Kernel Adaptive Filtering Subject to Equality Function Constraints

Badong Chen[1], Zhengda Qin[1], Nanning Zheng[1], José C. Príncipe[1,2]

1. Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, 710049, China
2. Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA

***Abstract*—Kernel adaptive filters (KAFs) are powerful tools for online nonlinear system modeling, which are direct extensions of traditional linear adaptive filters in kernel space, with growing linear-in-the-parameters (LIP) structure. However, like most other nonlinear adaptive filters, the KAFs are "black box" models where no prior information about the unknown nonlinear system is utilized. If some prior information is available, the "grey box" models may achieve improved performance. In this work, we consider the kernel adaptive filtering with prior information in terms of equality function constraints. A novel Mercer kernel, called the *constrained Mercer kernel* (CMK), is proposed. With this new kernel, we develop the *kernel least mean square subject to equality function constraints* (KLMS-EFC), which can satisfy the constraints perfectly while achieving significant performance improvement.**

Keywords—Kernel adaptive filtering; kernel least mean square; equality function constraints;

I. INTRODUCTION

Nonlinear system modeling finds a wide range of applications in many real-world problems and is still an active area of research. There are various nonlinear adaptive filters that can be used for nonlinear system modeling, among which the *kernel adaptive filters* (KAFs) [1] are very attractive because of their desirable features such as convexity, universal approximation, and online learning manner. The KAFs are developed by implementing the well-established linear adaptive filters in kernel space, building a growing *linear-in-the-parameters* (LIP) nonlinear model in the original input space. They also belong to a class of learning machines called *convex universal learning machines* (CULMs) [2]. So far many kernel adaptive filtering algorithms have been developed [3-19], among which the *kernel least mean square* (KLMS) [4] is the simplest, yet often most effective one.

The KAFs are "*black box*" models and take no prior information about the unknown system into consideration, like most other nonlinear adaptive filters. In many cases, however, some prior knowledge about the unknown system is available and can be incorporated into the model ("*grey box*" model) to improve the learning performance. A general way of expressing mathematically the prior knowledge about the system is through some kinds of constraints [20]. In recent years, a variety of constraints have been successfully incorporated into *artificial neural networks* (ANNs) to achieve improved learning performance [20-25]. In the present paper, we propose to incorporate some prior constraints into the KAFs. In particular, we consider incorporating the equality function constraints (EFC) into the KLMS algorithm. A similar approach has been applied in ANNs [20]. As stated in [20], the equality function constraints have some advantages over other equality constraints such as *boundary value constraints* (BVC) [25]. To solve the kernel adaptive filtering subject to equality function constraints, we propose a novel Mercer kernel, called in this paper the *constrained Mercer kernel* (CMK), which is defined by multiplying the original Mercer kernel by a weighting function corresponding to the sub-regions of the equality function constraints. Using this new kernel, we develop the *KLMS subject to equality function constraints* (KLMS-EFC). Of course, the proposed CMK can also be applied to other KAFs and other kernel methods such as SVM.

The rest of the paper is organized as follows. In section II, we briefly introduce the KLMS algorithm and describe the learning problem subject to equality function constraints. In section III, we propose the constrained Mercer kernel and develop the KLMS-EFC algorithm. In section IV, we present simulation results to demonstrate the desirable performance of the KLMS-EFC. Finally in section V, we present our conclusions.

II. KLMS AND EQUALITY FUNCTION CONSTRAINTS

A. KLMS

Given a sequence of input-output training examples $\{u_i, d_i\}$, $i = 1,2,\ldots$, where $u_i \in U \subset R^m$, $d_i \in R$,with U being the input domain, our goal is to learn a nonlinear mapping $f : U \to R$ that fits the data well under a specific learning criterion. Under the *mean square error* (MSE) criterion, this learning problem can be solved in an online manner (sample by sample) by using the KLMS algorithm [4]:

$$\begin{cases} f_0 = 0 \\ e_i = d_i - f_{i-1}(u_i) \\ f_i = f_{i-1} + \eta e_i \kappa(u_i, \cdot) \end{cases} \tag{1}$$

