九州大学学術情報リポジトリ
Kyushu University Institutional Repository

Radial Basis Function Network in Reproducing Kernel Hilbert Space

Dachapak, Chooleewan
Department of Electrical and Electronic Systems Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University : Graduate Student

Kanae, Shunshoku
Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineeringg, Kyushu University

Yang, Zi-Jiang
Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Wada, Kiyoshi
Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

https://doi.org/10.15017/1516054

出版情報：九州大学大学院システム情報科学紀要. 10 (1), pp.9-14, 2005-03-25. 九州大学大学院システム情報科学研究院
バージョン：
権利関係：

九州大学大学院
システム情報科学紀要
第10巻 第1号 平成17年3月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.10, No.1, March 2005

Radial Basis Function Network in Reproducing Kernel Hilbert Space

Chooleewan DACHAPAK* , Shunshoku KANAE**,
Zi-Jiang YANG** and Kiyoshi WADA**

(Received December 10, 2004)

Abstract: The present study employs an idea of mapping data into a high dimensional feature space which is known as Reproducing Kernel Hilbert Space (RKHS), then performs Radial Basis Function (RBF) network in the feature space where the new basis function will be obtained and finally, Orthogonal Least Squares (OLS) method is employed to select a suitable set of centers (regressors) from a large set of candidates in order to obtain a sparse regression model in the feature space. The proposed method is employed to the simple scalar function estimation problems and nonlinear system identification problem by simulations.

Keywords: Reproducing kernel Hilbert space, Orthogonal least squares algorithm, Radial basis function, Neural networks

1. Introduction

A kernel-based algorithm's idea of implicitly nonlinear mapping the data into a high-dimensional feature space RKHS has been a very fruitful one in the context of support vector machine (SVM)[1]. The basic insight gained by Vapnik was that problems that are difficult to solve in low dimensions may become much easier if the data is mapped to a high-dimensional space. The kernel-based algorithm is a nonlinear version of a linear algorithm where the data has been previously (nonlinearly) transformed to a higher dimensional space in which we only need to be able to compute inner products . However, the direct computation in the high-dimensional feature space is very time-consuming or impossible. Therefore, Mercer kernels are employed to make the calculation in feature space practical. Such technique has been adopted in many studies other than SVM such as Kernel Principal Component Analysis (KPCA)[2] showing a high performance nonlinear form of PCA.The attractiveness of the kernel-based algorithm stems from their elegant treatment of nonlinear problems and their efficiency in high-dimensional problems, where they allow to work in a simple (linear) way. Transforming the data nonlinearly to a higher dimensional space ensures that a linear algorithm can be used over it to obtain a linear explantion of the data.

Radial Basis Function neural network has been successfully applied for nonlinear function approximation and data classification in wide range areas. A standard RBF network has a feedforward structure consisting of two layers, a nonlinear hidden layer and a linear output layer. The nodes or basis functions in the hidden layer operate on the distance from an applied input data vector to an internal parameter vector called a center. In practices the centers are often chosen to be a subset of input data. The output layer implements a linear combiner and only adjustable parameters are the weights of this linear combiner of the basis function responses. These parameters can be determined by using the linear least squares (LS) method.

Orthogonal decomposition is well known to be a numerically robust method for solving least squares problem and can be applied to obtain the weighted parameter of RBF network. OLS was extended to select suitable RBF network centers from a large number of candidates by evaluating an error reduction ratio in a forward selection algorithm. The network is then built up by adding center, which has the largest error reduction ratio in each step until the adequate network has been constructed. Each selected center maximizes the increment to the explained variance or energy of the desired output and does not suffer numerical ill-conditioning problem that occurs frequently in random selection of centers.

