

# A Tunable Radial Basis Function Model for Nonlinear System Identification Using Particle Swarm Optimisation

S. Chen, X. Hong, B.L. Luk and C.J. Harris

**Abstract**—A tunable radial basis function (RBF) network model is proposed for nonlinear system identification using particle swarm optimisation (PSO). At each stage of orthogonal forward regression (OFR) model construction, PSO optimises 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 computationally more efficient.

## I. INTRODUCTION

The radial basis function (RBF) network is a popular neural network architecture for nonlinear system identification [1]. In this application, the parameters of a RBF network, including nodes' centre vectors and variances or covariance matrices as well as connecting weights, can be estimated based on nonlinear optimisation using gradient-descent algorithms [2], [3], the expectation-maximisation algorithm [4], [5] or the population-based evolutionary algorithms [6], [7]. However, these nonlinear estimation methods are computationally expensive and, 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 [8]– [10]. 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 other means, such as cross validation.

A most popular approach for identifying RBF models is to formulate the problem as a linear regression by considering the training input data points as candidate RBF centres and to employ a common variance for every RBF node. A parsimonious RBF network can then be identified efficiently using the orthogonal least squares (OLS) algorithm [11]– [14]. Similarly, the support vector machine (SVM) and other sparse kernel modelling methods [15]– [17] also place the kernel centres to the training input data points and adopt a common kernel variance for every kernel. A sparse kernel model 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 SVM, two more hyperparameters must also be specified via cross validation. Our previous results [14] show that the local regularisation assisted OLS (LROLS) algorithm based on the leave-one-out (LOO) cross validation compares favourably with other kernel methods, in terms of model sparsity and generalisation performance as well as efficiency of model construction. This LROLS-LOO algorithm [14] offers a state-of-the-art construction method for fixed-node RBF models.

This paper considers the tunable RBF model, where each RBF node has a tunable centre vector and an adjustable diagonal covariance matrix. We do not attempt to optimise all the RBF parameters together, as it is a too large and complex nonlinear 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). Specifically, we use particle swarm optimisation (PSO) [18], [19] to optimise one RBF node's centre vector and covariance matrix at each stage of the OFR. PSO is a population based stochastic optimisation technique [18], [19] inspired by social behaviour of bird flocking or fish schools. It is becoming popular due to its simplicity in implementation, ability to quickly converge to a reasonably good solution and its robustness against local minima. The method has been applied to many optimisation problems successfully [7], [19]– [25]. 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, compared with the state-of-the-art LROLS-LOO construction algorithm for fixed-node RBF models [14].

## II. IDENTIFICATION USING TUNABLE RBF MODELS

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

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

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

$$\begin{aligned} \mathbf{x}_k &= [x_{1,k} \ x_{2,k} \ \dots \ x_{m,k}]^T \\ &= [y_{k-1} \ \dots \ y_{k-m_y} \ u_{k-1} \ \dots \ u_{k-m_u}]^T \end{aligned} \quad (2)$$

S. Chen and C.J. Harris are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

X. Hong is with School of Systems Engineering, University of Reading, Reading RG6 6AY, UK

B.L. Luk is with Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, China

denotes the system input vector with the known dimension  $m = m_y + m_u$ . The NARX system (1) is a special case of the following generic NARMAX system [26]

$$y_k = f_s(y_{k-1}, \dots, y_{k-m_y}, u_{k-1}, \dots, u_{k-m_u}, e_{k-1}, \dots, e_{k-m_e}) + e_k. \quad (3)$$

The techniques developed for the NARX structure can be extended to the general NARMAX system [11], [26], [27].

