376 |
| | 100 | 0.1 | 0.0931 | 0.0313 |
| | | 0.5 | 0.1229 | 0.0340 |
| | | 0.01 | 0.0698 | 0.0362 |
| | | 0.05 | 0.0791 | 0.0384 |
| | | 0.1 | 0.0857 | 0.0345 |
| Q-ARX SVR [13] | 1 | 0.5 | 0.0749 | 0.0116 |
| | | 0.01 | 0.0783 | 0.0412 |
| | | 0.05 | 0.0918 | 0.0328 |
| | 10 | 0.1 | 0.0922 | 0.0242 |
| | | 0.5 | 0.1483 | 0.0338 |
| | | 0.01 | 0.0872 | 0.0400 |
| | 100 | 0.05 | 0.1071 | 0.0237 |
| | | 0.1 | 0.8186 | 0.0166 |
| | | 0.5 | 0.1516 | 0.0487 |

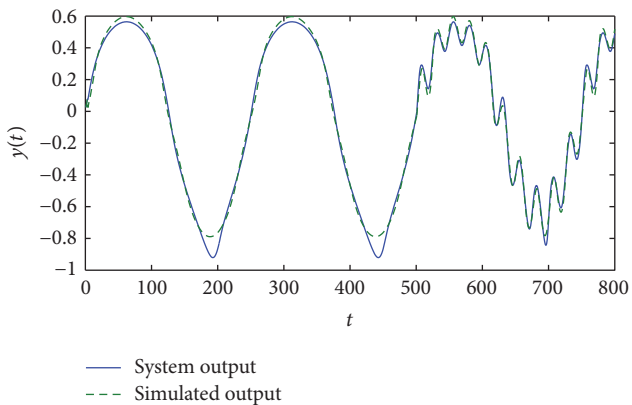


FIGURE 6: Simulated result with the quasi-ARX RBFN model for rational system.

are provided. From the simulation results under a short training sequence (100 samples), it is seen that when the design

parameters are optimized, SVR with quasi-linear kernel performs much better than the ones with Gaussian kernel and linear kernel, and the quasi-linear kernel also performs little sensitively with respect to the SVR super-parameter setting. Moreover, although the Q-ARX SVR method utilizes the quasi-ARX model structure, it only provides a similar simulation RMSE to SVR with Gaussian kernel. However, these simulation results cannot be resorted to refute the effectiveness of the SVR with Gaussian kernel and Q-ARX SVR method for nonlinear system identification. In the simulations, for a long training sequence (1000 samples), it is found that Q-ARX SVR method outperforms all the others, and SVR with Gaussian kernel also performs much better than the ones with quasi-linear and linear kernel.

On the other hand, from the perspective of the performance variation caused by different training sequences, histograms of simulated error for SVR-based methods are given in Figure 7, where performance of simulations is illustrated using, respectively, the short training sequence and the

