Non-linear system identification using particle swarm optimisation tuned radial basis function models

Sheng Chen*

School of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
E-mail: sqc@ecs.soton.ac.uk
*Corresponding author

Xia Hong

School of Systems Engineering,
University of Reading,
Reading RG6 6AY, UK
E-mail: x.hong@reading.ac.uk

Bing L. Luk

Department of Manufacturing Engineering and Engineering Management,
City University of Hong Kong,
Hong Kong, China
E-mail: meblluk@cityu.edu.hk

Chris J. Harris

School of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
E-mail: cjh@ecs.soton.ac.uk

Abstract: A novel particle swarm optimisation (PSO) tuned radial basis function (RBF) network model is proposed for identification of non-linear systems. At each stage of orthogonal forward regression (OFR) model construction process, PSO is adopted to tune one RBF unit's centre vector and diagonal covariance matrix by minimising the leave-one-out (LOO) mean square error (MSE). This PSO aided OFR automatically determines how many tunable RBF nodes are sufficient for modelling. Compared with the-state-of-the-art local regularisation assisted orthogonal least squares algorithm based on the LOO MSE criterion for constructing fixed-node RBF network models, the PSO tuned RBF model construction produces more parsimonious RBF models with better generalisation performance and is often more efficient in model construction. The effectiveness of the proposed PSO aided OFR algorithm for constructing tunable node RBF models is demonstrated using three real data sets.

Keywords: non-linear system; orthogonal least squares algorithm; OLS algorithm; leave-one-out cross validation; tunable radial basis function network; tunable RBF network; particle swarm optimisation; PSO.

Reference to this paper should be made as follows: Chen, S., Hong, X., Luk, B.L. and Harris, C.J. (2009) 'Non-linear system identification using particle swarm optimisation tuned radial basis function models', *Int. J. Bio-Inspired Computation*, Vol. 1, No. 4, pp.246–258.

Biographical notes: Sheng Chen received his BEng from Huadong Petroleum Institute, China, in January 1982 and PhD from City University, London, UK, in September 1986, both in Control Engineering. He was awarded DSc by University of Southampton, UK, in 2005. He has been with School of Electronics and Computer Science, University of Southampton, since 1999. He is a Fellow of IET and a Fellow of IEEE. His research interests include wireless communications, machine learning, intelligent control systems and evolutionary computation.

Xia Hong received her BSc and MSc from National University of Defence Technology, China, in 1984 and 1987, respectively and PhD from University of Sheffield, Sheffield, UK, in 1998, all in automatic control. She has been with School of Systems Engineering, University of Reading, Reading, UK, since 2001. Her research interests include non-linear systems identification, data modelling, estimation and intelligent control, neural networks, pattern recognition, learning theory and their applications. She was awarded a Donald Julius Groen Prize by IMechE in 1999.

Bing L. Luk received his BSc in Electrical and Electronic Engineering from Portsmouth Polytechnic, UK, in 1985, MSc in Digital Computer Systems from Brunel University, Uxbridge, UK, in 1986 and PhD in Robotics from University of Portsmouth, Portsmouth, UK, in 1991. Since 2000, he has been with Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China. His research interests include mobile robotics, non-destructive test methods, machine learning and evolutionary computation.

Chris J. Harris received his PhD from the University of Southampton, Southampton, UK, in 1972. He was awarded with DSc by the University of Southampton in 2001. He is a Professor of Computational Intelligence at University of Southampton. His research interests include intelligent autonomous systems, intelligent control, estimation of dynamic processes and multi-sensor data fusion. He is a fellow of UK Royal Academy of Engineering. He was awarded with IEE Faraday medal in 2001 for his work in intelligent control and neurofuzzy systems.

1 Introduction

The radial basis function (RBF) network is popular neural network architecture for non-linear system modelling and identification (Chen et al., 1990a). Like other neural network models, the parameters of a RBF network, which includes nodes' centre vectors and variances or covariance matrices as well as the connecting weights, can be estimated based on non-linear optimisation using gradient descent algorithms (Chen et al., 1990c; McLoone et al., 1998), the expectation-maximisation algorithm (Yang and Chen, 1998; Mak and Kung, 2000) or the population-based evolutionary algorithms (Gonzalez et al., 2003; Sun et al., 2006; Feng, 2006; Cardi et al., 2008; Guerra and Coelho, 2008). Generally speaking, these non-linear estimation methods are computationally expensive. Moreover, the RBF model structure or the number of RBF nodes has to be determined via other means, typically based on costly cross validation. Clustering algorithms can alternatively be applied to find the RBF centre vectors as well as the associated basis function variances (Moody and Darken, 1989; Chen et al., 1992; Chen, 1995). The remaining RBF weights can then be determined using the simple linear least squares (LS) estimate. However, the number of the clusters again has to be determined via cross validation.

A most popular approach for identifying RBF network models, however, is to consider the training input data points as candidate RBF centres and to employ a common variance for every RBF node. The resulting fixed-node RBF model becomes linear in the RBF weights and a parsimonious RBF network can then be identified efficiently using the orthogonal least squares (OLS) algorithm (Chen et al., 1989, 1991, 2003, 2004). Similarly, the support vector machine (SVM) and other sparse kernel modelling methods (Vapnik, 1995; Gunn, 1998; Tipping, 2001; Schölkopf and Smola, 2002) also place the kernel centres to the training input data points and adopt a common kernel variance for every kernel. A sparse kernel representation is then sought. Since the common variance in this fixed-node RBF model is not provided by the learning algorithms, it must be determined via cross validation. For the kernel methods, such as the ε-SVM (Gunn, 1998), the regularisation parameter and the value of ε must also be specified via cross validation. The experimental results of (Chen et al., 2004) shows that the local regularisation assisted OLS (LROLS) algorithm based on the leave-one-out (LOO) cross validation compares favourably with other sparse kernel modelling methods, in terms of model sparsity and generalisation performance as well as efficiency of model construction. This LROLS-LOO algorithm (Chen et al., 2004) offers a state-of-the-art construction method for fixed-node RBF models.

In this paper we consider the tunable RBF model, where each RBF node has a tunable centre vector and an adjustable diagonal covariance matrix. However, unlike in the non-linear optimisation approach (Gonzalez et al., 2003; Sun et al., 2006; Feng, 2006; Cardi et al., 2008; Guerra and Coelho, 2008), we do not attempt to optimise all the RBF parameters together, which could be a too large and complex non-linear optimisation task. Instead, we adopt an orthogonal forward regression (OFR) to optimise RBF units one by one based on the LOO mean square error (MSE). More specifically, we use particle swarm optimisation (PSO) (Kennedy and Eberhart, 1995, 2001) to optimise one RBF node's centre vector and diagonal covariance matrix at each stage of the OFR. Applying a similar OFR to construct the tunable RBF model has been proposed previously (Wang et al., 2006a, 2006b), where a global search algorithm (Chen et al., 2005) is used to optimise the tunable RBF units one by one based on the training MSE and the SVM method is used to compute the weights of the resulting RBF model. Compared with the LROLS-LOO construction algorithm for fixed-node RBF models (Chen et al., 2004), the algorithm of (Wang et al., 2006a, 2006b) offers the advantages of sparser models with equally good

generalisation performance at a cost of much higher computational complexity in model construction.

PSO is a population based stochastic optimisation technique (Kennedy and Eberhart, 1995, 2001) inspired by social behaviour of bird flocking or fish schooling. The algorithm starts with random initialisation of a population of individuals, called particles, within the problem search space. It finds the global best solution by simply adjusting the trajectory of each individual toward its own best location and toward the best particle of the entire swarm at each time. The PSO method is becoming very popular due to its simplicity in implementation, ability to quickly converge to a reasonably good solution and its robustness against local minima. It has been applied to wide-ranging optimisation problems successfully (Kennedy and Eberhart, 2001; Ratnaweera et al., 2004; Guru et al., 2005; Feng, 2006; Soo et al., 2007; Cardi et al., 2008; Guerra and Coelho, 2008; Sun et al., 2008). We demonstrate that the proposed PSO aided OFR algorithm for tunable-node RBF models not only produces sparser models and better generalisation performance but also offers computational advantages in model construction process, compared with the state-of-the-art LROLS-LOO construction algorithm for fixed-node RBF models (Chen et al., 2004).