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

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

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

$$\mathbf{p}_M^T(k) = [p_1(\mathbf{x}_k) p_2(\mathbf{x}_k) \dots p_M(\mathbf{x}_k)] \quad (5)$$

is the response vector of the  $M$  RBF nodes to the input  $\mathbf{x}_k$ . We consider the general RBF regressor of the form

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

where  $\boldsymbol{\mu}_i$  and  $\boldsymbol{\Sigma}_i = \text{diag}\{\sigma_{i,1}^2, \sigma_{i,2}^2, \dots, \sigma_{i,m}\}$  are the centre vector and diagonal covariance matrix of the  $i$ th 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  $k$ th training data point  $(\mathbf{x}_k, y_k)$  as

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

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

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

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

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

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

#### A. Orthogonal Decomposition

Let an orthogonal decomposition of  $\mathbf{P}_M$  be  $\mathbf{P}_M = \mathbf{W}_M \mathbf{A}_M$ , where  $\mathbf{W}_M = [\mathbf{w}_1 \mathbf{w}_2 \dots \mathbf{w}_M]$  with the orthogonal columns that satisfy  $\mathbf{w}_i^T \mathbf{w}_l = 0$  for  $l \neq i$  and

$$\mathbf{A}_M = \begin{bmatrix} 1 & \alpha_{1,2} & \dots & \alpha_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha_{M-1,M} \\ 0 & \dots & 0 & 1 \end{bmatrix}. \quad (10)$$

The regression model (8) can alternatively be presented as

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

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

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

where  $\mathbf{w}_M^T(k) = [w_1(k) w_2(k) \dots w_M(k)]$  is the  $k$ th row of  $\mathbf{W}_M$ . Orthogonal decomposition can be carried out using the Gram-Schmidt procedure [11]. Using the model (11) instead of the original one (8) facilitates an efficient OFR model construction. 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 training MSE [14].

#### B. OFR Based on LOO Cross Validation

The evaluation of model generalisation capability is directly based on the concept of cross validation [28], and a commonly used cross validation is the LOO cross validation with its associated LOO test MSE [29]. Consider the OFR modelling process that has produced the  $n$ -node RBF model. Denote the constructed  $n$  columns of regressors as  $\mathbf{W}_n = [\mathbf{w}_1 \mathbf{w}_2 \dots \mathbf{w}_n]$ , the  $k$ th model output of this  $n$ -node RBF model identified using the entire training data set  $D_K$  as

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

and the corresponding  $k$ th modelling error as  $\varepsilon_k^{(n)} = y_k - \hat{y}_k^{(n)}$ . If we “remove” the  $k$ th 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 [29]

$$\varepsilon_k^{(n,-k)} = \varepsilon_k^{(n)} / \eta_k^{(n)}, \quad (14)$$

where  $\eta_k^{(n)}$  is referred to as the LOO error weighting [29]. The LOO MSE for the  $n$ -node RBF model is defined as

$$J_n = \frac{1}{K} \sum_{k=1}^K \left( \varepsilon_k^{(n,-k)} \right)^2. \quad (15)$$

The LOO MSE  $J_n$  is a measure of the model generalisation capability [28], [29]. For the model (11),  $J_n$  can be computed very efficiently because  $\varepsilon_k^{(n)}$  and  $\eta_k^{(n)}$  can be calculated recursively using [14]

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

and

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

respectively, where  $\lambda \geq 0$  is a small regularisation parameter [14]. 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  $n$ th stage of the construction, the  $n$ th RBF node is determined by minimising  $J_n$  with respect to  $\boldsymbol{\mu}_n$  and  $\boldsymbol{\Sigma}_n$

$$\min_{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n} J_n(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n). \quad (18)$$

In the next section, we will detail how to use PSO to perform this optimisation. The LOO MSE  $J_n$  is locally convex with respect to the model size  $n$  [14]. Thus, there exists an ‘‘optimal’’ number of RBF nodes  $M$  such that: for  $n \leq M$   $J_n$  decreases as the model size  $n$  increases while

$$J_M \leq J_{M+1}. \quad (19)$$

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 [14] 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.

### III. PARTICLE SWARM OPTIMISATION AIDED OFR

The task at the  $n$ th stage of the OFR for constructing a tunable RBF network is to solve the optimisation problem (18). This optimisation problem is non-convex with respect to  $\boldsymbol{\mu}_n$  and  $\boldsymbol{\Sigma}_n$ , and we adopt PSO [18], [19] to determine  $\boldsymbol{\mu}_n$  and  $\boldsymbol{\Sigma}_n$ . Our study demonstrates that PSO is particularly suited for the optimisation task (18).