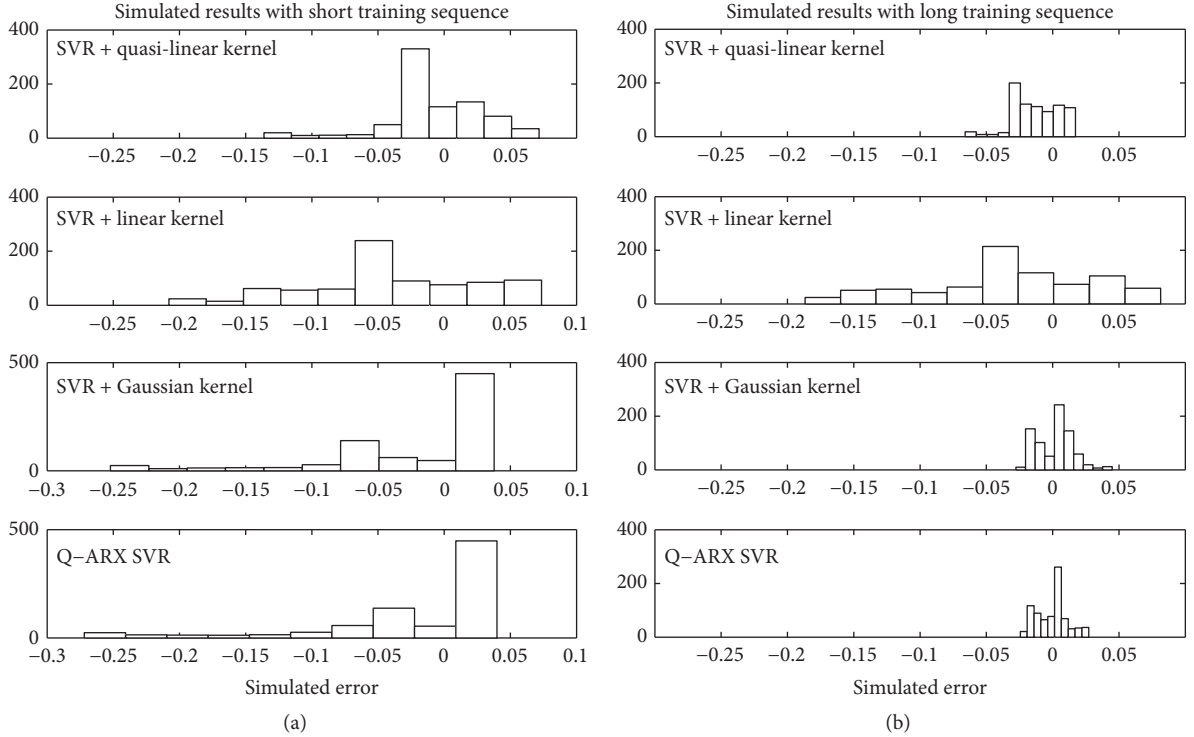


FIGURE 7: Histograms of the simulated errors. The horizontal coordinate in each subfigure denotes the simulated error of the model, whose elements are binned into 10 equally spaced containers. Four models are trained by using both short (a) and long (b) training sequences, then the simulated performance variation can be investigated by comparison.

long training sequence. It indicates that the SVR with linear kernel has the most robust performance to amount of training data, and the robust performance is also found in the quasi-linear kernel compared with the Gaussian kernel and Q-ARX SVR method, where significant deterioration is found in the simulations when a limited amount of training samples are used. This result implies that Gaussian kernel and Q-ARX SVR may be overfitted since the implicit nonlinear mapping is carried out, which has strong nonlinear learning ability but with no idea about how “strong” the nonlinearity need is. In contrast, the truth behind the impressive and robust performance of the quasi-linear kernel is that prior knowledge is utilized in the kernel learning (nonlinear parameter estimation), and a number of parameters are determined in terms of data distribution, where complexity of the model (nonlinearity) is tunable according to the number of local linear subspaces clustered. In other words, the quasi-ARX RBFN model performs in a local linear way; hence it can be trained in a multilinear way, better than some unknown nonlinear approaches for the situation with insufficient training samples.

Moreover, the RBF-AR model is utilized with SNPOM estimation method for this identification problem, where the number of RBF neurons are determined by trail-and-error, whose initial values are given randomly. Considering randomness of the algorithm, ten runs are implemented except that the results fail to be simulated, and the maximum

iterations value in SNPOM is set to 50. Consequently, four RBFs are selected for RBF-AR model, which gives a mean RMSE of 0.0696 using short training sequence, compared with the result of 0.0336 when the long training one is utilized. Although the parameter setting for this method may not be optimal, we can generate the same conclusion for the Q-ARX SVR method, which is overfitted in the case of training by short sequence.

4.3. Modeling a Real System. This is an example modeling a hydraulic robot actuator, where the position of the robot arm is controlled by a hydraulic actuator. The oil pressure in the actuator is controlled by the size of the valve opening through which the oil flows into the actuator. What we want to model is the dynamic relationship between the position of the valve $u(t)$ and the oil pressure $y(t)$.

A sample of 1024 pairs of $\{y(t), u(t)\}$ has been observed as shown in Figure 8. The data is divided into two equal parts, the first 512 samples are used as training data, and the rest are used to test the simulated model. For the purpose of comparison, the regression vector is set as $\varphi(t) = [y(t-1), y(t-2), y(t-3), u(t-1), u(t-2)]^T$. We simulate the quasi-ARX RBFN model on the testing data by

$$y(t) = \varphi^T(t) \Omega_0 + \sum_{j=1}^{\bar{M}} \varphi^T(t) \Omega_j \mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j, \varphi(t)) \quad (29)$$