The remainder of this paper is organised as follows. In Section 2, non-linear system identification is considered using the novel tunable RBF model, while the proposed PSO-OFR algorithm is derived in Section 3. Our experimental results based on three real-life data sets are presented in Section 4. Section 5 offers our conclusions.

2 System identification using tunable RBF models

Consider the class of discrete-time non-linear systems that can be represented by the following NARX structure

$$\begin{aligned} y_k &= f_s(y_{k-1},\cdots,y_{k-m_y},u_{k-1},\cdots,u_{k-m_u})+e_k \\ &= f_s(\mathbf{x}_k)+e_k \end{aligned} \tag{1}$$

where u_k and y_k are the system input and output variables, respectively, m_u and m_y are the known lags for u_k and y_k, respectively, e_k is a zero-mean uncorrelated observation noise, $f_s(\bullet)$ denotes the unknown system mapping, and

$$\begin{aligned} \mathbf{x}_k &= \left[x_{1,k}\ \ x_{2,k}\cdots x_{m,k}\right]^T \\ &= \left[y_{k-1}\cdots y_{k-m_y}\ \ u_{k-1}\cdots u_{k-m_u}\right]^T \end{aligned} \tag{2}$$

denotes the system input vector with the known dimension $m = m_y + m_u$. The NARX system (1) is a special case of the generic NARMAX system that takes the form (Chen and Billings, 1989)

$$\begin{aligned} y_k = f_s(&y_{k-1},\ldots,y_{k-m_y},u_{k-1},\ldots,u_{k-m_u}, \\ &e_{k-1},\ldots,e_{k-m_e})+e_k. \end{aligned} \tag{3}$$

The techniques developed for the NARX structure can be extended to the general NARMAX system (Chen and Billings, 1989; Chen et al., 1989, 1990b).

Given the training data set $D_K = \{\mathbf{x}_k, y_k\}_{k=1}^K$, the task is to identify the system (1) using the RBF network model

$$\hat{y}_k^{(M)} = \hat{f}_{RBF}^{(M)}(\mathbf{x}_k) = \sum_{i=1}^{M}\theta_i p_i(\mathbf{x}_k) = \mathbf{p}_M^T(k)\boldsymbol{\theta}_M \tag{4}$$

where $\hat{f}_{RBF}^{(M)}(\bullet)$ denotes the mapping of the M-term RBF model, M is the number of RBF units, $\boldsymbol{\theta}_M = [\theta_1\ \theta_2 \ldots\ \theta_M]^T$ is the RBF weight vector and

$$\mathbf{p}_M^T(k) = \left[p_1(\mathbf{x}_k)\ p_2(\mathbf{x}_k)\ldots p_M(\mathbf{x}_k)\right] \tag{5}$$

is the response vector of the M RBF nodes to the input $\mathbf{x}_k$. In this study, we consider the general RBF regressor of the form

$$p_i(\mathbf{x}) = \varphi\left(\sqrt{(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}\right), \tag{6}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i = \mathrm{diag}\{\sigma_{i,1}^2,\sigma_{i,2}^2,\ldots,\sigma_{i,m}\}$ are the centre vector and diagonal covariance matrix of the ith RBF node, respectively, $\varphi(\bullet)$ is the chosen RBF basis function. In this study, the Gaussian basis function is employed.

Let us define the modelling error at the kth training data point $(\mathbf{x}_k, y_k)$ as

$$\varepsilon_k^{(M)} = yk - \hat{y}_k^{(M)}. \tag{7}$$

Then the regression model (4) over the training set D_K can be written in the matrix form

$$\mathrm{y} = \mathbf{P}_M\boldsymbol{\theta}_M + \boldsymbol{\varepsilon}^{(M)}, \tag{8}$$

where $\mathrm{y} = [y_1\ y_2 \ldots y_K]^T$ is the desired output vector, $\boldsymbol{\varepsilon}^{(M)} = [\varepsilon_1^{(M)}\ \varepsilon_2^{(M)}\ldots\varepsilon_K^{(M)}]^T$ is the modelling error vector of the M-term model and the regression matrix $\mathbf{P}_M = [\mathbf{p}_1\ \mathbf{p}_2\cdots\mathbf{p}_M]$ with the ith regressor given by

$$\mathbf{p}_i = [p_i(\mathbf{x}_1)\ p_i(\mathbf{x}_2)\ \ldots\ p_i(\mathbf{x}_K)]^T, \tag{9}$$

where $1 \le i \le M$. Note that $\mathbf{p}_k$ is the kth column of $\mathbf{P}_M$ while $\mathbf{p}_M^T(k)$ denotes the kth row of $\mathbf{P}_M$.

2.1 Orthogonal decomposition

Let an orthogonal decomposition of $\mathbf{P}_M$ be $\mathbf{P}_M = \mathbf{W}_M \mathbf{A}_M$, where

$$\mathbf{A}_M = \begin{bmatrix} 1 & \alpha_{1,2} & \cdots & \alpha_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha_{M-1,M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{10}$$

and $\mathbf{W}_M = [\mathrm{w}_1\ \mathrm{w}_2 \ldots \mathrm{w}_M]$ with the orthogonal columns that satisfy $\mathrm{w}_i^T \mathrm{w}_l = 0$ for $l \neq i$. The regression model (8) can alternatively be presented as

$$\mathrm{y} = \mathbf{W}_M \mathbf{g}_M + \varepsilon^{(M)}, \tag{11}$$

where $\mathbf{g}_M = \left[g_1\ g_2 \ldots g_M\right]^T$ satisfies the triangular system $\mathbf{A}_M \boldsymbol{\theta}_M = \mathbf{g}_M$. Since the space spanned by the original model bases $p_i(\bullet), 1 \le i \le M$, is identical to the space spanned by the orthogonal model bases, the RBF model output can equivalently be expressed as

$$\hat{y}_k^{(M)} = \mathrm{w}_M^T(k)\, \mathbf{g}_M, \tag{12}$$

where $\mathrm{w}_M^T(k) = \left[w_1(k)\ w_2(k) \ldots w_M(k)\right]$ is the kth row of $\mathbf{W}_M$.

Orthogonal decomposition can be carried out for example using the Gram-Schmidt orthogonalisation procedure (Chen et al., 1989). Using the model (11) instead of the original one (8) facilitates an efficient OFR model construction process. In particular, calculation of the LOO MSE becomes very fast, making it possible to construct the model by directly optimising the model generalisation capability rather than minimising the usual training MSE (Chen et al., 2004).