This study proposes a combining between the idea of mapping the data into the feature space and RBF network. Firstly map the data into the high-dimensional feature space then apply the RBF network in feature space. By use of Mercer kernel,

* Department of Electrical and Electronic Systems Engineering, Graduate Student

** Department of Electrical and Electronic Systems Engineering

it can compute the distance between input vector and centers, which are mapped into feature space, without mapping both of them explicitly. The new basis function will be derived and illustrated in the section 3. Finally, perform OLS subset regression procedure to select the centers in feature space and obtain the parsimonious regression model in the feature space.

2. Radial Basis Function Network

Assume that $f_r(\boldsymbol{x})$ is centered then a basic architecture of RBF network with N inputs and a scalar output is illustrated in **Fig.1**.

$$f_r(\boldsymbol{x}) = \sum_{i=1}^{n_r} \lambda_i \phi(\|\boldsymbol{x} - \boldsymbol{c}_i\|). \tag{1}$$

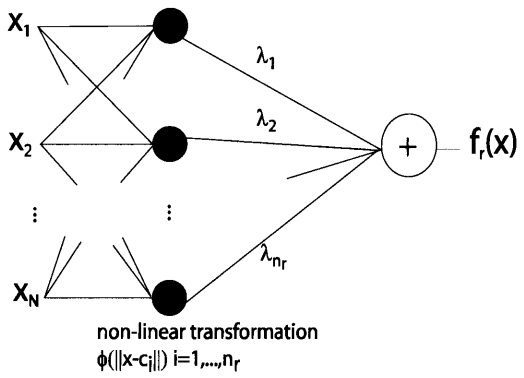

Fig.1 Radial basis function network.

where $\boldsymbol{x} \in R^N$ is an input vector, $\phi(\cdot)$ is a given function, $\|\cdot\|$ denotes the Euclidean norm, λ_i, $0 \leq i \leq n_r$, are the weights or parameters, $\boldsymbol{c}_i \in \boldsymbol{R}^N$, $1 \leq i \leq n_r$, are known as the RBF centers, and n_r is the number of centers. In the RBF network, the function form $\phi(\cdot)$ and the centers $\boldsymbol{c}_i$ are assumed to have been fixed. The values of weights λ_i can be determined by using the linear least squares method. There are many kinds of $\phi(\cdot)$ function available such as thin-plate-spline function,

$$\phi(\nu) = \nu^2 \log(\nu), \tag{2}$$

Gaussian function,

$$\phi(\nu) = \exp(-\nu^2/\beta^2). \tag{3}$$

All data samples $\boldsymbol{x}_t$, $t = 1, \cdots, M$ will be used as centers $\boldsymbol{c}_i$ to initialize a model set. This means that n_r equals to M.

3. RBF network in RKHS

This study, firstly map the data nonlinearly into the feature space F by the nonlinear function as following,

$$\Phi^* : \boldsymbol{R}^N \to F, \quad \boldsymbol{x} \mapsto \phi^*(\boldsymbol{x}). \tag{4}$$

Note that the feature space F could have an arbitrarily large, possibly infinite, dimensionality. Feature space can be regarded as Reproducing Kernel Hilbert Space[1]. Then perform the radial basis function network in feature space as following:

$$f_r(\phi^*(\boldsymbol{x})) = \sum_{i=1}^{n_r} \lambda_i \phi(\|\phi^*(\boldsymbol{x}) - \phi^*(\boldsymbol{c}_i)\|_{\mathcal{H}}). \tag{5}$$

$\|\cdot\|_{\mathcal{H}}$ denotes norm in Hilbert space. From (5), consider term of $\|\phi^*(\boldsymbol{x}) - \phi^*(\boldsymbol{c}_i)\|_{\mathcal{H}}$ as following,

$$\begin{aligned} &\|\phi^*(\boldsymbol{x}) - \phi^*(\boldsymbol{c}_i)\|_{\mathcal{H}} \\ &= (\langle \phi^*(\boldsymbol{x}) - \phi^*(\boldsymbol{c}_i), \phi^*(\boldsymbol{x}) - \phi^*(\boldsymbol{c}_i) \rangle)^{\frac{1}{2}} \\ &= (\phi^*(\boldsymbol{x}) \cdot \phi^*(\boldsymbol{x}) \\ &\quad -2\phi^*(\boldsymbol{x}) \cdot \phi^*(\boldsymbol{c}_i) + \phi^*(\boldsymbol{c}_i) \cdot \phi^*(\boldsymbol{c}_i))^{\frac{1}{2}}. \end{aligned} \tag{6}$$

