Adaptive Modelling With Tunable RBF Network Using Multi-innovation RLS Algorithm Assisted By Swarm Intelligence

Hao Chen, Yu Gong and Xia Hong

School of Systems Engineering, University of Reading, Reading, RG6 6AY, UK
E-mail: hao.chen@pgr.reading.ac.uk, {y.gong, x.hong}@reading.ac.uk

Abstract—In this paper, we propose a new on-line learning algorithm for the non-linear system identification: the swarm intelligence aided multi-innovation recursive least squares (SI-MRLS) algorithm. The SI-MRLS algorithm applies the particle swarm optimization (PSO) to construct a flexible radial basis function (RBF) model so that both the model structure and output weights can be adapted. By replacing an insignificant RBF node with a new one based on the increment of error variance criterion at every iteration, the model remains at a limited size. The multi-innovation RLS algorithm is used to update the RBF output weights which are known to have better accuracy than the classic RLS. The proposed method can produces a parsimonious model with good performance. Simulation result are also shown classic RLS. The proposed method can produces a parsimonious output weights which are known to have better accuracy than the classic RLS. The proposed method can produces a parsimonious model with good performance. Simulation result are also shown to verify the SI-MRLS algorithm.

I. INTRODUCTION

Radial basis function (RBF) network as a linear-in-the-parameter model has been widely used in non-linear system identification [1]. A typical RBF model consists of a batch of non-linear nodes and an output layer producing a weighted summation of the nodes’ outputs. While the output layer weights can be estimated by on-line learning algorithms such as the least mean square (LMS) [2] and recursive least square (RLS) [3], the performance of the RBF network depends very much on the network size and structure parameters of the inner nodes (e.g. centre vectors and variances for Gaussian nodes). The RBF node selection/optimization becomes necessary especially for non-stationary systems. This is generally a complicated optimization problem.

The orthogonal least squares (OLS) algorithm and its variants (e.g. ROLS, LROLS, LROLS-LOO and PSO-OFR) [4]-[7] are the most popular approaches for the RBF node selection. While the state-of-art OLS methods can achieve parsimonious RBF models with efficient computation and good generalisation performance, they are mostly for off-line approaches which use entire training data for the model construction and not ideal for non-stationary systems. On-line identification (sequential learning) approaches that can update both the weight coefficients and model structure parameters are of great interest.

The resource allocating network (RAN) is one of the earliest on-line system identification approaches with adapting structure [8]. It starts from an empty node and adds Gaussian RBF nodes every time the new data sample is detected as “novel”. If the new data is not “novel”, only the weight coefficients are updated by the LMS algorithm. As an enhancement of the RAN, the RAN extended Kalman filter (RANEKF) interprets the RAN from a function space approach and replaces the LMS by the extended Kalman filter to adjust the network parameters [9]. Both RAN and RANEKF algorithms are growing network approaches that the network size is always increased and the redundant RBF nodes are not removed. Molina improves the algorithm by limiting the size of the RBF network [10]. The minimal RAN (M-RAN) algorithm modifies the RANEKF algorithm with a pruning procedure based on relative contribution [11], [12]. It can obtain a more parsimonious model, but requires many controlling parameters. The growing and pruning RBF (GAP-RBF) and Generalized GAP-RBF (GGAP-RBF) algorithms were proposed in [13] and [14] respectively. The growing criteria is based on the “significance” of a specific neuron that is determined by its average contribution to the network performance over all of the data it receives so far. It has direct link to the learning accuracy, but requires the input sampling range or distribution information as a prior knowledge. It is worth noting that while all of the above on-line learning algorithms can grow and/or pruning the RBF network nodes, the model generalisation is not guaranteed. For many algorithms, the network size can continuously grow to a huge number for large non-stationary systems. In this paper, we propose the swarm intelligence aided multi-innovation recursive least squares (SI-MRLS) for the on-line non-linear system identification. The proposed tunable RBF model starts with a structure with a preset number of nodes. When the model exists inactive nodes, the node with least “significant” to the overall performance is replaced by a new node. On the other hand, if the model has no inactive nodes, only the weights are adjusted by the MRLS. The least significant node is determined by the increment of error variance (IEV) [15]. If a new node is used to replace an old one, the particle swarm optimization (PSO) [16] is used to find the best center vector and variance of the new node. The PSO algorithm can quickly converge to a global optimize solution with affordable computation as only one node needs to be
optimized at a time. The RBF output layer weights are updated by the MRRLS algorithm [17]. As a generalization of the classic RLS algorithm, the MRLS algorithm uses current error as well as past errors to update the coefficients so that the estimation accuracy can be much improved.

The tunable RBF model with the proposed SI-MRLS algorithm can generate a sparse model with good generalization performance. Numerical simulation are given to test the proposed SI-MRLS algorithm, in which the SI-MRLS is compared with existing approaches for a times series prediction application. The results well verify the effectiveness of the SI-MRLS algorithm.