2.2 OFR based on LOO cross validation

The evaluation of model generalisation capability is directly based on the concept of cross validation (Stone, 1974) and a commonly used cross validation is the LOO cross validation with its associated LOO test MSE (Myers, 1990). Consider the OFR modelling process that has produced the n-node RBF model. Let us denote the constructed *n* columns of regressors as $\mathbf{W}_n = \left[\mathrm{w}_1\ \mathrm{w}_2 \ldots \mathrm{w}_n\right]$, the *k*th model output of this n-node RBF model identified using the entire training data set D_K as

$$\hat{y}_k^{(n)} = \sum_{i=1}^{n} g_i w_i(k), \tag{13}$$

and the corresponding kth modelling error as $\varepsilon_k^{(n)} = y_k - \hat{y}_k^{(n)}$. If we 'remove' the kth data point from the training set D_K and use the remaining $K-1$ data points $D_K \setminus (\mathbf{x}_k, y_k)$ to identify the n-node RBF model instead, the 'test' error of the resulting model can be calculated on the data point $(\mathbf{x}_k, y_k)$ not used in training. This LOO modelling error, denoted as $\varepsilon_k^{(n,-k)}$, is given by (Myers, 1990)

$$\varepsilon_k^{(n,-k)} = \frac{\varepsilon_k^{(n)}}{\eta_k^{(n)}}, \tag{14}$$

where $\eta_k^{(n)}$ is referred to as the LOO error weighting (Myers, 1990). The LOO MSE for the n-node RBF model is defined as

$$J_n = E\left[\left(\varepsilon_k^{(n,-k)}\right)^2\right] \approx \frac{1}{K}\sum_{k=1}^{K}\left(\varepsilon_k^{(n,-k)}\right)^2. \tag{15}$$

The LOO MSE J_n is a measure of the model generalisation capability (Stone, 1974; Myers, 1990). For the model (11), J_n can be computed very efficiently because $\varepsilon_k^{(n)}$ and $\eta_k^{(n)}$ can be calculated recursively using (Chen et al., 2004)

$$\varepsilon_k^{(n)} = y_k - \sum_{i=1}^{n} g_i w_i(k) = \varepsilon_k^{(n-1)} - g_n w_n(k) \tag{16}$$

and

$$\eta_k^{(n)} = 1 - \sum_{i=1}^{n} \frac{w_i^2(k)}{\mathrm{w}_i^T \mathrm{w}_i + \lambda} = \eta_k^{(n-1)} - \frac{w_n^2(k)}{\mathrm{w}_n^T \mathrm{w}_n + \lambda}, \tag{17}$$

respectively, where $\lambda \ge 0$ is a small regularisation parameter (Chen et al., 2004). The regularisation parameter can simply be set to $\lambda = 0$ (no regularisation) or a very small value (10^{-6}).

We can use an OFR procedure based on this LOO MSE to construct the RBF nodes one by one. At the nth stage of the construction, the nth RBF node is determined by minimising J_n with respect to the node's centre vector $\boldsymbol{\mu}_n$ and diagonal covariance matrix $\boldsymbol{\Sigma}_n$

$$\min_{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n} J_n\left(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\right). \tag{18}$$

In the next section, we will detail how to use PSO to perform this optimisation. Note that the LOO MSE J_n is locally convex with respect to the model size n (Chen et al., 2004). Thus, there exists an 'optimal' number of RBF nodes M such that: for $n \le M$ J_n decreases as the model size n increases while

$$J_M \le J_{M+1}. \tag{19}$$

Therefore, this OFR construction process is automatically terminated when the condition (19) is met, yielding a very small model set containing only M RBF nodes. After constructing the M-node RBF model using this OFR procedure, we may apply the LROLS-LOO algorithm of (Chen et al., 2004) to automatically update individual regularisation parameter for each RBF weight which may further reduce the model size. This refinement with the LROLS-LOO algorithm requires a very small amount of computation since the regression matrix $\mathbf{P}_M$ is completely specified with only a few columns.

3 PSO aided OFR

The task at the nth stage of the OFR for constructing a tunable RBF network is to solve the optimisation problem (18). Since this optimisation problem is non-convex with respect to $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$, a gradient-based algorithm may become trapped at a local minimum. Alternatively, global search methods, such as the genetic algorithm (Goldberg, 1989; Man et al., 1998), the adaptive simulated annealing (Ingber, 1993; Chen and Luk, 1999) or the repeated weighted boosting search (Chen et al., 2005) may be used to perform this optimisation task. In this study, we adopt PSO (Kennedy and Eberhart, 1995, 2001) to determine $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$. Our study demonstrates that PSO is particularly suited for the optimisation task (18). In fact, our experimental results will show that the PSO aided OFR for constructing tunable-node RBF models is computationally more efficient than the LROLS-LOO algorithm for constructing fixed-node RBF models.

3.1 Particle swarm optimisation

In a PSO algorithm (Kennedy and Eberhart, 1995, 2001), a group of S particles that represent potential solutions are initialised over the search space randomly. Each particle has a fitness value associated with it, based on the related cost function of the optimisation problem and this fitness value is evaluated at each iteration. The best position, $\mathbf{pbst}$, visited by each particle provides the particle the so called 'cognitive information', while the best position so far among the entire group, $\mathbf{gbst}$, offers the 'social information'. The $\mathbf{pbst}$s and $\mathbf{gbst}$ are updated at each iteration. Each particle has its own velocity to direct its flying, which relies on its previous speed as well as its cognitive and social information. In each iteration, the velocity and the position of the particle are updated based on the following equations

$$\begin{aligned}\mathbf{v}_i^{[l+1]} = \xi * \mathbf{v}_i^{[l]} &+ rand() * c_1 * (\mathbf{pbst}_i^{[l]} - \mathbf{u}_i^{[l]}) \\ &+ rand() * c_2 * (\mathbf{gbst}^{[l]} - \mathbf{u}_i^{[l]}),\end{aligned} \tag{20}$$

$$\mathbf{u}_i^{[l+1]} = \mathbf{u}_i^{[l]} + \mathbf{v}_i^{[l+1]}, \tag{21}$$

l iteration index, $1 \le l \le L$, L is the maximum number of iterations

i particle index, $1 \le i \le S$, S is the particle size

$\mathbf{v}_i^{[l]}$ velocity of ith particle at lth iteration. The jth elements of $\mathbf{v}_i^{[l]}$ are in the range $\left[-V_{j_{max}}, V_{j_{max}}\right]$

ξ inertia weight

c_j the acceleration coefficients, j = 1, 2

$rand()$ uniform random number between 0 and 1

$\mathbf{u}_i^{[l]}$ position of ith particle at lth iteration. The jth elements of $\mathbf{u}_i^{[l]}$ are in the range $\left[U_{j_{min}}, U_{j_{max}}\right]$

$\mathbf{pbst}_i^{[l]}$ best position that the ith particle has visited up to lth iteration

$\mathbf{gbst}^{[l]}$ best position that all the particles have visited up to lth iteration

It is reported in (Ratnaweera et al., 2004) that using a time varying acceleration coefficient (TVAC) can enhance the performance of PSO. The reason is that at the initial stages, a large cognitive component and a small social component help particles to wander around the search space and to avoid local minima. In the later stages, a small cognitive component and a large social component help particles to converge quickly to a global minimum. We adopt this TVAC mechanism as suggested in (Ratnaweera et al., 2004), in which c_1 for the cognitive component is reduced from 2.5 to 0.5 and c_2 for the social component varies from 0.5 to 2.5 during the iterative procedure according to

$$c_1 = 2.5 - \frac{2.0 * l}{1.0 * L} \tag{22}$$

and

$$c_2 = 0.5 + \frac{2.0 * l}{1.0 * L}, \tag{23}$$

respectively. In our experiment, we have found out that using a random inertia weight

$$\xi = rand() \tag{24}$$

achieves better performance than using $\xi = 0$ or constant ξ. If the velocity in (20) approaches zero, it is reinitialised randomly to proportional to the maximum velocity

$$\mathbf{v}_i^{[l+1]}\Big|_j = \pm rand() * \gamma * V_{j_{max}}, \tag{25}$$

where $\mathbf{v}_i^{[l+1]}\Big|_j$ denotes the jth element of $\mathbf{v}_i^{[l+1]}$ and γ = 0.1 is a constant.