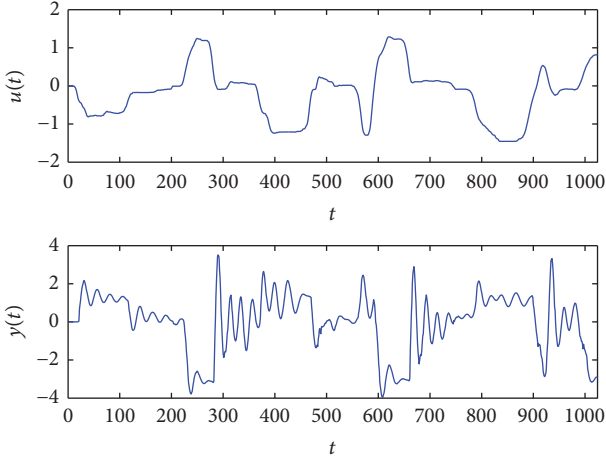
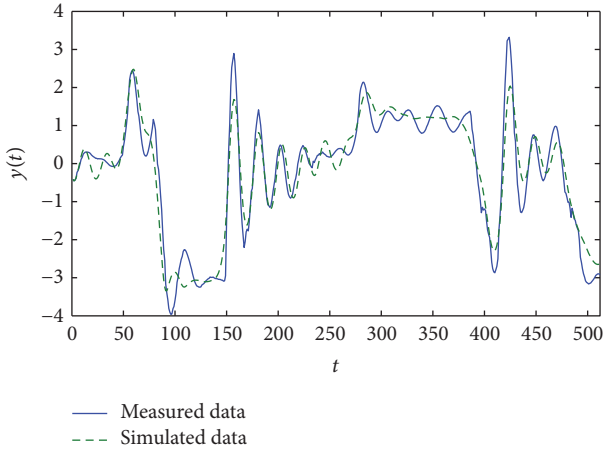
FIGURE 8: Measurements of $u(t)$ and $y(t)$.

FIGURE 9: Simulated result with quasi-ARX RBFN model for the real system.

with

$$\mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j, \varphi(t)) = \exp\left(-\frac{\|\varphi(t) - \hat{\mu}_j\|^2}{\lambda \hat{\sigma}_j^2}\right), \quad (30)$$

where $\varphi(t) = [\hat{y}(t-1), \hat{y}(t-2), \hat{y}(t-3), u(t-1), u(t-2)]^T$ and λ is set as 50 heuristically due to the complex dynamics and data distribution in this case, which insures that the RBFs are wide enough to cover the whole space well. Similar setting of λ can also be found in the literature for the same purpose [34, 35].

To determine the nonlinear parameters of the quasi-ARX RBFN model, AP clustering algorithm is implemented, and 11 clusters are generated automatically. Then, SVR is utilized for the linear parameter estimation. Finally, the model is identified and simulated in Figure 9 by the testing data, which gives a RMSE of 0.462. This simulation result is compared with the ones of linear ARX model, NN, WN, and SVR-based methods shown in Table 3. From Table 3, it is known that the proposed method outperforms the others for the real system. In addition, RBF-AR model with SNPOM estimation method

TABLE 3: Comparison results for the real system.

| Model | Super-parameters | | RMSE |
|---------------------------|------------------|---------------------|--------------|
| | C | γ (Gaussian) | |
| ARX model | - | - | 1.016 |
| NN [1] | - | - | 0.467 |
| WN [6] | - | - | 0.529 |
| SVR + quasi-linear kernel | 1 | - | 0.462 |
| | 5 | - | 0.487 |
| | 10 | - | 0.491 |
| SVR + Gaussian kernel | 1 | 0.05 | 1.060 |
| | | 0.1 | 0.828 |
| | | 0.2 | 0.643 |
| | | 0.5 | 1.122 |
| | | 0.05 | 0.850 |
| | | 0.1 | 0.740 |
| | 10 | 0.2 | 0.562 |
| | | 0.5 | 0.633 |
| | | 0.05 | 0.775 |
| | | 0.1 | 0.665 |
| | | 0.2 | 0.608 |
| | | 0.5 | 1.024 |
| Q-ARX SVR [13] | 1 | 0.05 | 0.737 |
| | | 0.1 | 0.592 |
| | | 0.2 | 0.801 |
| | 5 | 0.5 | 0.711 |
| | | 0.05 | 0.609 |
| | | 0.1 | 0.600 |
| | 10 | 0.2 | 0.715 |
| | | 0.5 | 0.890 |
| | | 0.05 | 0.593 |
| 10 | 0.1 | 0.632 | |
| | 0.2 | 1.231 | |
| | 0.5 | 1.285 | |

fails to be simulated in this case, where the number of RBF neurons is tested from 3 to 6, and their initial values are given randomly.