II. THE SI-MRLS ALGORITHM

A. RBF Regression Model with MRLS

Assuming the sampling data set as $S_N = \{x_t, y_t\}_{t=1}^N$, the input data matrix at time $t$ is given by

$$X_t = [x_t, x_{t-1}, \ldots, x_{t-p+1}]^T \in \mathbb{R}^{p \times N_x}$$

(1)

where $x_i = [x_i(1), x_i(2), \ldots, x_i(N_x)]^T \in \mathbb{R}^{1 \times N_x}$ is the $i$th input data vector, $p$ is the innovation length which determines the number of the past errors used for the adaptation, and $N_x$ is the input dimensions. When $p = 1$, the MRLS algorithm reduces to the classical RLS algorithm. Suppose there are $M$ RBF nodes as $g = [g_1, g_2, \ldots, g_M]$. The RBF information matrix is given by

$$\Phi_t = \begin{bmatrix} g_1(x_t) & g_2(x_t) & \cdots & g_M(x_t) \\ g_1(x_{t-1}) & g_2(x_{t-1}) & \cdots & g_M(x_{t-1}) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(x_{t-p+1}) & g_2(x_{t-p+1}) & \cdots & g_M(x_{t-p+1}) \end{bmatrix} \in \mathbb{R}^{p \times M}$$

(2)

$$= X_t g = [\theta_1, \theta_2, \ldots, \theta_M] = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_M^T \end{bmatrix} \in \mathbb{R}^{p \times M}$$

In this paper, we assume the RBF basis function as Gaussian so that $g_i(x_n) = \exp(-0.5 * (x_n - c_i)^T H_i (x_n - c_i))$, where $c_i = [c(1), c(2), \ldots, c(N_x)]^T$ and $H_i = \text{diag}(\sigma_{i1}^2, \sigma_{i2}^2, \ldots, \sigma_{iN_x}^2)$ which are the centre vector and diagonal covariance matrix of the $i$th RBF node respectively. A linear-in-the-parameters model can be formulated as

$$Y_t = \Phi_t w_t + E_t$$

(3)

where $w_t = [\omega_t(1), \omega_t(2), \ldots, \omega_t(M)]^T$ which is the weights vector, $\Phi_t w_t$ gives the RBF model output, $Y_t \in \mathbb{R}^{p \times N_y}$ is the system output ($N_y$ is the output dimensions, we just consider the case $N_y = 1$ in this paper) which is given by

$$Y_t = [y_t, y_{t-1}, \ldots, y_{t-p+1}]^T \in \mathbb{R}^{p \times 1}$$

(4)

and $E$ is the residual vector as

$$E = \begin{bmatrix} y_t - \Phi_t^T w_{t-1} \\ y_{t-1} - \Phi_t^T w_{t-1} \\ \vdots \\ y_{t-p+1} - \Phi_t^T w_{t-1} \end{bmatrix} \in \mathbb{R}^{p \times 1}$$

(5)

B. Determining the least significant node

In order for the RBF structure to adapt to the changing scenario, we propose to replace the node which has least contribution to the overall performance with a new node. First, (3) can be re-written as

$$Y_t = \Phi_{t-i} w_{t-i} + \omega_t(i) \theta_i$$

(6)

where $\Phi_{t-i}$ is derived from $\Phi_t$ by removing the $i$th column $\theta_i$, $w_{t-i}$ is $w_t$ with $i$th element being removed.

In order to separate the contribution from the $i$th node, $\theta_i$ is made orthogonal to $\Phi_{t-i}$ with orthogonal projection operation as

$$\theta_{i,o} = \{ I - \Phi_{t-i} [\Phi_{t-i}^T \Phi_{t-i}]^{-1} \Phi_{t-i}^T \} \theta_i$$

(7)

From (7), (6) can be rewritten as

$$Y_t = \Phi_{t-i} \nu_{t-i} + \omega_t(i) \omega_{i,o}$$

(8)

where $\nu_{t-i} = w_{t-i} + [\Phi_{t-i}^T \Phi_{t-i}]^{-1} \Phi_{t-i}^T \omega_{i,o}$, and

$$\omega_{i,o} = \theta_{i,o}^T \frac{y_t - \Phi_t \omega_{t-1}}{\theta_t^T \theta_t} \theta_{i,o}$$

(9)

Therefore, the contribution from the $i$th RBF node to the overall performance is obtained as the increment of error variance (IEV) which is given by

$$\text{IEV}_i = \omega_{i,o}^2 \theta_{i,o}^T \theta_{i,o}$$

(10)

The node with the smallest IEV has the least contribution to the overall performance.

C. Swarm Intelligence

At every iteration, if a node is determined as “insignificant” to the overall performance, it is replaced by a new node. The centre and the variance of the new node are searched by the swarm intelligence approach as below.