Compared to (1), it is clear that the term of $\|\boldsymbol{x} - \boldsymbol{c}_i\|$ is directly calculated with N dimensional data $\boldsymbol{x}$ and $\boldsymbol{c}_i$. However, the feature space has large or infinite dimensionality. Moreover, $\phi^*(\boldsymbol{x})$ and $\phi^*(\boldsymbol{c}_i)$ cannot be explicitly mapped. These make the direct calculation of dot product in feature space impossible.

By use of Mercer kernel[1], it makes such calculation of dot product in feature space possible without mapping the data explicitly. This way can avoid dealing with the mapped data explicitly, which may be perhaps impossible or intractable in terms of memory and computational cost.

Let $\boldsymbol{K}$ be Kernel matrix, which is known as Gram matrix,

$$K = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_M) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_M) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_M, x_1) & k(x_M, x_2) & \cdots & k(x_M, x_M) \end{bmatrix}. \tag{7}$$

Kernel matrix $\boldsymbol{K}$ is composed of the kernel function of data $\boldsymbol{x}_t$, $1 \leq t \leq M$. It is clearly seen that $\boldsymbol{K}$ is symmetric and semi-positive definite matrix. These mean its all eigenvalues are real and non-negative. Here, a function that generates symmetric positive definite Gram matrix for any finite sample data, is a valid kernel function[1].

This study employs Gaussian kernel function

$k(\boldsymbol{x},\boldsymbol{y}) = \exp(\frac{-\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2})$. With Gaussian kernel function, the component $K(i,j)$, where $i=j$, has value equal to one.

Then any occurrence of $\phi^*(\boldsymbol{x})\cdot\phi^*(\boldsymbol{y})$ is replaced by kernel function $k(\boldsymbol{x},\boldsymbol{y})$. Hence, (6) can be rewritten in the term of kernel function as,

$$\|\phi^*(\boldsymbol{x})-\phi^*(\boldsymbol{c}_i)\|_{\mathcal{H}} = (k(\boldsymbol{x},\boldsymbol{x})-2k(\boldsymbol{x},\boldsymbol{c}_i)+k(\boldsymbol{c}_i,\boldsymbol{c}_i))^{\frac{1}{2}}. \tag{8}$$

From above, it is clear that kernel function makes the calculation of dot product in feature space practical without dealing with mapping explicitly the data into the feature space. In practice, firstly define a kernel function $k(\boldsymbol{x},\boldsymbol{y})$ directly, then the feature space will be defined implicitly corresponding to kernel function[4]. Then, use RBF network (5) to construct a linear regression model as,

$$d(t)=\sum_{i=1}^{n_r} p_i(t)\theta_i+\epsilon(t), \tag{9}$$

where $d(t)$ is the desired output for $t=1$ to M, the θ_i are parameters, and the $p_i(t)$ are known as the regressors which are some fixed functions of $\phi^*(\boldsymbol{x}(t))$:

$$p_i(t)=p_i(\phi^*(\boldsymbol{x}(t))). \tag{10}$$

Here $p_i(\phi^*(\boldsymbol{x}(t)))$ is regarded as $\phi(\|\phi^*(\boldsymbol{x}(t))-\phi^*(\boldsymbol{c}_i)\|_{\mathcal{H}})$ and the error signal $\epsilon(t)$ is assumed to be uncorrelated with the regressors $p_i(t)$.
From 9 and 10,