#### A. Particle Swarm Optimisation

In a PSO algorithm [19], 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,  $\mathbf{pb}$ , visited by each particle provides the particle the so-called *cognitive information*, while the best position visited so far among the entire group,  $\mathbf{gb}$ , offers the *social information*. The  $\mathbf{pb}$ s and  $\mathbf{gb}$  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

$$\mathbf{v}_i^{[l+1]} = \xi * \mathbf{v}_i^{[l]} + rand() * c_1 * (\mathbf{pb}_i^{[l]} - \mathbf{u}_i^{[l]}) + rand() * c_2 * (\mathbf{gb}^{[l]} - \mathbf{u}_i^{[l]}), \quad (20)$$

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

$l$	iteration index, $1 \leq l \leq L$ , $L$ is the maximum number of iterations
$i$	particle index, $1 \leq i \leq S$ , $S$ is the particle size
$\mathbf{v}_i^{[l]}$	velocity of $i$ th particle at $l$ th iteration. The $j$ th elements of $\mathbf{v}_i^{[l]}$ are in range $[-V_{j_{\max}}, V_{j_{\max}}]$
$\xi$	inertia weight
$c_j$	the acceleration coefficients, $j = 1, 2$
$rand()$	uniform random number between 0 and 1
$\mathbf{u}_i^{[l]}$	position of $i$ th particle at $l$ th iteration. The $j$ th elements of $\mathbf{u}_i^{[l]}$ are in range $[U_{j_{\min}}, U_{j_{\max}}]$
$\mathbf{pb}_i^{[l]}$	best position that the $i$ th particle has visited upto $l$ th iteration
$\mathbf{gb}^{[l]}$	best position that all the particles have visited upto $l$ th iteration

It is reported in [20] that using a time varying acceleration coefficient (TVAC) enhances 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 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

$$c_1 = 2.5 - \frac{2.0 * l}{1.0 * L} \quad \text{and} \quad c_2 = 0.5 + \frac{2.0 * l}{1.0 * L}, \quad (22)$$

respectively. In our experiment, we have found out that using a random inertia weight,  $\xi = rand()$ , achieves better performance than using  $\xi = 0$  or constant  $\xi$ . If the velocity in (20) approaches zero, it is reinitialised randomly to proportional to the maximum velocity

$$\mathbf{v}_i^{[l+1]}|_j = \pm rand() * \gamma * V_{j_{\max}}, \quad (23)$$

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

#### B. PSO Aided OFR for Tunable RBF Model

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

$$\begin{cases} U_{j_{\min}} = \min\{x_{j,k}, 1 \leq k \leq K\}, & 1 \leq j \leq m, \\ U_{j_{\max}} = \max\{x_{j,k}, 1 \leq k \leq K\}, \end{cases} \quad (24)$$

$$U_{j_{\min}} = \sigma_{\min}^2, \quad U_{j_{\max}} = \sigma_{\max}^2, \quad 1 + m \leq j \leq 2m. \quad (25)$$

The velocity bounds are defined by

$$V_{j_{\max}} = 0.5 * (U_{j_{\max}} - U_{j_{\min}}), \quad 1 \leq j \leq 2m. \quad (26)$$

Give the following initial conditions

$$\left. \begin{aligned} \varepsilon_k^{(0)} &= y_k \quad \text{and} \quad \eta_k^{(0)} = 1, \quad 1 \leq k \leq K, \\ J_0 &= \frac{1}{N} \mathbf{y}^T \mathbf{y} = \frac{1}{N} \sum_{k=1}^K y_k^2. \end{aligned} \right\} \quad (27)$$

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

**Initialisation.** Randomly generate the particles  $\mathbf{u}_i^{[0]}$ ,  $1 \leq i \leq S$ , in the search space defined by (24) and (25). Set the initial velocities  $\mathbf{v}_i^{[0]} = \mathbf{0}_{2m}$ ,  $1 \leq i \leq S$ , where  $\mathbf{0}_{2m}$  denotes the zero vector of dimension  $2m$ . Initialise  $J_n(\mathbf{g}\mathbf{b}^{[0]})$  and  $J_n(\mathbf{p}\mathbf{b}_i^{[0]})$ ,  $1 \leq i \leq S$ , to a value larger than  $J_0$ .