where f_i denotes an estimate of f at iteration i, e_i stands for the prediction error based on the last estimate f_{i-1}, $\eta > 0$ denotes the step-size parameter, and κ is a reproducing Mercer kernel function defined on $U \times U$, i.e. $\kappa : U \times U \to R$. The KLMS algorithm (1) is actually the least mean square (LMS) algorithm in kernel space, derived by transforming the input u_i into the reproducing kernel Hilbert space (RKHS) H_κ induced by the Mercer kernel κ and applying the LMS on the transformed data [4]. The widely adopted kernel is the Gaussian kernel:

$$\kappa(u,u') = \exp\left(-\frac{\|u-u'\|^2}{2\sigma^2}\right) \tag{2}$$

where $\sigma > 0$ is the kernel bandwidth, and $\|.\|$ denotes the Euclidean norm. As one can see from (1), the KLMS creates a growing LIP nonlinear model where the nonlinear transformation is determined by the selected Mercer kernel. An appealing feature of the KLMS is that the linear combination coefficients are directly related to the prediction errors.

B. *Equality function constraints*

The KLMS is a "*black box*" method with which the learned model is completely determined by the training data (assuming that the step-size η and kernel function κ are given). In this paper, some additional constraints will be incorporated into the KLMS. Specifically, consider the following equality function constraints (EFC) on the learned mapping f_i

$$f_i(u) = f_C(u) \ \ , \forall u \in U_C \subset U \tag{3}$$

where $f_C(u)$ is a prior known function defined on U_C, with U_C being the constraint domain, a subset of the input domain U. The definition domain of the function $f_C(u)$ can be extended to the whole input domain U by defining

$$f_C(u) = f_C(u_C) \quad \text{s.t.}\, u_C = \underset{c \in U_C}{\arg\min} \|u-c\| \tag{4}$$

Especially, $u_C = u$ when $u \in U_C$.

Now our goal is to modify the original KLMS such that the learned model strictly satisfies the above constraints while achieving improved learning performance. A novel approach will be proposed in the next section to address this issue. Note that in [20], the equality function constraints were successfully applied in an RBF model, which is implemented in a batch mode (not an online manner).

III. CONSTRAINED MERCER KERNEL AND KLMS-EFC

A. *Constrained Mercer kernel*

Definition 1: The constrained Mercer kernel with respect to the constraint domain U_C is defined by

$$\kappa_C(u,u') = \rho_C(u)\rho_C(u')\kappa(u,u') \quad , \quad u,u' \in U \tag{5}$$

where $\kappa(u,u')$ is the original Mercer kernel, and $\rho_C(u)$ is a weighting function with respect to the constraint domain U_C, given by

$$\rho_C(u) = 1 - \exp(-\beta\Delta(u)) \tag{6}$$

where $\Delta(u) = \min_{c \in U_C} \|u-c\|$ is the minimal distance from to the constraint domain U_C ,and $\beta > 0$ is a parameter for adjusting the slope of $\rho_C(u)$.

Remark 1: $\rho_C(u)$ is a continuous function of u, which takes the value of zero when $u \in U_C$, while approaching 1.0 when u is apart from the constraint domain U_C. By definition, the constrained Mercer kernel $\kappa_C(u,u')$ will gradually lose its learning capability when u gets close to U_C.

Now we prove that $\kappa_C(u,u')$ is really a Mercer kernel function over $U \times U$. Obviously, $\kappa_C(u,u')$ is a continuous and symmetric function over $U \times U$. So we only need to prove the positive-definiteness of $\kappa_C(u,u')$. For any $n \in N$, and any choice of $u_1, u_2, \text{L}\ , u_N \in U$ and $a_1, a_2, \text{L}\ , a_N \in R$, we have

$$\begin{aligned}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j \kappa_C(u_i,u_j) &= \sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j \rho_C(u_i)\rho_C(u_j)\kappa(u_i,u_j) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} b_i b_j \kappa(u_i,u_j) \\ &\overset{(a)}{\geq} 0\end{aligned} \tag{7}$$

where $b_i = a_i\rho_C(u_i)$, $b_j = a_j\rho_C(u_j)$, and (7) follows from the fact that κ is a Mercer kernel. Thus $\kappa_C(u,u')$ is positive-definite and hence, is also a Mercer kernel.

Remark 2: Though $\kappa_C(u,u')$ is positive-definite, it is not *strictly positive definite* (SPD) since we have $\kappa_C(u,u') = 0$ for any $u \in U_C$ or $u' \in U_C$.