$$d(t)=\sum_{i=1}^{n_r}\theta_i\phi(\|\phi^*(\boldsymbol{x}(t))-\phi^*(\boldsymbol{c}_i)\|_{\mathcal{H}})+\epsilon(t), \tag{11}$$

where the proposed basis function $p_i(t)$, can be derived as following,

$$\begin{aligned}
p_i(t) &= \exp(-\frac{\|\phi^*(x)-\phi^*(c_i)\|^2}{\beta^2})\\
&= \exp\Big(-\frac{k(x,x)-2k(x,c_i)+k(c_i,c_i)}{\beta^2}\Big)\\
&= \exp\Big(-\frac{k(x,x)}{\beta^2}\Big)\exp\Big(-\frac{k(c_i,c_i)}{\beta^2}\Big)\exp\Big(\frac{2k(x,c_i)}{\beta^2}\Big)\\
&= \exp\Big(-\frac{1}{\beta^2}\Big)\exp\Big(-\frac{1}{\beta^2}\Big)\exp\Big(\frac{2k(x,c_i)}{\beta^2}\Big)\\
&= \exp\Big(-\frac{2}{\beta^2}\Big)\exp\Big(\frac{2k(x,c_i)}{\beta^2}\Big)\\
&= \exp\Big(-\frac{2(1-k(x,c_i))}{\beta^2}\Big)\\
&= \Big\{\exp\Big(-\frac{1-k(x,c_i)}{\beta^2}\Big)\Big\}^2\\
&= \Big\{\exp\Big(-\frac{1-\exp(-\frac{\|x-c_i\|^2}{2\sigma^2})}{\beta^2}\Big)\Big\}^2.
\end{aligned}$$

Figure 2 and **Fig.3** plot the proposed basis function when the center is fixed to value of -0.0690.

4. Orthogonal Least Squares Learning

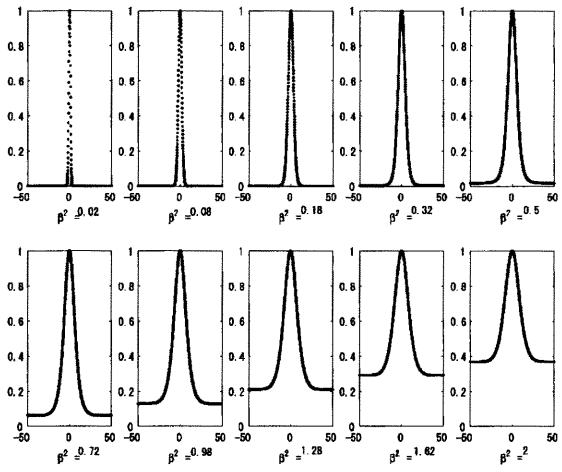

Fig.2 Basis function for various value of β^2 with fixed σ ($\sigma=0.1$).

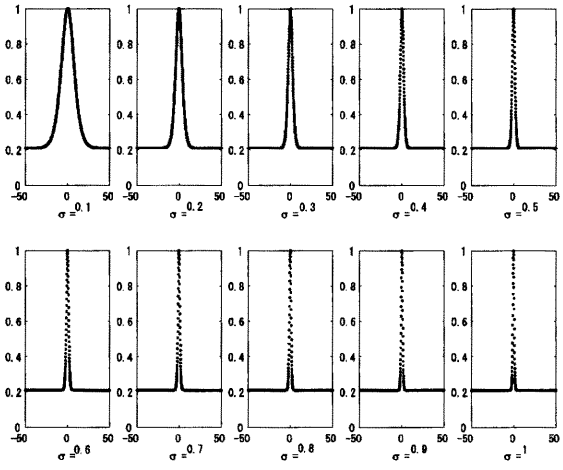

Fig.3 Basis function for various value of σ with fixed β^2 ($\beta^2=1.28$).