**Iteration loop.** For ( $l = 0$ ;  $l \leq L$ ;  $l++$ ) {

*Orthogonalisation and cost function evaluation.*

- 1) For  $1 \leq i \leq S$ , generate  $\mathbf{p}_n^i$  from  $\mathbf{u}_i^{[l]}$ , the candidates for the  $n$ th model column, according to (9), and orthogonalise them according to [11]

$$\alpha_{j,n}^i = \mathbf{w}_j^T \mathbf{p}_n^i / \mathbf{w}_j^T \mathbf{w}_j, \quad 1 \leq j < n, \quad (28)$$

$$\mathbf{w}_n^i = \mathbf{p}_n^i - \sum_{j=1}^{n-1} \alpha_{j,n}^i \mathbf{w}_j, \quad (29)$$

$$g_n^i = \left( \mathbf{w}_n^i \right)^T \mathbf{y} / \left( \left( \mathbf{w}_n^i \right)^T \mathbf{w}_n^i + \lambda \right). \quad (30)$$

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

$$\varepsilon_k^{(n)}(i) = \varepsilon_k^{(n-1)} - w_n^i(k) g_n^i, \quad 1 \leq k \leq K, \quad (31)$$

$$\eta_k^{(n)}(i) = \eta_k^{(n-1)} - \frac{\left( w_n^i(k) \right)^2}{\left( \mathbf{w}_n^i \right)^T \mathbf{w}_n^i + \lambda}, \quad 1 \leq k \leq K, \quad (32)$$

$$J_n^i = \frac{1}{K} \sum_{k=1}^K \left( \frac{\varepsilon_k^{(n)}(i)}{\eta_k^{(n)}(i)} \right)^2, \quad (33)$$

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

*Update cognitive and social information.*

- 1) For ( $i = 1$ ;  $i \leq S$ ;  $i++$ )

If ( $J_n^i < J_n(\mathbf{p}\mathbf{b}_i^{[l]})$ )

$J_n(\mathbf{p}\mathbf{b}_i^{[l]}) = J_n^i$ ;

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

End if;

End for;

- 2) Find

$$i^* = \arg \min_{1 \leq i \leq S} J_n(\mathbf{p}\mathbf{b}_i^{[l]}).$$

If ( $J_n(\mathbf{p}\mathbf{b}_{i^*}^{[l]}) < J_n(\mathbf{g}\mathbf{b}^{[l]})$ )

$J_n(\mathbf{g}\mathbf{b}^{[l]}) = J_n(\mathbf{p}\mathbf{b}_{i^*}^{[l]})$ ;

$\mathbf{g}\mathbf{b}^{[l]} = \mathbf{p}\mathbf{b}_{i^*}^{[l]}$ ;

End if;

*Update velocities and positions of particles*

- 1) For ( $i = 1$ ;  $i \leq S$ ;  $i++$ )

$$\mathbf{v}_i^{[l+1]} = \text{rand}() * \mathbf{v}_i^{[l]} + \text{rand}() * c_1 * (\mathbf{p}\mathbf{b}_i^{[l]} - \mathbf{u}_i^{[l]}) + \text{rand}() * c_2 * (\mathbf{g}\mathbf{b}^{[l]} - \mathbf{u}_i^{[l]});$$

For ( $j = 1$ ;  $j \leq 2m$ ;  $j++$ )

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

If ( $\text{rand}() < 0.5$ )

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

Else

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

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 ( $i = 1$ ;  $i \leq S$ ;  $i++$ )

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

For ( $j = 1$ ;  $j \leq 2m$ ;  $j++$ )

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  $\mathbf{u} = \mathbf{g}\mathbf{b}^{[L]}$ , i.e.  $\boldsymbol{\mu}_n$  and  $\boldsymbol{\Sigma}_n$  of the  $n$ th RBF node, the  $n$ th model column  $\mathbf{p}_n$ , the orthogonalisation coefficients  $\alpha_{j,n}$ ,  $1 \leq j < n$ , 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  $1 \leq k \leq K$ .