3.2 PSO aided OFR for tunable RBF model

The procedure of using the PSO aided OFR to determine the nth RBF node is now summarised. Let u be the vector that contains $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$. The dimension of **u** is thus $2m$. The search space is specified by

$$\begin{cases} U_{j_{min}} = \min\{x_{j,k}, 1 \le k \le K\}, \\ U_{j_{max}} = \max\{x_{j,k}, 1 \le k \le K\}, \end{cases} \quad 1 \le j \le m, \tag{26}$$

and

$$U_{j_{min}} = \sigma^2_{\min}, \; U_{j_{max}} = \sigma^2_{\max}, \; 1+m \le j \le 2m. \tag{27}$$

The velocity bounds are defined by

$$V_{j_{max}} = 0.5 * \left(U_{j_{max}} - U_{j_{min}}\right), \; 1 \le j \le 2m. \tag{28}$$

Give the following initial conditions

$$\left.\begin{aligned} &\varepsilon_k^{(0)} = y_k \text{ and } \eta_k^{(0)} = 1, \; 1 \le k \le K, \\ &J_0 = \tfrac{1}{N}\mathbf{y}^T\mathbf{y} = \tfrac{1}{N}\sum\nolimits_{k=1}^{K} y_k^2. \end{aligned}\right\} \tag{29}$$

Specify the number of iterations *L* and the particle size *S*.

PSO initialisation Randomly generate the particles $\mathbf{u}_i^{[0]}, 1 \le i \le S$, within the search space defined by (26) and (27). Set the initial velocities $\mathbf{v}_i^{[0]} = \mathbf{0}_{2m}, 1 \le i \le S$, where $\mathbf{0}_{2m}$ denotes the zero vector of dimension $2m$. Initialise $J_n\left(\mathbf{gbst}^{[0]}\right)$ and $J_n\left(\mathbf{pbst}_i^{[0]}\right)$ for $1 \le i \le S$ to a value larger than J_0.

Iteration loop For $(l = 0; \; l \le L; \; l++)\{$

Orthogonalisation and cost function evaluation.

1 For $1 \le i \le S$, generate $\mathbf{p}_n^{i)}$ from $\mathbf{u}_i^{[l]}$, the candidates for the nth model column, according to (9) and orthogonalise them according to the Gram-Schmidt orthogonalisation procedure (Chen et al., 1989)

$$\alpha_{j,n}^{i)} = \mathbf{w}_j^T\mathbf{p}_n^{i)} \,/\, \mathbf{w}_j^T\mathbf{w}_j, \; 1 \le j \le n, \tag{30}$$

$$\mathbf{w}_n^{i)} = \mathbf{p}_n^{i)} - \sum_{j=1}^{n-1} \alpha_{j,n}^{i)} \mathbf{w}_j, \tag{31}$$

$$g_n^{i)} = \left(\mathbf{w}_n^{i)}\right)^T \mathbf{y} \,/\, \left(\left(\mathbf{w}_n^{i)}\right)^T \mathbf{w}_n^{i)} + \lambda\right). \tag{32}$$

2 For $1 \le i \le S$, calculate the LOO cost for each $\mathbf{u}_i^{[l]}$

$$\varepsilon_k^{(n)}\left(i\right) = \varepsilon_k^{(n-1)} - w_n^{i)}\left(k\right) g_n^{i)}, \; 1 \le k \le K, \tag{33}$$

$$\eta_k^{(n)}\left(i\right) = \eta_k^{(n-1)} - \frac{\left(w_n^{i)}\left(k\right)\right)^2}{\left(\mathbf{w}_n^{i)}\right)^T \mathbf{w}_n^{i)} + \lambda}, \; 1 \le k \le K, \tag{34}$$

$$J_n^{i)} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\varepsilon_k^{(n)}\left(i\right)}{\eta_k^{(n)}\left(i\right)}\right)^2, \tag{35}$$

where $w_n^{i)}\left(k\right)$ is the kth element of $\mathbf{w}_n^{i)}$.

Update cognitive and social information

1 For $\left(i = 1; \; i \le S; \; \mathrm{i}++\right)$

If $\left(J_n^{i)} < J_n\left(\mathbf{pbst}_i^{[l]}\right)\right)$

$J_n\left(\mathbf{pbst}_i^{[l]}\right) = J_n^{i)}$;

$\mathbf{pbst}_i^{[l]} = \mathbf{u}_i^{[l]}$;

End if;

End for;

2 Find

$$i^* = \arg\min_{1 \le i \le S} \; J_n\left(\mathbf{pbst}_i^{[l]}\right).$$

If $\left(J_n\left(\mathbf{pbst}_{i^*}^{[l]}\right) < J_n\left(\mathbf{gbst}_i^{[l]}\right)\right)$

$J_n\left(\mathbf{gbst}^{[l]}\right) = J_n\left(\mathbf{pbst}_{i^*}^{[l]}\right)$;

$\mathbf{gbst}^{[l]} = \mathbf{pbst}_{i^*}^{[l]}$;

End if;

Update velocities and positions of particles

1 For $\left(i = 1; \; i \le S; \; \mathrm{i}++\right)$

$$\mathbf{v}_i^{[l+1]} = rand() * \mathbf{v}_i^{[l]} + rand() * c_1 * \left(\mathbf{pbst}_i^{[l]} - \mathbf{u}_i^{[l]}\right) + rand() * c_2 * \left(\mathbf{gbst}^{[l]} - \mathbf{u}_i^{[l]}\right);$$

For $(j = 1; \; j \le 2m; \; \mathrm{j}++)$

If $(\mathbf{v}_i^{[l+1]}|_j == 0)$

If $(rand() < 0.5)$

$\mathbf{v}_i^{[l+1]}|_j = rand() * \gamma * V_{j_{max}}$;

Else

$$\mathbf{v}_i^{[l+1]}|_j = -rand() * \gamma * V_{j_{max}};$$

End if;

End if;

If $(\mathbf{v}_i^{[l+1]}|_j > V_{j_{max}})$

$$\mathbf{v}_i^{[l+1]}|_j = V_{j_{max}};$$

Else if $(\mathbf{v}_i^{[l+1]}|_j < -V_{j_{max}})$

$$\mathbf{v}_i^{[l+1]}|_j = -V_{j_{max}};$$

End if;

End for;

End for;

2 For $\left(i = 1;\ i \le S;\ \mathrm{i}++\right)$

$$\mathbf{u}_i^{[l+1]} = \mathbf{u}_i^{[l]} + \mathbf{v}_i^{[l+1]};$$

For $\left(j = 1;\ j \le 2m;\ \mathrm{j}++\right)$

If $(\mathbf{u}_i^{[l+1]}|_j > U_{j_{max}})$

$$\mathbf{u}_i^{[l+1]}|_j = U_{j_{max}};$$

Else if $(\mathbf{u}_i^{[l+1]}|_j < U_{j_{min}})$

$$\mathbf{u}_i^{[l+1]}|_j = U_{j_{min}};$$

End if

End for;

End for;

} End of iteration loop

This yields the solution $\mathbf{u} = \mathbf{gbst}^{[L]}$, i.e. the centre vector $\boldsymbol{\mu}_n$ and the diagonal covariance matrix $\boldsymbol{\Sigma}_n$ of the nth RBF node, the nth model column $\mathbf{p}_n$, the orthogonalisation coefficients $\alpha_{j,n}$, $1 \le j \le n$, the corresponding orthogonal model column $\mathbf{w}_n$ and the weight g_n, as well as the n-term modelling errors $\varepsilon_k^{(n)}$ and the associated LOO error weightings $\eta_k^{(n)}$ for $1 \le k \le K$.

3.3 Computational complexity comparison

The LROLS-LOO algorithm (Chen et al., 2004) is an efficient construction algorithm for fixed-node RBF models. In particular, it has significant computational advantages over many other sparse kernel modelling methods. We therefore compare the computational complexity of the proposed PSO aided OFR for constructing tunable-node RBF models with that of the LROLS-LOO algorithm for constructing fixed-node RBF models.

Given the RBF variance, the LROLS-LOO algorithm involves a few iterations. The first iteration works on the $K \times K$ full regression matrix and selects a subset of M' RBF nodes, where $M' \ll K$. The computational complexity of the algorithm is dominated by this first iteration and the complexity of the rest iterations is negligible. For the LROLS-LOO algorithm, it is straightforward to verify that the computational complexity of one model column orthogonalisation and the associated LOO cost function evaluation is the order of K, $\mathcal{O}(\mathrm{K})$. Thus, we can characterise the complexity of the algorithm by the required number of the LOO cost function evaluations and associated model column orthogonalisations, which is given by

$$C_{\mathrm{LROLS-LOO}} \approx \sum_{i=1}^{M'} (K - (i-1)) \approx M' \times K, \tag{36}$$

where the first approximation is due to the fact that we only count the complexity of the first iteration and the second approximation is arrived because the selected model size M' is much smaller than the training data size K.