Algorithm

The problem of how to select a suitable set of RBF centers from the data set can be regarded as an example of how to select a subset of significant regressors from a given candidate set. An efficient learning procedure for selecting a subset model from (9) can be derived based on the OLS method. Rewrite equation (9) into the matrix form as

$$\boldsymbol{d}=\boldsymbol{P}\boldsymbol{\Theta}+\boldsymbol{E}, \tag{12}$$

where

$$\begin{aligned}
&\boldsymbol{d}=[d(1)\cdots d(M)]^T,\\
&\boldsymbol{P}=[\boldsymbol{p}_1\cdots\boldsymbol{p}_{n_r}],\\
&\boldsymbol{p}_i=[p_i(1)\cdots p_i(M)]^T, \quad 1\le i\le n_r\\
&\boldsymbol{\Theta}=[\theta_1\cdots\theta_{n_r}]^T,\\
&\boldsymbol{E}=[\epsilon(1)\cdots\epsilon(M)]^T.
\end{aligned} \tag{13}$$

Note that number of centers n_r equals to M since all M data samples are employed as centers to initialize the model. Vectors $\boldsymbol{p}_i$ form a set of basis vectors, and the linear squares solution $\hat{\boldsymbol{\Theta}}$ satisfies the condition that the square of the projection $\boldsymbol{P}\hat{\boldsymbol{\Theta}}$ is part of the desired output energy that can be counted by the regressors. Because different regressors are generally correlated, it is not clear how an individual regressor contributes to this output energy. The OLS method involves the transformation of the set of $\boldsymbol{p}_i$ into a set of orthogonal basis vectors, and thus makes it possible to calculate the individual contribution to the desired output energy from each basis vector. The regression matrix $\boldsymbol{P}$ can be decomposed into

$$\boldsymbol{P} = \boldsymbol{W}\boldsymbol{A} \tag{14}$$

where $\boldsymbol{A}$ is an $n_r \times n_r$ triangular matrix with 1's on the diagonal and 0's below the diagonal, that is

$$\boldsymbol{A} = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & \alpha_{1n_r} \\ 0 & 1 & \alpha_{23} & \cdots & \alpha_{2n_r} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & \alpha_{n_r-1n_r} \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

and $\boldsymbol{W}$ is an $M \times n_r$ matrix with orthogonal columns $\boldsymbol{w}_i$ such that

$$\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{H} \tag{16}$$

where $\boldsymbol{H}$ is diagonal matrix with elements h_i:

$$h_i = \boldsymbol{w}_i^T\boldsymbol{w}_i = \sum_{t=1}^{M} w_i(t)w_i(t), \quad 1 \le i \le n_r. \tag{17}$$

And (12) can be rewritten as

$$\boldsymbol{d} = \boldsymbol{W}\boldsymbol{g} + \boldsymbol{E}. \tag{18}$$

The OLS solution $\hat{\boldsymbol{g}}$ is given by

$$\hat{\boldsymbol{g}} = \boldsymbol{H}^{-1}\boldsymbol{W}^T\boldsymbol{d} \tag{19}$$

or

$$\hat{g}_i = \boldsymbol{w}_i^T\boldsymbol{d}/(\boldsymbol{w}_i^T\boldsymbol{w}_i), \quad 1 \le i \le n_r. \tag{20}$$

The quantities $\hat{\boldsymbol{g}}$ and $\hat{\boldsymbol{\Theta}}$ satisfy the triangular system

$$\boldsymbol{A}\hat{\boldsymbol{\Theta}} = \hat{\boldsymbol{g}}. \tag{21}$$

The OLS method is to use for subset selection of the candidate RBF centers. In practice, the number of data is often very large and centers are to be chosen as a subset of data set. All the candidate regressors $n_r = M$ can be very large and a suitable modeling may only require $M_s (<< n_r = M)$ significant regressors. Because $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ are orthogonal for $i \neq j$ and $\boldsymbol{E}$ is supposed to be uncorrelated with regressors, the sum of square or energy of $d(t)$ is

$$\boldsymbol{d}^T\boldsymbol{d} = \sum_{i=1}^{n_r} g_i^2\boldsymbol{w}_i^T\boldsymbol{w}_i + \boldsymbol{E}^T\boldsymbol{E}. \tag{22}$$