B. *KLMS-EFC*

To satisfy the equality function constraints, the initial estimate of f is set at

$$f_0(u) = \left(1 - \rho_C(u)\right) f_C(u) \tag{8}$$

which equals $f_C(u)$ when $u \in U_C$, and approaches zero when u is apart from U_C. Here, $f_C(u)$ is given by (4) to cover the whole input domain. With the above initialization, the proposed KLMS-EFC algorithm becomes the KLMS with the constrained Mercer kernel $\kappa_C(u,u')$, that is

$$\begin{cases} f_0 = \left(1-\rho_C(.)\right) f_C(.) \\ e_i = d_i - f_{i-1}(u_i) \\ f_i = f_{i-1} + \eta e_i \kappa_c\left(u_i, \cdot\right) \end{cases} \tag{9}$$

Remark 3: If $f_C(u)$ is a continuous function, the learned mapping f_i is also a continuous function since both $\rho_C(u)$ and $\kappa_C(u,u')$ are continuous functions.

Remark 4: The computational complexity of the KLMS-EFC is almost the same as that of the original KLMS. In addition, there is only one extra free parameter in KLMS-EFC, namely the parameter β, which controls the learning rate and smoothness around the boundary of U_C.

Substituting (5) into the update rule of KLMS-EFC, we obtain

$$f_i = f_{i-1} + \eta e_i \rho_C(u_i)\rho_C(.)\kappa(u_i,\cdot) \tag{10}$$

From (10) one can observe: 1) if $u_i \in U_C$, there is no update on f_{i-1} (also no dictionary update on the hidden nodes) ; 2) if u_i is very close to U_C, the update rate is very small because $\rho_C(u_i) \to 0$ as $\Delta(u_i) \to 0$.

Clearly, the learned mapping at iteration i is

$$f_i(u) = \left(1-\rho_C(u)\right) f_C(u) + \sum_{j=1}^{i} \eta e_j \rho_C(u_j)\rho_C(u)\kappa(u_j,u) \tag{11}$$

which basically consists of two parts, namely the prior known part $\left(1-\rho_C(u)\right) f_C(u)$ and the sequentially learned part $\sum_{j=1}^{i} \eta e_j \rho_C(u_j)\rho_C(u)\kappa(u_j,u)$. The second part still has a growing *linear-in-the-parameters* (LIP) structure like usual KAFs although there is no dictionary update when $u_i \in U_C$. To reduce the computational costs and memory requirements, one can use some sparsification techniques [1] or quantization methods [12-14] to curb the network growth and obtain a compact model.

IV. SIMULATION RESULTS

We present simulation results to demonstrate the performance of the proposed KLMS-EFC. Consider the following "hyperboloid" function (as shown in Fig.1) [20]

$$y = u_1 \cdot u_2, \quad u_1,\ u_2 \in [-1,1] \tag{12}$$

and the equality function constraints:

$$f_C(u) = u_1^2,\ u \in U_C \tag{13}$$

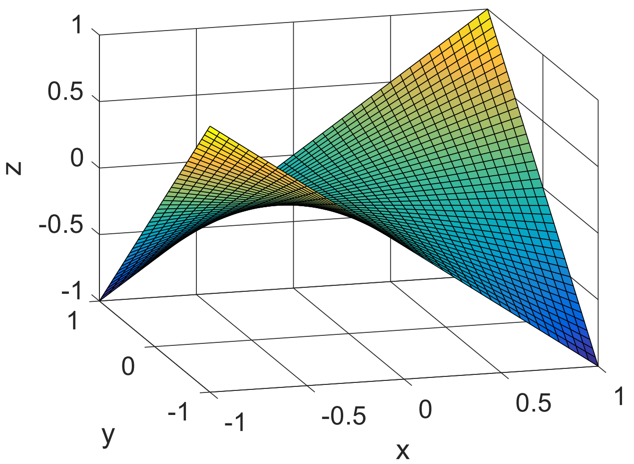

Fig.1 "hyperboloid" function

where $U_C = \{u \mid u_1 = u_2, u_1, u_2 \in [-1,1]\}$. The goal is to fit the function based on the data and the equality constraints. We draw 1000 training samples in which 980 samples are drawn from the uniform distribution over $u_1, u_2 \in [-1,1]$, and 20 samples equally spaced in U_C. In addition, the testing data contain 880 samples in which 800 samples are drawn from the uniform distribution over $u_1, u_2 \in [-1,1]$, and 80 samples equally spaced in U_C. In the simulation, the training data are corrupted by additive Gaussian noise with zero mean and 0.05 standard deviation.