5. Discussions and Conclusions

The proposed method has a twofold role in the quasi-ARX model identification. For one thing, the clustering method has been used to uncover the local linear information of the dataset. Although similar methods have appeared in the parameter estimation of RBFNs, meaningful interpretation has been given here to the nonlinear parameters of quasi-ARX model in the manner of multilocal linear model with interpolations. In fact, explicit local linearity does not always exist in many real problems, whereas clustering can provide at least a rational multidimensional space partition approach. In the future, a more accurate and general space partition algorithm is to be investigated for identification of quasi-ARX models. For another, SVR has been utilized for the model's

linear parameter estimation; meanwhile, a quasi-linear kernel is deduced and performed as a composite kernel. The parameter M in the kernel function (19) corresponds to the amount of subspaces partitioned, which is therefore preferred not to be a big value to cope with the potential overfitting.

In this paper, a two-step learning approach has been proposed for identification of quasi-ARX model. Unlike the conventional black-box identification approaches, prior knowledge is introduced and makes sense for the interpretability of quasi-ARX models. By minimizing the training data error, linear parameters to the model are estimated. In the simulations, the quasi-ARX model is denoted in the form of SVR with quasi-linear kernel, which shows great approximation ability as optimization-based methods for quasi-ARX models but outperforms them when the training sequence is limited. Finally, the best performance of the proposed method has been demonstrated with a real system identification problem.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 81320108018 and 31570943 and the Six Talent Peaks Project for the High Level Personnel from the Jiangsu Province of China under Grant 2015-DZXX-003.

References

- [1] J. Sjöberg, Q. Zhang, L. Ljung et al., "Nonlinear black-box modeling in system identification: a unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.
- [2] I. Machon-Gonzalez and H. Lopez-Garcia, "Feedforward nonlinear control using neural gas network," *Complexity*, Article ID 3125073, 11 pages, 2017.
- [3] J. P. Noël and G. Kerschen, "Nonlinear system identification in structural dynamics: 10 more years of progress," *Mechanical Systems and Signal Processing*, vol. 83, pp. 2–35, 2017.
- [4] G. Nagamani, S. Ramasamy, and P. Balasubramaniam, "Robust dissipativity and passivity analysis for discrete-time stochastic neural networks with time-varying delay," *Complexity*, vol. 21, no. 3, pp. 47–58, 2016.
- [5] I. Sutrisno, M. A. Jami'in, J. HU, and M. H. Marhaban, "A self-organizing Quasi-linear ARX RBFN model for nonlinear dynamical systems identification," *SICE Journal of Control, Measurement, and System Integration*, vol. 9, no. 2, pp. 70–77, 2016.
- [6] J. Hu, K. Hirasawa, and K. Kumamaru, "A hybrid quasi-ARMAX modeling and identification scheme for nonlinear systems," *Research Reports on Information Science and Electrical Engineering of Kyushu University*, vol. 2, no. 2, pp. 213–218, 1997.
- [7] L. Ljung, *System Identification: Theory for the User*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1999.
- [8] Y. Chen, B. Yang, J. Dong, and A. Abraham, "Time-series forecasting using flexible neural tree model," *Information Sciences*, vol. 174, no. 3–4, pp. 219–235, 2005.
- [9] H. Peng, T. Ozaki, V. Haggan-Ozaki, and Y. Toyoda, "A parameter optimization method for radial basis function type models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 14, no. 2, pp. 432–438, 2003.
- [10] Y. Chen, B. Yang, and J. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, no. 4–6, pp. 449–465, 2006.
- [11] C. Harpham and C. W. Dawson, "The effect of different basis functions on a radial basis function network for time series prediction: a comparative study," *Neurocomputing*, vol. 69, no. 16–18, pp. 2161–2170, 2006.
- [12] H. Du and N. Zhang, "Time series prediction using evolving radial basis function networks with new encoding scheme," *Neurocomputing*, vol. 71, no. 7–9, pp. 1388–1400, 2008.
- [13] H. T. Toivonen, S. Tötterman, and B. Åkesson, "Identification of state-dependent parameter models with support vector regression," *International Journal of Control*, vol. 80, no. 9, pp. 1454–1470, 2007.
- [14] J. Hu, K. Kumamaru, and K. Hirasawa, "A quasi-ARMAX approach to modelling of non-linear systems," *International Journal of Control*, vol. 74, no. 18, pp. 1754–1766, 2001.
- [15] J. Hu and K. Hirasawa, "A method for applying neural networks to control of nonlinear systems," in *Neural Information Processing: Research and Development*, vol. 152 of *Studies in Fuzziness and Soft Computing*, pp. 351–369, Springer, Berlin, Germany, 2004.
- [16] L. Wang, Y. Cheng, and J. Hu, "Stabilizing switching control for nonlinear system based on quasi-ARX RBFN model," *IEEE Transactions on Electrical and Electronic Engineering*, vol. 7, no. 4, pp. 390–396, 2012.
- [17] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 1, no. 1, pp. 4–27, 1990.
- [18] P. C. Young, P. McKenna, and J. Bruun, "Identification of non-linear stochastic systems by state dependent parameter estimation," *International Journal of Control*, vol. 74, no. 18, pp. 1837–1857, 2001.
- [19] M. Gan, H. Peng, X. Peng, X. Chen, and G. Inoussa, "A locally linear RBF-network-based state-dependent AR model for nonlinear time series modeling," *Information Sciences*, vol. 180, no. 22, pp. 4370–4383, 2010.
- [20] A. Janot, P. C. Young, and M. Gautier, "Identification and control of electro-mechanical systems using state-dependent parameter estimation," *International Journal of Control*, vol. 90, no. 4, pp. 643–660, 2017.
- [21] A. Patra, S. Das, S. N. Mishra, and M. R. Senapati, "An adaptive local linear optimized radial basis functional neural network model for financial time series prediction," *Neural Computing and Applications*, vol. 28, no. 1, pp. 101–110, 2017.
- [22] M. Gan, H. Peng, and L. Chen, "A global-local optimization approach to parameter estimation of RBF-type models," *Information Sciences*, vol. 197, pp. 144–160, 2012.
- [23] Y. Cheng, L. Wang, and J. Hu, "Identification of quasi-ARX neurofuzzy model with an SVR and GA approach," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E95-A, no. 5, pp. 876–883, 2012.
- [24] Y. Cheng, *Study on Identification of Nonlinear Systems Using Quasi-ARX Models [Ph.D dissertation]*, Waseda University, June 2012.