Since for the PSO aided OFR algorithm, the computational requirement of one model column orthogonalisation and the associated LOO cost function evaluation is also the order of K, $\mathcal{O}(\mathrm{K})$, we can also characterise the computational requirements of the algorithm by the number of the LOO cost function evaluations and associated model column orthogonalisations. This number is given as

$$C_{\mathrm{PSO-OFS}} = M \times S \times L, \tag{37}$$

where M is the constructed model size, S the particle size and L the number of iterations. The model size M is usually much smaller than the model size M' obtained by the LROLS-LOO algorithm. Our experimental results will show that typically $C_{\mathrm{PSO-OFS}} < C_{\mathrm{LROLS-LOO}}$. Note that the complexity of (37) is the true complexity of the PSO aided OFR algorithm, while the complexity of (36) is the complexity of the LROLS-LOO algorithm given a RBF variance. Since the RBF variance is not provided by the LROLS-LOO algorithm, it must be determined using for example a grid search based cross validation. If an N-point grid search is used to determine the RBF variance, the true complexity of the LROLS-LOO algorithm will be approximately $N \times C_{\mathrm{LROLS-LOO}}$. Taking this fact

into account, computational advantages of the proposed PSO aided OFR algorithm becomes even more significant.

4 Experimental results

Three real data sets were used to investigate the proposed PSO aided OFR for constructing tunable-node RBF models. For these three data sets, our previous work has shown that the LROLS-LOO algorithm for constructing fixed node RBF models (Chen et al., 2004) offers considerable advantages, in terms of model size and generalisation performance as well as computational requirement of modelling process, over many existing sparse kernel modelling methods, such as the SVM. We therefore used this LROLS-LOO algorithm as the benchmark in our experiment.

4.1 Engine data set

The data set contains 410 data pairs of the input u_k (the fuel rack position) and the output y_k (the engine speed), collected from a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed (Billings et al., 1989). The data set is depicted in Figure 1. The first 210 data points were used in modelling and the last 200 points in model validation. The study (Billings et al., 1989) has shown that at low engine speed the system is non-linear and this data set can be modelled adequately as

$$y_k = f_s(\mathbf{x}_k) + e_k, \tag{38}$$

where $f_s(\bullet)$ describes the unknown underlying system to be identified, e_k denotes the system noise and $\mathbf{x}_k = [y_{k-1}\ u_{k-1}\ u_{k-2}]^T$.

We first applied the LROLS-LOO algorithm (Chen et al., 2004) to fit a fixed-node RBF model with a common RBF variance σ^2 to the data set. An appropriate value for the RBF variance was found to be $\sigma^2 = 1.69$ via cross validation. Given the RBF variance $\sigma^2 = 1.69$, the first iteration of the LROLS-LOO algorithm automatically selected $M' = 24$ RBF nodes and the subsequent iterations reduced the final model size to 22 RBF nodes. The complexity of the algorithm given $\sigma^2 = 1.69$ was therefore $C_{\text{LROLS-LOO}}$ = 24 × 210 = 5,040. The performance of this 22-term RBF network model are summarised in Table 1.

We next applied the PSO aided OFR algorithm to construct a tunable-node RBF model. It was found empirically that setting the particle size to S = 10 and the number of iterations to L = 20 were sufficient. The model construction process is illustrated in Figure 2, where it is seen that the PSO aided OFR algorithm automatically constructed a tunable-node RBF network of M = 22 units since $J_{22} < J_{23}$. The LROLS-LOO algorithm was then applied to this 22-term model set to obtain a final model of 15 units. The performance of this 15-node RBF model are given in Table 1, in comparison with the benchmark results of the 22-term RBF model constructed by the LROLS-LOO algorithm. The complexity of the PSO aided OFR algorithm was calculated as $C_{\text{PSO-OFS}}$ = 22 × 10 × 20 = 4,400. Figure 3 depicts the model prediction $\hat{y}_k$ and the prediction error $\varepsilon_k = y_k - \hat{y}_k$ generated by the 15-node tunable RBF model constructed by the PSO aided OFR algorithm. In comparison with the benchmark LROLS-LOO algorithm, the PSO aided OFR algorithm not only produced a smaller RBF model with better test MSE performance but also was more efficient in modelling process. Note that the computational advantage of the PSO aided OFR was much more significant than shown in Table 1 as we did not count the computational requirements for determining an appropriate RBF variance needed by the LROLS-LOO algorithm.

Figure 1 Engine data set: (a) system input u_k, and (b) system output y_k

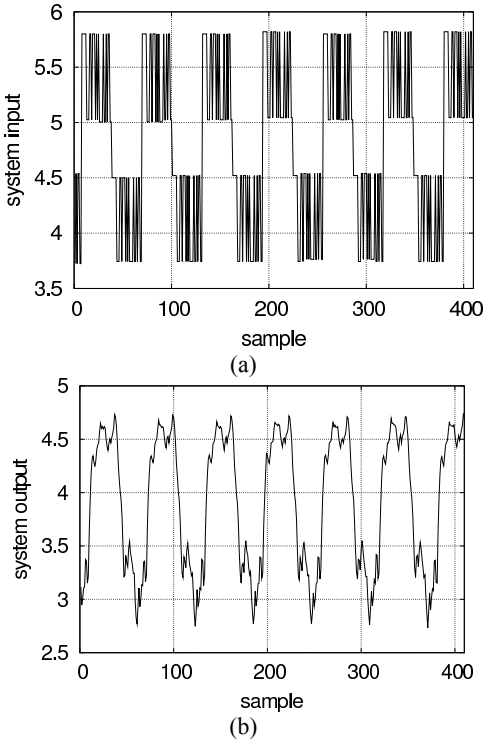


Table 1 Comparison of the two Gaussian RBF network models obtained by the LROLS-LOO and PSO-OFR algorithms for the engine data set

Algorithm	*RBF type*	*Model size*	*Training MSE*
LROLS-LOO	Fixed	22	0.000453
PSO-OFR	Tunable	15	0.000426
Algorithm	*Test MSE*		*Complexity*
LROLS-LOO	0.000490		5,040
PSO-OFR	0.000466		4,400

Figure 2 Leave-one-out MSE versus model size for the engine data set obtained by the PSO aided OFR

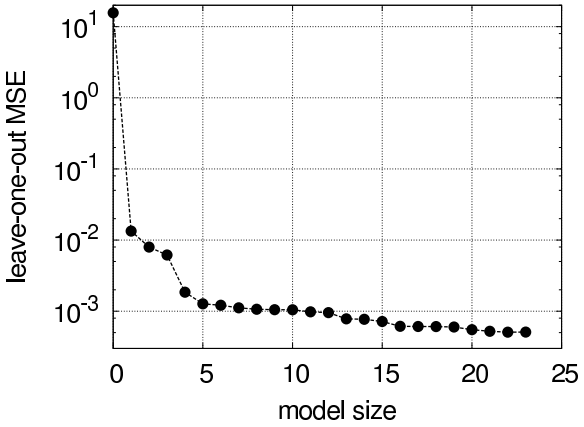
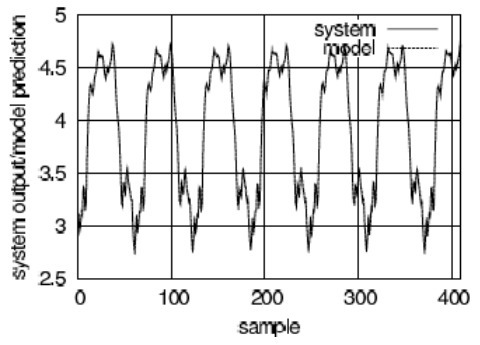
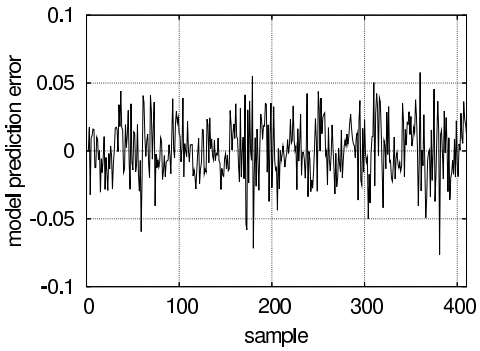