Fig.2 shows the performance comparison between KLMS-EFC ($\eta = 1.2$, $\beta = 1.2$) and the original KLMS with different step sizes (0.2, 0.7, 1.2, 1.6). In both KLMS and KLMS-EFC, the kernel function $\kappa(.,.)$ is chosen as the Gaussian kernel with bandwidth $\sigma = 0.4$. At each iteration, the testing mean square error (testing MSE) is computed on the testing set using the filter resulting from the training set. The plotted results are obtained by averaging over 50 Monte Carlo runs. As one can see, the equality constraints can improve the learning performance, and the KLMS-EFC can outperform the KLMS with different step sizes. The testing outputs and desired responses in constraint domain are shown in Fig. 3. It is evident that the model trained by KLMS-EFC fits the data much better than the model trained by KLMS.

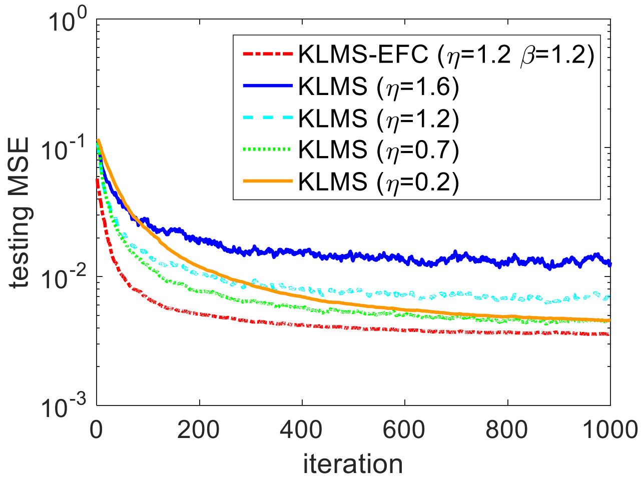

Fig.2 Convergence curves in terms of the testing MSE

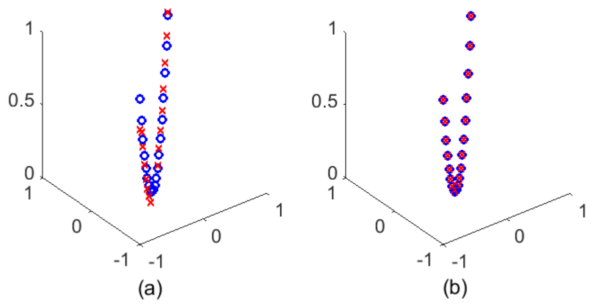

Fig.3 Testing outputs (cross) and desired responses (circle) in constraint domain: (a) KLMS; (b) KLMS-EFC

Fig.4 shows the performance of the KLMS-EFC with different values of β. As we can see, the leaning process is stopped when $\beta = 0$. In this case we have $\rho_C(u) \equiv 0$ and $f_i \equiv f_{i-1}$. But the algorithm can work very well even with a very small β. When β is too large (say $\beta \ge 12$), the performance will deteriorate. In this example, the best performance is achieved at round $\beta = 1.2$. In the simulation, the step sizes are manually chosen such that all the initial convergence speeds (except the case $\beta = 0$) are visually similar. β is an important parameter, but it is relatively easy to choose, as in general a small value of β will bring satisfactory results.