### C. Computational Complexity Comparison

We compare the computational complexity of the proposed PSO aided OFR algorithm for tunable RBF models with that of the LROLS-LOO algorithm for fixed-node RBF models [14]. 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 can be verify that the computational complexity of one model column orthogonalisation and the associated LOO cost function evaluation is  $\mathcal{O}(K)$ . 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

$$C_{\text{LROLS-LOO}} \approx \sum_{i=1}^{M'+1} (K - (i-1)) \approx (M'+1) \times K, \quad (34)$$

where the second approximation arrives because  $M' \ll K$ .

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

TABLE I

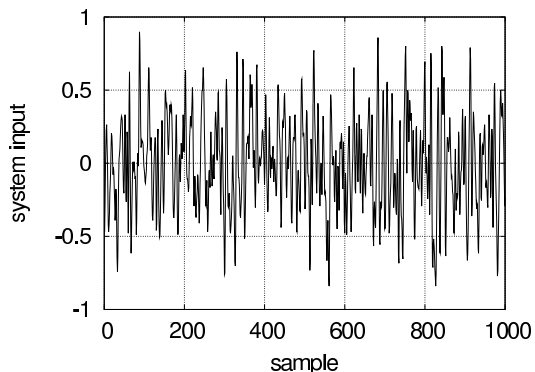
COMPARISON OF THE TWO GAUSSIAN RBF NETWORK MODELS OBTAINED BY THE LROLS-LOO AND PSO-OFR ALGORITHMS.

Algorithm	RBF type	Model size	Training MSE	Test MSE	complexity
LROLS-LOO	fixed	30	0.001400	0.002532	$15500 \times \mathcal{O}(500)$
PSO-OFR	tunable	20	0.001461	0.002463	$4200 \times \mathcal{O}(500)$

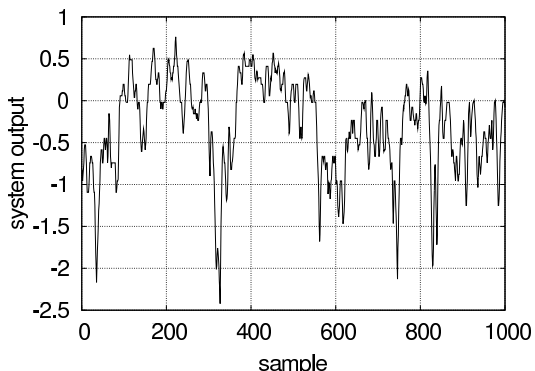
This number is given as

$$C_{\text{PSO-OFR}} \approx (M + 1) \times S \times L, \quad (35)$$

where  $M$  is the constructed model size,  $S$  the particle size and  $L$  the number of iterations. Since the model size  $M$  is usually much smaller than the model size  $M'$  obtained by the LROLS-LOO algorithm, we always have  $C_{\text{PSO-OFR}} < C_{\text{LROLS-LOO}}$  whenever  $K \geq S \times L$ . Thus, the PSO aided OFR algorithm for constructing tunable-node RBF models has clearly computational advantages over the LROLS-LOO algorithm for selecting fixed-node RBF models when the size of the training data set is large. Note that the PSO algorithm is very efficient. Our experimental results have shown that typically  $L = 20$  and  $S = 10$  to  $20$  are often sufficient. Furthermore, the complexity of (35) is the true complexity of the PSO aided OFR algorithm, while the complexity of (34) 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 based cross validation. Taking this fact into account, computational advantages of the proposed PSO aided OFR algorithm becomes even more significant.



(a)



(b)

Fig. 1. Liquid level data: (a) system input  $u_k$ , and (b) system output  $y_k$ .