If $\boldsymbol{d}$ is the desired output vector after it has been centered, the variance of $d(t)$ is given by

$$M^{-1}\boldsymbol{d}^T\boldsymbol{d} = M^{-1}\sum_{i=1}^{n_r} g_i^2\boldsymbol{w}_i^T\boldsymbol{w}_i + M^{-1}\boldsymbol{E}^T\boldsymbol{E}. \tag{23}$$

It is seen that $M^{-1}\sum_{i=1}^{n_r} g_i^2\boldsymbol{w}_i^T\boldsymbol{w}_i$ is the increment to explained desired output variance introduced by $\boldsymbol{w}_i$, and error reduction ratio due to $\boldsymbol{w}_i$ can be defined as

$$[err]_i = g_i^2\boldsymbol{w}_i^T\boldsymbol{w}_i/(\boldsymbol{d}^T\boldsymbol{d}), \quad 1 \le i \le n_r. \tag{24}$$

The ratio offers simple and effective means of seeking a subset of significant regressors in a forward-regression manner[3]. By adding one more regressor, it increases the explained variance of the dependent variable. And the iteration procedure is terminated at $M_s{}^{th}$ step when $1 - \sum_{i=1}^{M_s}[err]_i$ reaches a chosen tolerance ρ, where $0 < \rho < 1$. This gives a subset model containing M_s significant regressors. For more details of numerical iteration see[3].

5. Simulation Results

Two examples were employed in the simulations to show the performance of the proposed method. Gaussian RBF and Gaussian kernel function were utilized in all simulations.

The first example is a modeling of the scalar function,

$$f(x) = \sin(2\pi x), \quad 0 \le x \le 1.$$

Two hundred data were generated from $y = f(x)+\epsilon$, where x was taken from the uniform distribution in $(0, 1)$ and the noise ϵ had a normal distribution with zero mean and variance 0.1. The first one hundred data were employed as a training data set, the last one hundred data were employed for a possible cross-validation procedure. One hundred noise-free data $f(x)$ were also generated as the testing data set for model evaluation. The noisy training points y and the underlying function $f(x)$ are plotted in **Fig.4**. As each training data x was considered as a candidate RBF center, there were $n_r = M = 100$ regressors in the initial regression model. The iteration procedure stopped at 5^{th} step. This produces 5 terms model from 100 regressors model at the ini-

tial. A simulated function is illustrated in **Fig.4**. **Figure 5** shows the proposed basis function of each first five chosen centers.

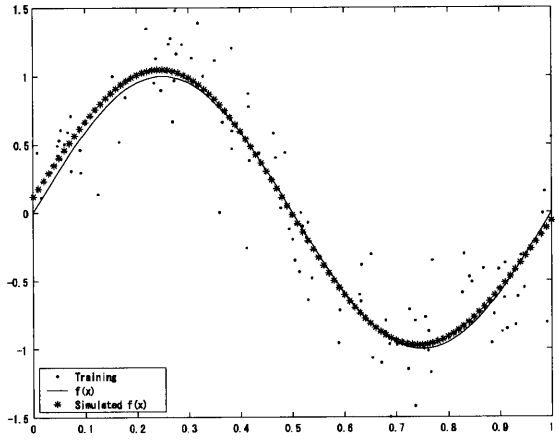

Fig.4 Noisy training data set of $f(x)$, the underlying function $f(x)$ and simulated $f(x)$.