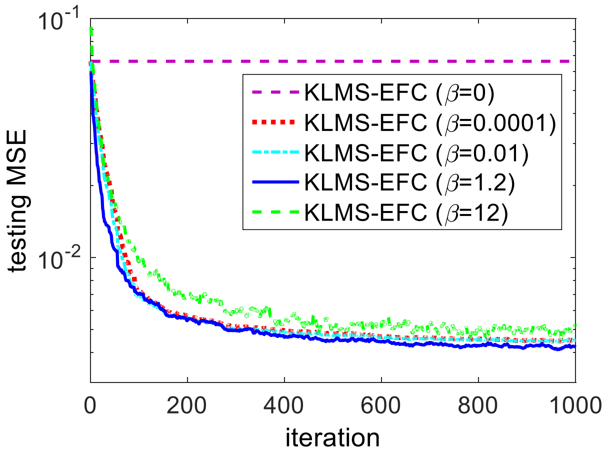

Fig.4 Convergence curves with different values of β

To curb the network growth, one can use a quantization approach to develop the quantized KLMS-EFC (QKLMS-EFC) algorithm (see [12] for the details about QKLMS). Fig.5 illustrates the convergence performance of the QKLMS-EFC with different quantization sizes (ε), and Fig. 6 shows the corresponding testing MSEs versus the dictionary sizes. Similar to the QKLMS, there is a trade-off between accuracy and dictionary size for the QKLMS-EFC. Usually, a larger quantization size leads to a poorer accuracy but a smaller dictionary size. With a proper quantization size, however, the algorithm can produce a small network while achieving desirable performance.

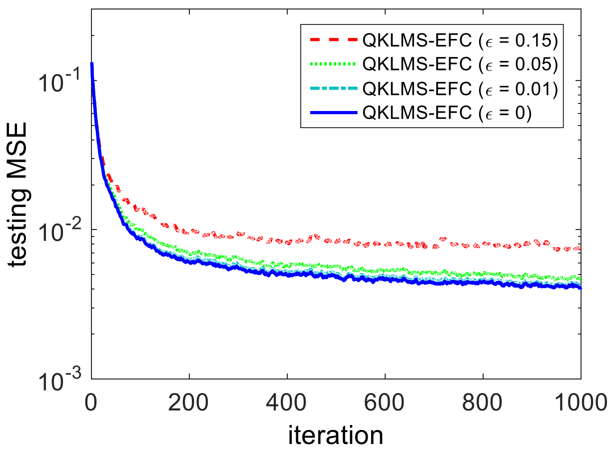

Fig.5 Convergence curves with different quantization sizes

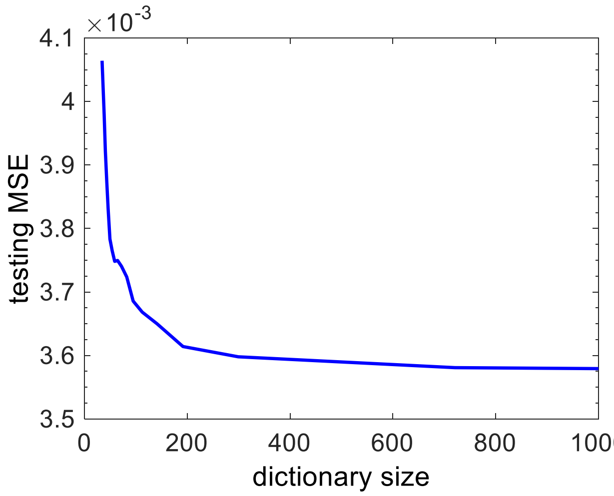

Fig.6 Testing MSEs versus dictionary sizes

V. CONCLUSION

Kernel adaptive filters (KAFs) are powerful online learning machines. But they are "*black box*" models and their performance can be significantly improved if some prior knowledge is incorporated into the learned models. In this study, we developed an efficient kernel adaptive filtering algorithm by incorporating the *equality function constraints* into the kernel least mean square (KLMS) algorithm. A novel Mercer kernel, called the *constrained Mercer kernel* (CMK), was proposed. The *kernel least mean square subject to equality function constraints* (KLMS-EFC) was then developed with this new kernel, which can satisfy the constraints perfectly while achieving significant performance improvement. Simulation results confirmed the excellent performance of the new algorithm.

ACKNOWLEDGMENTS

This work was supported by 973 Program (No. 2015CB351703) and National NSF of China (No. 61372152).

REFERENCES

[1] W. Liu, J. Principe, S. Haykin, *Kernel Adaptive Filtering: A Comprehensive Introduction*, Wiley, 2010.

[2] J. C. Principe, B. Chen, Universal Approximation with Convex Optimization: Gimmick or Reality? *IEEE Computational Intelligence Magazine*. vol. 10, no. 2, pp. 68-77, 2015.