Figure 3 Modelling of the engine data set by the 15-node RBF network constructed using the PSO aided OFR algorithm, (a) model prediction $\hat{y}_k$ superimposed on system output y_k (b) model prediction error $\varepsilon_k = y_k - \hat{y}_k$



(a)



(b)

4.2 Gas furnace data set

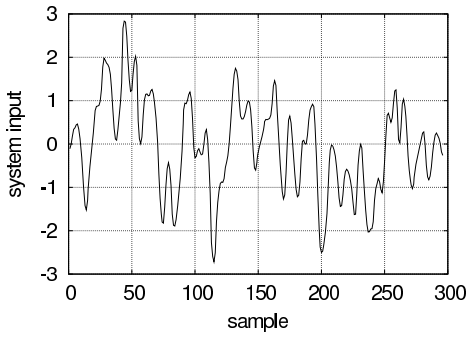
The gas furnace data set was the time series *J* in (Box and Jenkins, 1976). The data set contained 296 pairs of input-output points as depicted in Figure 4, where the input u_k was the coded input gas feed rate and the output y_k represented the CO_2 concentration from the gas furnace. From the 296 pairs of input and output data, we constructed 296 data points $\{\mathbf{x}_k, y_k\}$ with $\mathbf{x}_k$ given by

$$\mathbf{x}_k = [y_{k-1}\ y_{k-2}\ y_{k-3}\ u_{k-1}\ u_{k-2}\ u_{k-3}]^T. \quad (39)$$

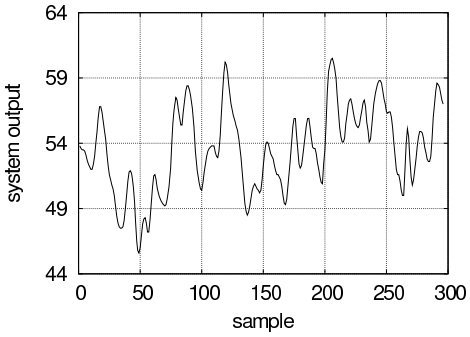
From Figure 4, it can be observed that the second half of the data set was different from the first half. Therefore, we used the even-number pairs of $\{\mathbf{x}_k, y_k\}$ for training and the odd-number pairs of $\{\mathbf{x}_k, y_k\}$ for testing. Both the training and testing sets had 148 data points.

For the fixed-node RBF model, an adequate RBF variance was found to be σ^2 = 1,000.0 after a grid search based cross validation using the LROLS-LOO algorithm. Given σ^2 = 1,000.0, the first iteration of the LROLS-LOO algorithm terminated with M' = 13 and further iterations yielded a final model of 12 RBF nodes. The modelling performance of the LROLS-LOO algorithm are summarised in Table 2, where the complexity was calculated according to (36) for the given RBF variance σ^2 = 1,000.0.

Figure 4 Gas furnace data set: (a) system input u_k, and (b) system output y_k



(a)



(b)

Table 2 Comparison of the two Gaussian RBF network models obtained by the LROLS-LOO and PSO-OFR algorithms for the gas furnace data set

Algorithm	*RBF type*	*Model size*	*Training MSE*
LROLS-LOO	fixed	12	0.047448
PSO-OFR	tunable	8	0.041639
Algorithm	*Test MSE*		*Complexity*
LROLS-LOO	0.080491		1,924
PSO-OFR	0.078884		1,600

Figure 5 Leave-one-out MSE versus model size for the gas furnace data set obtained by the PSO aided OFR

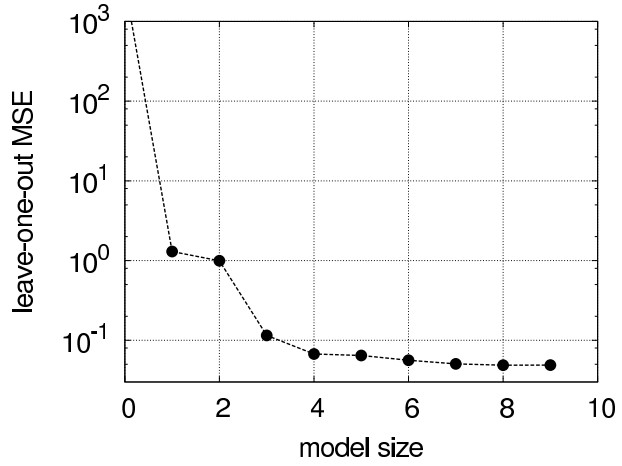
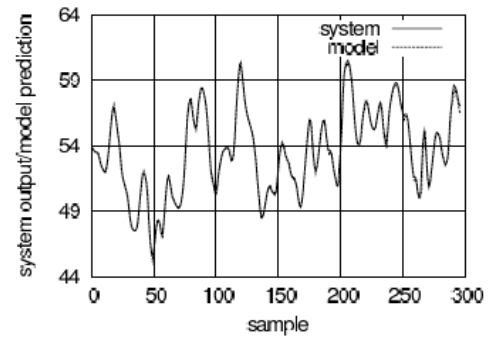
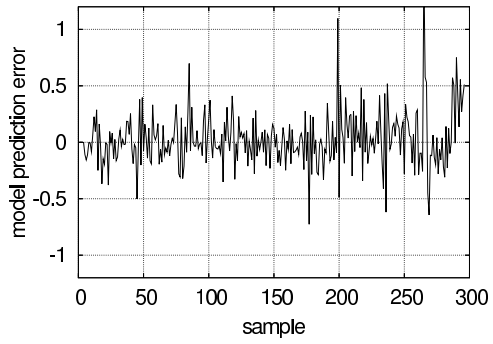


Figure 6 Modelling of the gas furnace data set by the 8-node RBF network constructed using the PSO aided OFR algorithm (a) model prediction $\hat{y}_k$ superimposed on system output y_k (b) model prediction error $\varepsilon_k = y_k - \hat{y}_k$



(a)



(b)

For constructing the tunable-node RBF model using the PSO aided OFR algorithm, we set the particle size to $S = 10$ and the number of iterations to $L = 20$. Figure 5 depicted the LOO MSE as a function of the model size. It can be seen from Figure 5 that $J_8 < J_9$. Thus, the PSO aided OFR algorithm terminated with an 8-node RBF network. No further reduction in the model size was achieved by applying the LROLS-LOO algorithm to this 8-term model set. The performance of the 8-node RBF model constructed by the PSO aided OFR are given in Table 2, in comparison with the results of the 12-term RBF model selected by the LROLS-LOO algorithm. Figure 6 depicts the model prediction $\hat{y}_k$ and the prediction error $\varepsilon_k = y_k - \hat{y}_k$ produced by the 8-node RBF model constructed using the PSO aided OFR algorithm. For this example, again the PSO aided OFR algorithm offered clear advantages in more parsimonious model and better generalisation performance as well as more efficient model construction, compared with the benchmark LROLS-LOO algorithm.