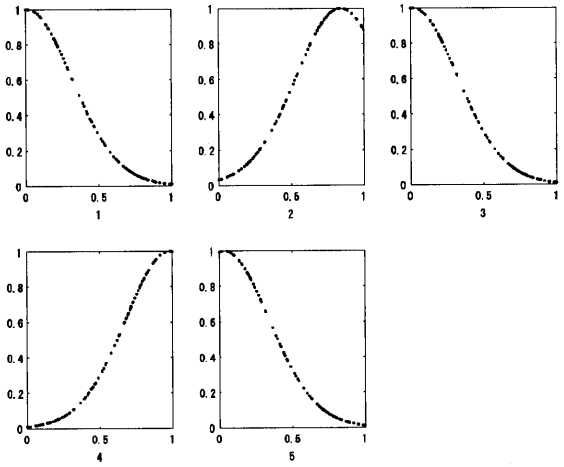

Fig.5 Basis function plotted for the first five chosen centers, 1. the first chosen center 0.0077, 2. the second chosen center 0.8415, 3. the third chosen center 0.0159, 4. the fourth chosen center 0.9826 and 5. the fifth chosen center 0.0346.

The second example is to apply the proposed method to nonlinear system identification problem[5] in order to show the performance of the proposed method. The model is assumed to be of the form

$$y_p(k+1) = f[y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)] + e(k)$$

where the unknown function f has the form

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

One thousand input samples are uniformly distributed in the interval $[-1, 1]$ as random input $u(k)$ for training procedure and illustrated in **Fig.6**. And Gaussian noises are generated with variance 0.01 and zero mean. In order to understand clearly **Fig.7** shows only the first one hundred data of the observed output and predicted output obtained from training procedure. **Figure 8** shows one thousand input signal for testing procedure given by

$$u(k) = \sin(2\pi k/250) \quad \text{for } k \leq 500$$
$$u(k) = 0.8\sin(2\pi k/250) + 0.2\sin(2\pi k/25) \quad \text{for } k > 500.$$

Other than the error reduction ratio criteria, we can use AIC to decide the iteration procedure to terminate. AIC criteria is given by

$$\mathrm{AIC}(i) = M \ln \hat{\sigma}_e^2(i) + 2 \times \mathrm{i},$$

where M is the number of training data samples, i is the iteration order and also the number of subset regressors at i^{th} step of iteration. And $\hat{\sigma}_e^2(i)$ can be obtained in each i^{th} iteration as

$$\hat{\sigma}_e^2(i) = \frac{1}{M} \sum_{t=1}^{N} \hat{e}_t^2.$$

There were $n_r = M = 1000$ candidates in the initial model set for this example. The iteration terminated at the 35^{th} step when it detected that the the 35^{th} step gave the least AIC. This generates 35 terms sparse model from 1000 regressors model from the initial model. **Figure 9** shows the observed output and predicted output obtained from the proposed method.

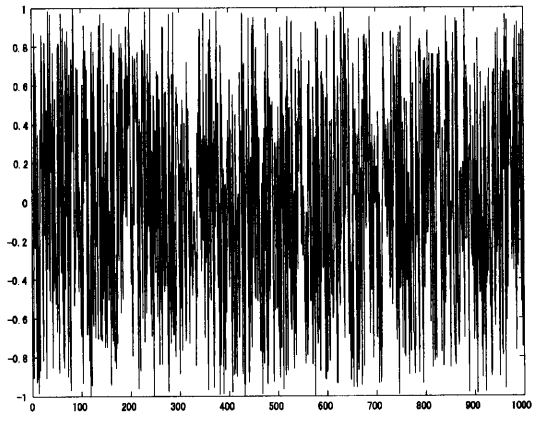

Fig.6 Input signal for identification.

6. Conclusions

The present study proposes OLS for RBF Network in RKHS. The idea is from support vector ma-

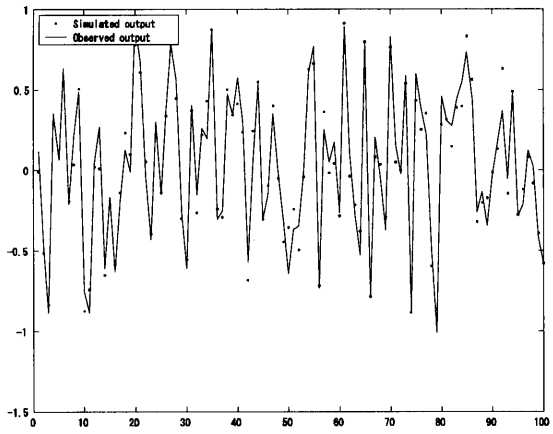

Fig.7 Observed output and predicted output from identification.