- [25] Y. Cheng, L. Wang, and J. Hu, "Quasi-ARX wavelet network for SVR based nonlinear system identification," *Nonlinear Theory and Its Applications, IEICE*, vol. 2, no. 2, pp. 165–179, 2011.
- [26] B. M. Åkesson and H. T. Toivonen, "State-dependent parameter modelling and identification of stochastic non-linear sampled-data systems," *Journal of Process Control*, vol. 16, no. 8, pp. 877–886, 2006.
- [27] B.-G. Hu, H.-B. Qu, Y. Wang, and S.-H. Yang, "A generalized-constraint neural network model: associating partially known relationships for nonlinear regressions," *Information Sciences*, vol. 179, no. 12, pp. 1929–1943, 2009.
- [28] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, Mass, USA, 2001.
- [29] C. Panchapakesan, M. Palaniswami, D. Ralph, and C. Manzie, "Effects of moving the centers in an RBF network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 13, no. 6, pp. 1299–1307, 2002.
- [30] J. González, I. Rojas, H. Pomares, J. Ortega, and A. Prieto, "A new clustering technique for function approximation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 13, no. 1, pp. 132–142, 2002.
- [31] A. Guillén, H. Pomares, I. Rojas et al., "Studying possibility in a clustering algorithm for RBFNN design for function approximation," *Neural Computing and Applications*, vol. 17, no. 1, pp. 75–89, 2008.
- [32] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *American Association for the Advancement of Science: Science*, vol. 315, no. 5814, pp. 972–976, 2007.
- [33] C. Chang and C. Lin, "LIBSVM: a Library for support vector machines," <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [34] Y. Oussar and G. Dreyfus, "Initialization by selection for wavelet network training," *Neurocomputing*, vol. 34, no. 4, pp. 131–143, 2000.
- [35] Y. Bodyanskiy and O. Vynokurova, "Hybrid adaptive wavelet-neuro-fuzzy system for chaotic time series identification," *Information Sciences*, vol. 220, pp. 170–179, 2013.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