[3] J. Kivinen, A. J. Smola, and R. C. Williamson, Online learning with kernels, *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2165–2176, Aug. 2004.

[4] W. Liu, P. Pokharel, J. Principe, The kernel least mean square algorithm, *IEEE Transactions on Signal Processing*, vol. 56, pp. 543-554, 2008.

[5] W. Liu, J. Principe, Kernel affine projection algorithm, *EURASIP J. Adv. Signal Process.*, vol. 2008, Article ID 784292, 12 pages, doi: 10.1155/2008/784292.

[6] Y. Engel, S. Mannor, R. Meir, The kernel recursive least-squares algorithm, *IEEE Transactions on Signal Processing*, vol. 52, pp. 2275-2285, 2004.

[7] W. Liu, Il Park, Y. Wang, J. C. Principe, Extended kernel recursive least squares algorithm, *IEEE Transactions on Signal Processing*, vol. 57, pp. 3801-3814, 2009.

[8] C. Richard, J. Bermudez, P. Honeine, Online prediction of time series data with kernels, *IEEE Transactions on Signal Processing*, vol. *57*, pp.1058-1067, 2009.

[9] M. Yukawa, Multikernel adaptive filtering, *IEEE Transactions on Signal Processing*, vol. *60*, pp. 4672-4682, 2012.

[10] K. Slavakis, S. Theodoridis, and I. Yamada, Online kernel-based classification using adaptive projection algorithms, *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2781–2796, Jul. 2008.

[11] F. Orabona, J. Keshet, B. Caputo, Bo*unded kernel-based online le*arning, *Journal of Machine Learning Research*, vol. 10, pp. 2643-2666, 2009.

[12] B. Chen, S. Zhao, P. Zhu, J. C. Principe, Quantized kernel least mean square algorithm, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 22-32, 2012.

[13] B. Chen, S. Zhao, P. Zhu, J. C. Principe, Quantized kernel recursive least squares algorithm, *IEEE Trans. On Neural Networks and Learning Systems*, vol. 24, no. 9, 1484-1491, 2013.

[14] Xu X., Qu H., Zhao J., Yang X., Chen B., Quantized kernel least mean square with desired signal smoothing. *Electronics Letters*, vol. 51, no.18, pp. 1457-1459, 2015.

[15] B. Chen, S. Zhao, P. Zhu, J. Principe, Mean square convergence analysis for kernel least mean square algorithm, *Signal Processing*, vol. 92, pp. 2624-2632, 2012.

[16] S. Zhao, B. Chen, J. C. Principe, Kernel adaptive filtering with maximum correntropy criterion, in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, San Jose, California, USA, pp. 2012-2017, 2011.

[17] B.Chen, Z. Yuan, N. Zheng, J. C. Principe, Kernel minimum error entropy algorithm. *Neurocomputing,* vol. 121, 160-169, 2013.

[18] C. Saide, R. Lengelle, P. Honeine, C. Richard, R. Achkar, Nonlinear adaptive filtering using kernel based algorithms with dictionary adaptation. *International Journal of Adaptive Control and Signal Processing*, 2015. Published online in Wiley Online Library (wileyonlinelibrary.com). doi: 10.1002/acs.2548.

[19] T. K. Paul, T. Ogunfunmi, A kernel adaptive algorithm for quaternion-valued inputs, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, 2422-2439, 2015.

[20] L. Cao and B. Hu, "Generalized constraint neural network regression model subject to equality function constraints." *Proceedings of International Joint Conference on Neural Networks (IJCNN)* , pp. 1-8, 2015.

[21] O. L. Mangasarian and E. W. Wild, "Nonlinear knowledge in kernel approximation," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 300-306, 2007.

[22] Y. Qu and B.G. Hu, "Generalized constraint neural network regression model subject to linear priors," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2447-2459, 2011

[23] K. S. McFall and J. R. Mahan, "Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions," *IEEE Transactions on Neural Networks*, vol. 20, no. 8, pp. 1221-1233, 2009.

[24] S. Chen, X. Hong, and C. J. Harris, "Grey-box radial basis function modelling," *Neurocomputing*, vol. 74, no. 10, pp. 1564-1571, 2011.

[25] X. Hong and S. Chen, "A new RBF neural network with boundary value constraints," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 1, pp. 298-303, 2009.