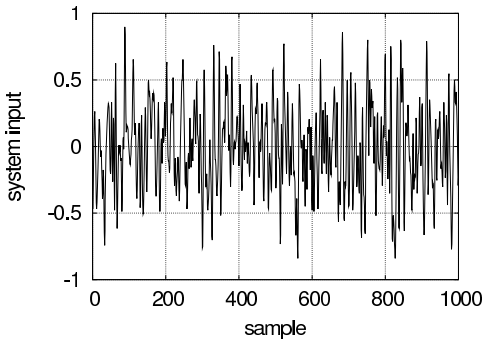
4.3 Liquid level data set

The data set was collected from a non-linear liquid level system, which consisted of a DC water pump feeding a conical flask which in turn fed a square tank. The system input u_k was the voltage to the pump motor and the system output y_k was the water level in the conical flask. A description of this non-linear process can be found in (Billings and Voon, 1986) and Figure 7 shows the 1,000 data points of the data set used in this experiment. From the data set, 1,000 data points $\{\mathbf{x}_k, y_k\}$ were constructed with $\mathbf{x}_k$ given by

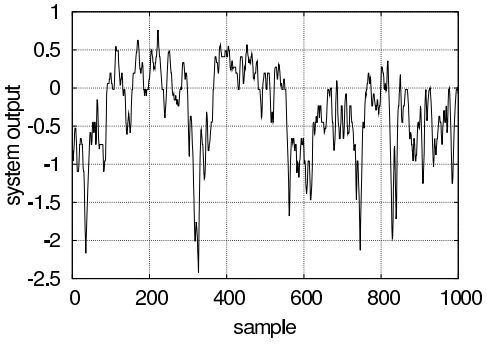
$$\mathbf{x}_k = [y_{k-1}\ y_{k-2}\ y_{k-3}\ u_{k-1}\ u_{k-2}\ u_{k-3}\ u_{k-4}]^T. \tag{40}$$

The first 500 pairs of the data were used for training and the remaining 500 pairs for testing the constructed model.

Figure 7 Liquid level data set (a) system input u_k (b) system output y_k



(a)



(b)

For the fixed-node RBF model with every training input data used as a candidate RBF centre vector, an appropriate RBF variance was found to be $\sigma^2 = 2.0$ via a grid search based cross validation using the LROLS-LOO algorithm. With $\sigma^2 = 2.0$, the first iteration of the LROLSLOO algorithm produced a model set of $M' = 30$ nodes and further iterations did not reduce the final model size. The results obtained by the LROLS-LOO algorithm are given in Table 3, where the complexity was computed as $C_{\text{LROLS–LOO}} = 30 \times 500 = 15{,}000$ for the fixed $\sigma^2 = 2.0$.

Table 3 Comparison of the two Gaussian RBF network models obtained by the LROLS-LOO and PSO-OFR algorithms for the liquid level data set

Algorithm	*RBF type*	*Model size*	*Training MSE*
LROLS-LOO	fixed	30	0.001400
PSO-OFR	tunable	20	0.001461
Algorithm	*Test MSE*		*Complexity*
LROLS-LOO	0.002532		15,000
PSO-OFR	0.002463		4,200

We again set the particle size to $S = 10$ and the number of iterations to $L = 20$. The process of constructing the tunable-node RBF model using the PSO aided OFR algorithm is illustrated in Figure 8, where it can be seen that the algorithm produced a model set of $M = 21$ nodes since $J_{21} < J_{22}$. Applying the LROLS-LOO algorithm to this 21-term model set yielded the final model containing 20 nodes. The results produced by the PSO aided OFR are also listed in Table 3. Figure 9 shows the model prediction $\hat{y}_k$ and the prediction error $\varepsilon_k = y_k - \hat{y}_k$ produced by the 20-node RBF model constructed using the PSO aided OFR algorithm. For this example, the PSO aided OFR algorithm has clear advantages over the benchmark LROLS-LOO algorithm, in terms of model size and generalisation capability as well as complexity of model construction.

Figure 8 Leave-one-out MSE versus model size for the liquid level data set obtained by the PSO aided OFR

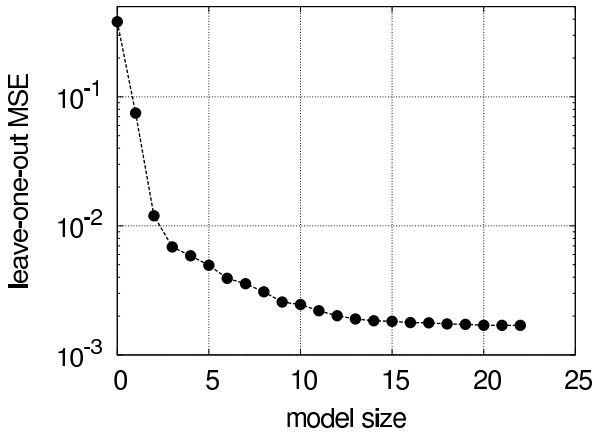
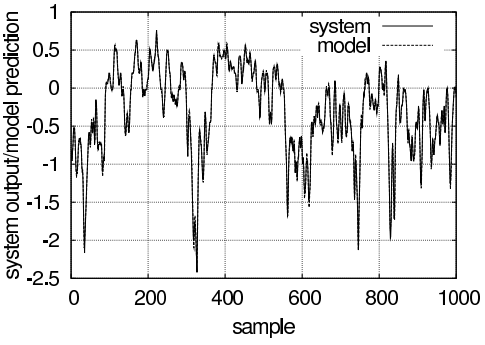
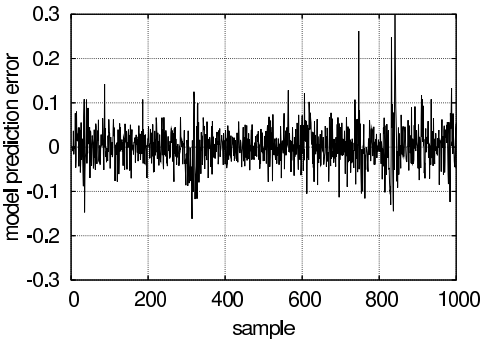


Figure 9 Modelling of the liquid level data set by the 20-node RBF network constructed using the PSO aided OFR algorithm (a) model prediction $\hat{y}_k$ superimposed on system output y_k (b) model prediction error $\varepsilon_k = y_k - \hat{y}_k$



(a)



(b)

5 Conclusions

In this contribution we have proposed a novel PSO aided OFR algorithm to construct tunable-node RBF network models for non-linear system identification. Unlike the standard fixed-node RBF model where the RBF centre vectors are placed at the training input data points and a common RBF variance is used for every RBF node, the proposed algorithm optimises one RBF node's centre vector and diagonal covariance matrix by minimising the LOO MSE at each stage of the OFR. The model construction procedure automatically determines how many tunable nodes are sufficient and PSO ensures that this model construction procedure is computationally very efficient. Using the best existing algorithm for constructing fixed-node RBF models, the LROLS-LOO algorithm, as the benchmark, the experimental results involving three real-life data sets have confirmed that the proposed PSO aided OFR algorithm for constructing tunable-node RBF models offers clear advantages over the benchmark LROLS-LOO algorithm for constructing fixed-node RBF models, in terms of more parsimonious model and better generalisation performance as well as more efficient model construction.

In the proposed PSO aided OFR procedure, the RBF units are optimised one by one. A criticism is that this is a suboptimal procedure. Alternatively, one could optimise all

the RBF units together using the PSO, as in (Sun et al., 2006; Feng, 2006; Cardi et al., 2008; Guerra and Coelho, 2008). However, as emphasised in the introduction section, this fully non-linear optimisation approach will be computationally very expensive and, moreover, there is no guarantee that a better RBF model can always be obtained, compared with our much simpler approach. Our proposed approach can be viewed as combining both the non-linear and linear learning methods. It provides greater modelling capability of the non-linear approach while offers computational simplicity of the linear fixed-node approach. In fact, as demonstrated clearly in the experimental results, the computational complexity of the proposed PSO aided OFR procedure is simpler than the best linear fixed-node approach, the LROLS-LOO algorithm.

References

Billings, S.A. and Voon, W.S.F. (1986) 'A prediction-error and stepwise-regression estimation algorithm for nonlinear systems', *Int. J. Control*, Vol. 44, pp.803–822.