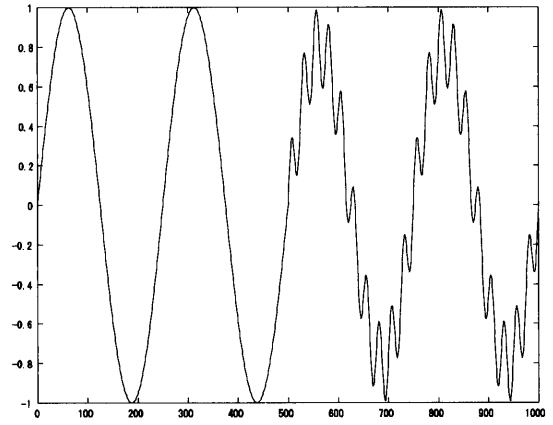

Fig.8 Input signal for testing.

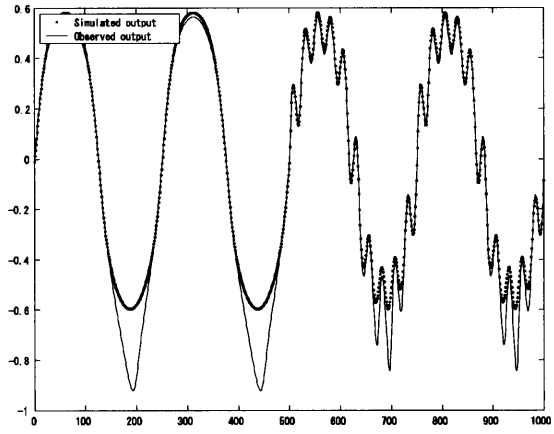

Fig.9 Predicted output from testing.

chine of mapping the data into the high-dimensional feature space, which is known as Reproducing Kernel Hilbert Space. Then, perform the sparse RBF network by OLS subset selection for RBF network approach. The curious fact about using Mercer kernel is that it does not need to know the underlying feature map in order to be able to learn in the feature space. The new basis function derived by means of kernel function are obatained and illustrated. The simulation results illustrate that this new learning strategy offers a powerful procedure for fitting adequate and parsimonious regression model in RKHS for practical signal processing.

References

1) N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods.* Cambridge University Press, 2000.
2) B. Schölkopf, A.J. Smola, and K.R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
3) S. Chen, C.F.N. Cowan, and P.M. Grant. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions Neural Networks*, 2:302–309, 1991.
4) B. Schölkopf, C.J.C. Burges, and A.J. Smola. *Advances in kernel methods support vector learning.* The MIT Press, Cambridge, Massachusetts, London England, 1998.
5) K.S. Narendra and K. Parthasathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks.*, 1:4–27, 1990.
6) R. Rosipal and L.J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, pages 97–123, December 2(2001).
7) R. Rosipal, M. Girolami, L.J. Trejo, and A. Cichocki. Kernel PCA for feature extraction and de-noising in non-linear regression. *Neural Computing & Applications*, 10(3), 2001.
8) B. Schölkopf. Support vector learning. *PhD thesis, Universitat Berlin, Berlin, Germany*, 1997.
9) M.G. Genton. Classes of kernels for machine learning : a statistics perspective. *Journal of Machine Learning Research*, pages 299–312, 2(2001).
10) I.T. Jolliffe. *Principal component analysis.* Springer-Verlag, New York, 1986.
11) H.D. Vinod and A. Ullah. *Recent advances in regression methods.* Marcel Dekker,INC., 1981.