#### IV. LIQUID LEVEL DATA SET MODELLING

The data set was collected from a nonlinear 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 [30]. Fig. 1 shows the 1000 data points of the data set used in this experiment. From the data set, 1000 data points  $\{\mathbf{x}_k, y_k\}$  were constructed with  $\mathbf{x}_k$  given by

$$\mathbf{x}_k = [y_{k-1} \ y_{k-2} \ y_{k-3} \ u_{k-1} \ u_{k-2} \ u_{k-3} \ u_{k-4}]^T. \quad (36)$$

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

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 [14]. With  $\sigma^2 = 2.0$ , the LROLS-LOO algorithm automatically selected a model set of  $M' = 30$  nodes from the candidate set of  $K = 500$  potential nodes. The results obtained by the LROLS-LOO algorithm are given in Table I, where the complexity was computed as  $C_{\text{LROLS-LOO}} = 31 \times 500 = 15500$  for the given  $\sigma^2 = 2.0$ .

For constructing the tunable-node RBF model, we set the particle size to  $S = 10$  and the number of iterations to  $L = 20$ . The PSO aided OFR algorithm automatically constructed a model set of  $M = 20$  nodes. The results produced by the the PSO aided OFR are also listed in Table I, where the complexity was given by  $C_{\text{PSO-OFR}} = 21 \times 10 \times 20 = 4200$ . Fig. 2 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.

#### V. CONCLUSIONS

In this contribution we have proposed a novel PSO aided OFR algorithm to construct tunable-node RBF network models for nonlinear 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 fixed-node RBF models, the LROLS-LOO algorithm, as

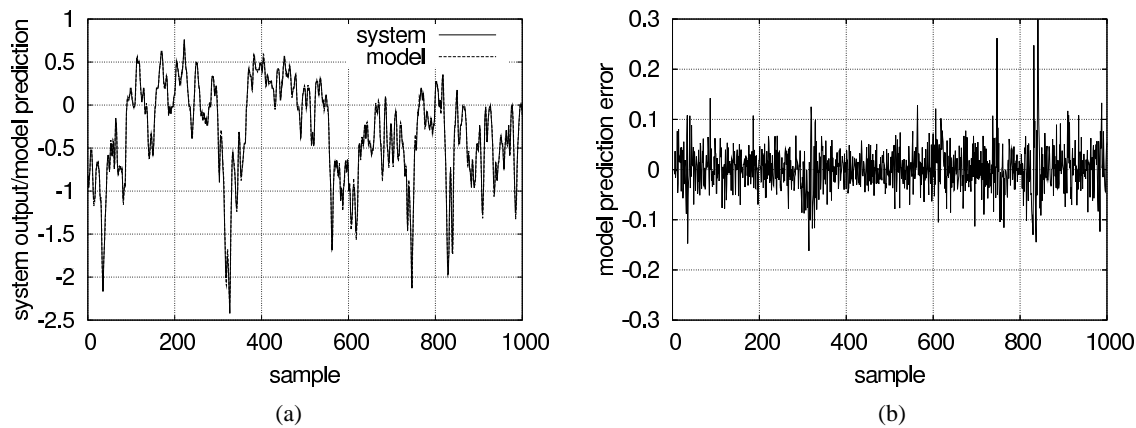


Fig. 2. Modelling of the liquid level data by the 20-node RBF network constructed using the PSO aided OFR: (a) model prediction  $\hat{y}_k$  superimposed on system output  $y_k$ , and (b) model prediction error  $\varepsilon_k = y_k - \hat{y}_k$ .

the benchmark, it has been shown 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.