Billings, S.A., Chen, S. and Backhouse, R.J. (1989) 'The identification of linear and non-linear models of a turbocharged automotive diesel engine', *Mechanical Systems and Signal Processing*, Vol. 3, No. 2, pp.123–142.

Box, G.E.P. and Jenkins, G.M. (1976) *Time Series Analysis, Forecasting and Control*, Holden Day Inc.

Cardi, A.A., Firpi, H.A., Bement, M.T. and Liang, S.Y. (2008) 'Workpiece dynamic analysis and prediction during chatter of turning process', *Mechanical Systems and Signal Processing*, Vol. 22, No. 6, pp.1481–1494.

Chen, S. (1995). 'Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning', *Electronics Letters*, Vol. 31, No. 2, pp.117–118.

Chen, S. and Billings, S.A. (1989) 'Representation of nonlinear systems: The NARMAX model', *Int. J. Control*, Vol. 49, No. 3, pp.1013–1032.

Chen, S. and Luk, B.L. (1999) 'Adaptive simulated annealing for optimization in signal processing applications', *Signal Processing*, Vol. 79, No. 1, pp.117–128.

Chen, S., Billings, S.A. and Grant, P.M. (1992) 'Recursive hybrid algorithm for non-linear system identification using radial basis function networks', *Int. J. Control*, Vol. 55, pp.1051–1070.

Chen, S., Billings, S.A. and Luo, W. (1989) 'Orthogonal least squares methods and their application to nonlinear system identification', *Int. J. Control*, Vol. 50, No. 5, pp.1873–1896.

Chen, S., Billings, S.A., Cowan, C.F.N. and Grant, P.M. (1990a) 'Non-linear systems identification using radial basis functions', *Int. J. Systems Sci.*, Vol. 21, pp.2513–2539.

Chen, S., Billings, S.A., Cowan, C.F.N. and Grant, P.M. (1990b) 'Practical identification of NARMAX models using radial basis functions', *Int. J. Control*, Vol. 52, pp.1327–1350.

Chen, S., Cowan, C.F.N. and Grant, P.M. (1991) 'Orthogonal least squares learning algorithm for radial basis function networks', *IEEE Trans. Neural Networks*, Vol. 2, No. 2, pp.302–309.

Chen, S., Cowan, C.F.N., Billings, S.A. and Grant, P.M. (1990c) 'Parallel recursive prediction error algorithm for training layered neural networks', *Int. J. Control*, Vol. 51, pp.1215–1228.

Chen, S., Hong X. and Harris, C.J. (2003) 'Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design', *IEEE Trans. Automatic Control*, Vol. 48, No. 6, pp.1029–1036.

Chen, S., Hong X., Harris, C.J. and Sharkey, P.M. (2004) 'Sparse modelling using orthogonal forward regression with PRESS statistic and regularization', *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol. 34, No. 2, pp.898–911.

Chen, S., Wang, X.X. and Harris, C.J. (2005) 'Experiments with repeating weighted boosting search for optimization in signal processing applications', *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol. 35, No. 4, pp.682–693.

Feng, H-M. (2006) 'Self-generation RBFNs using evolutional PSO learning', *Neurocomputing*, Vol. 70, Nos. 1–3, pp.241–251.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, Reading, MA.

Gonzalez, J., Rojas, I., Ortega, J., Pomares, H., Fernandez, F.J. and Diaz, A.F. (2003) 'Multiobjective evolutionary optimization of the size, shape and position parameters of radial basis function networks for function approximation', *IEEE Trans. Neural Networks*, Vol. 14, No. 6, pp.1478–1495.

Guerra, F.A. and Coelho, L.S. (2008) 'Multi-step ahead nonlinear identification of Lorenz's chaotic system using radial basis function neural network with learning by clustering and particle swarm optimization', *Chaos, Solitons and Fractals*, Vol. 35, No. 5, pp.967–979.

Gunn, S. (1998) 'Support vector machines for classification and regression', Technical Report, ISIS Research Group, Department of Electronics and Computer Science, University of Southampton, UK, p.11.

Guru, S.M., Halgamuge, S.K. and Fernando, S. (2005) 'Particle swarm optimisers for cluster formation in wireless sensor networks', *Proc. 2005 Int. Conf. Intelligent Sensors, Sensor Networks and Information Processing* (Melbourne, Australia), 5–8 December, 2005, pp.319–324.

Ingber, L. (1993) 'Simulated annealing: practice versus theory', *Mathematical and Computer Modeling*, Vol. 18, No. 11, pp.29–57.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', *Proc. 1995 IEEE Int. Conf. Neural Networks* (Perth, Australia), 27 November–1 December 1995, Vol. 4, pp.1942–1948.

Kennedy, J. and Eberhart, R. (2001) *Swarm Intelligence*, Morgan Kaufmann.

Mak, M-W. and Kung, S-Y. (2000) 'Estimation of elliptical basis function parameters by the EM algorithm with application to speaker verification', *IEEE Trans. Neural Networks*, Vol. 11, No. 4, pp.961–969.

Man, K.F., Tang, K.S. and Kwong, S. (1998) *Genetic Algorithms: Concepts and Design*, Springer-Verlag, London.

McLoone, S., Brown, M.D., Irwin, G. and Lightbody, A. (1998) 'A hybrid linear/nonlinear training algorithm for feedforward neural networks', *IEEE Trans. Neural Networks*, Vol. 9, No. 4, pp.669–684.

Moody, J. and Darken, C.J. (1989) 'Fast learning in networks of locally-tuned processing units,' *Neural Computation*, Vol. 1, pp.281–294.

Myers, R.H. (1990) *Classical and Modern Regression with Applications*, 2nd ed., PWS-KENT, Boston.

Ratnaweera, A., Halgamuge, S.K. and Watson, H.C. (2004) 'Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients', *IEEE Trans. Evolutionary Computation*, Vol. 8, No. 3, pp.240–255.

Schölkopf, B. and Smola, A.J. (2002) *Learning with Kernels: Support Vector Machines*, *Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA.

Soo, K.K., Siu, Y.M., Chan, W.S., Yang, L. and Chen, R.S. (2007) 'Particle-swarm-optimization-based multiuser detector for CDMA communications', *IEEE Trans. Vehicular Technology*, Vol. 56, No. 5, pp.3006–3013.

Stone, M. (1974) 'Cross validation choice and assessment of statistical predictions,' *J. Royal Statistics Society Series B*, Vol. 36, pp.111–147.

Sun, J., Xu, W-B. and Liu, J. (2006) 'Training RBF neural network via quantum-behaved particle swarm optimization', *Proc. ICONIP 2006* (Hong Kong, China), 3–6 October 2006, pp.1156–1163.

Sun, T-Y., Liu, C-C., Tsai, T-Y. and Hsieh, S-T. (2008) 'Adequate determination of a band of wavelet threshold for noise cancellation using particle swarm optimization', *Proc. CEC 2008* (Hong Kong, China), 1–6 June 2008, pp.1168–1175.

Tipping, M.E. (2001) 'Sparse Bayesian learning and the relevance vector machine', *J. Machine Learning Research*, Vol. 1, pp.211–244.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

Wang, X.X., Chen, S., Lowe, D. and Harris, C.J. (2006a) 'Parsimonious least squares support vector regression using orthogonal forward selection with the generalised kernel model', *Int. J. Modelling, Identification and Control*, Vol. 1, No. 4, pp.245–256.

Wang, X.X., Chen, S., Lowe, D. and Harris, C.J. (2006b) 'Sparse support vector regression based on orthogonal forward selection for the generalised kernel model', *Neurocomputing*, Vol. 70, Nos. 1–3, pp.462–474.

Yang, Z.R. and Chen, S. (1998) 'Robust maximum likelihood training of heteroscedastic probabilistic neural networks', *Neural Networks*, Vol. 11, pp.739–747.