#### REFERENCES

- [1] S. Chen, S.A. Billings, C.F.N. Cowan and P.M. Grant, "Non-linear systems identification using radial basis functions," *Int. J. Systems Sci.*, Vol.21, pp.2513–2539, 1990.
- [2] S. Chen, C.F.N. Cowan, S.A. Billings and P.M. Grant, "Parallel recursive prediction error algorithm for training layered neural networks," *Int. J. Control*, Vol.51, pp.1215–1228, 1990.
- [3] S. McLoone, M.D. Brown, G. Irwin and A. Lightbody, "A hybrid linear/nonlinear training algorithm for feedforward neural networks," *IEEE Trans. Neural Networks*, Vol.9, No.4, pp.669–684, 1998.
- [4] Z.R. Yang and S. Chen, "Robust maximum likelihood training of heteroscedastic probabilistic neural networks," *Neural Networks*, Vol.11, pp.739–747, 1998.
- [5] M.-W. Mak and S.-Y. Kung, "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, 2000.
- [6] J. Gonzalez, I. Rojas, J. Ortega, H. Pomares, F.J. Fernandez and A.F. Diaz, "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, 2003.
- [7] H.-M. Feng, "Self-generation RBFNs using evolutionary PSO learning," *Neurocomputing*, Vol.70, No.1-3, pp.241–251, 2006.
- [8] J. Moody and C.J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, Vol.1, pp.281–294, 1989.
- [9] S. Chen, S.A. Billings and P.M. Grant, "Recursive hybrid algorithm for non-linear system identification using radial basis function networks," *Int. J. Control*, Vol.55, pp.1051–1070, 1992.
- [10] S. Chen, "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, 1995.
- [11] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, Vol.50, No.5, pp.1873–1896, 1989.
- [12] S. Chen, C.F.N. Cowan and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.2, No.2, pp.302–309, 1991.
- [13] S. Chen, X. Hong and C.J. Harris, "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, 2003.
- [14] S. Chen, X. Hong, C.J. Harris and P.M. Sharkey, "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, 2004.
- [15] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [16] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Research*, Vol.1, pp.211–244, 2001.
- [17] B. Schölkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
- [18] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. 1995 IEEE Int. Conf. Neural Networks* (Perth, Australia), Nov.27–Dec.1, 1995, Vol.4, pp.1942–1948.
- [19] J. Kennedy and R. Eberhart, *Swarm Intelligence*. Morgan Kaufmann, 2001.
- [20] A. Ratnaweera, S.K. Halgamuge and H.C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evolutionary Computation*, Vol.8, No.3, pp.240–255, 2004.
- [21] S.M. Guru, S.K. Halgamuge and S. Fernando, "Particle swarm optimisers for cluster formation in wireless sensor networks," in *Proc. 2005 Int. Conf. Intelligent Sensors, Sensor Networks and Information Processing* (Melbourne, Australia), Dec.5–8, 2005, pp.319–324.
- [22] K.K. Soo, Y.M. Siu, W.S. Chan, L. Yang and R.S. Chen, "Particle-swarm-optimization-based multiuser detector for CDMA communications," *IEEE Trans. Vehicular Technology*, Vol.56, No.5, pp.3006–3013, 2007.
- [23] T.-Y. Sun, C.-C. Liu, T.-Y. Tsai and S.-T. Hsieh, "Adequate determination of a band of wavelet threshold for noise cancellation using particle swarm optimization," in *Proc. CEC 2008* (Hong Kong, China), June 1–6, 2008, pp.1168–1175.
- [24] F.A. Guerra and L.S. Coelho, "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, 2008.
- [25] W.-F. Leong and G.G. Yen, "PSO-based multiobjective optimization with dynamic population size and adaptive local archives," *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol.38, No.5, pp.1270–1293, 2008.
- [26] S. Chen and S.A. Billings, "Representation of non-linear systems: The NARMAX model," *Int. J. Control*, Vol.49, No.3, pp.1013–1032, 1989.
- [27] S. Chen, S.A. Billings, C.F.N. Cowan and P.M. Grant, "Practical identification of NARMAX models using radial basis functions," *Int. J. Control*, Vol.52, pp.1327–1350, 1990.
- [28] M. Stone, "Cross validation choice and assessment of statistical predictions," *J. Royal Statistics Society Series B*, Vol.36, pp.111–147, 1974.
- [29] R.H. Myers, *Classical and Modern Regression with Applications*. 2nd Edition, Boston: PWS-KENT, 1990.
- [30] S.A. Billings and W.S.F. Voon, "A prediction-error and stepwise-regression estimation algorithm for non-linear systems," *Int. J. Control*, Vol.44, pp.803–822, 1986.