1) Initialisation. IF ($l = 0$)

- Randomly generate the particles $\Upsilon_{k,l}$ with initial velocity $\nu_{k,l}$ in a bounded space.
- Initializes $J_l(p_{best_{k,l}})$ and $J_l(g_{best_l})$ to a big value, where $J_l(\bullet)$ is the cost function as

$$J_l = c_l^2 = (y_t - \Phi_t \omega_{t-1})^2$$

(11)

2) Iteration loop. FOR $l = 1 : L$

- update the cognitive information $p_{best_{k,l}}$
- update the social information $g_{best_l}$
- update the particle’s velocity $\nu_{k,l+1}$ based on $p_{best_{k,l}}$ and $g_{best_l}$
update the particle’s position $Y_{k,l+1} = Y_{k,l} + \nu_{k,l+1}$
when $l = L$, yields the final solution $\bar{Y} = g_{bestL}$.

End of loop

D. Algorithm Summary

The proposed SI-MRLS algorithm describes a new RBF model with tunable structure for the on-line learning. The algorithm is shown as following.

SI-MRLS algorithm

Set the innovation length as $p$, RBF nodes number as $M$ and SI particles (including RBF center and variance) as $\Sigma$. At every time $t$ that new sampling data $(x_t, y_t)$ arrives, follow the below steps:

1) Update data matrix $X_t, Y_t$, the information matrix $\Phi_t$ in (1), (4) and (2).

2) Adjust model.
   Find the least meaningful node $\theta_s, s \in \{1, 2, \ldots, M\}$ based on the IEV criterion:
   \begin{equation}
   IEV_1(\theta_s) \leq IEV_2 \leq \ldots \leq IEV_M
   \end{equation}
   If $\frac{\sum_{i=1}^M IEV_i}{\sum_{i=1}^M IEV_i} < \Delta_1$ OR $\frac{IEV_3 - IEV_1}{IEV_1} > \Delta_2$
   Remove node $\theta_s$.
   Replace $\theta_s$ with a new node $\bar{\theta}$
   Use the PSO to find the best center and variance of the new node.
   Else Keep the current nodes unchanged.

3) Update RBF output layer weights $w_t$ by the MRLS.

Note that node replacement does not occur at every sampling time, as otherwise it demands too much computation cost. A criterion is set up as above to indicate when the node should be replaced. The flowchart of the SI-MRLS algorithm is shown in Fig. 1.

III. Numerical Simulation

In this section, the proposed SI-MRLS algorithm is tested for the time series prediction based on a set of real financial data of the monthly return value for the IBM common stock from January 1961 to December 1967. To be specific, the time series $y_N = \{y_n\}_{n=1}^N$ consists of $N = 84$ data points. The simulation is to predict one step ahead monthly returns $y_n$ using the past three data points: $x_n = [y_{n-1}, y_{n-2}, y_{n-3}]^T$. Thus the $n^{th}$ observation is given by

$s_n = \{x_n, y_n\} = \{[y_{n-1}, y_{n-2}, y_{n-3}]^T, y_n\}$

In the simulation, the MRLS learning algorithm is used to adapt the weight vector, where $p = 3$ so that three past residues are used, the covariance matrix is initialized as $P = 10^6I$, and the forgotten factor is chosen as $\lambda = 0.99$. There are 4 RBF nodes. At every model adjusting step, we use 10 swarm particles running for $L = 15$ iteration which we found is sufficient to achieved a desired accuracy. The parameters for model adjust criterion are set as $\Delta_1 = 0.1$ and $\Delta_2 = 0.5$ respectively.

Fig. 2 shows the model output $y_n$ and model prediction $\hat{y}_n$ of the SI-MRLS algorithm. It is clearly shown that the SI-MRLS can well track the time series variation. Fig. 3 show the root mean square error (RMSE) which is defined as

\begin{equation}
RMSE = \sqrt{\frac{\sum_{i=1}^n [y_i - f(x_i)]^2}{n}}
\end{equation}

The RMSE is often used as a performance index to show the predication accuracy. For comparison, the RMSE for the following approaches are also shown: the MRLS (the linear model without any RBF nodes), the classic RBF model with 20 fixed nodes and MRLS for weight adaptation, and the more recently proposed GAP-RBF algorithm which can be used as a bench mark for the RBF with node selection approaches [13], [14]. The parameters for the linear MRLS approach are as same as those for the SI-MRLS. For the fixed RBF model, a common variance $\sigma^2 = 1.0$ is used for all RBF nodes and the centre vector is randomly generated from the data set. For the GAP-RBF, the expected approximation accuracy and input sampling range are selected as $e_{min} = 10^{-6}$ and $S(X) = 0.2$.
respectively. The GAP-RBF starts with an empty node, and after 84 sampling time it reaches a 7-nodes RBF model.

It is clearly shown in Fig. 3 that the linear MRLS approach has the worst RMSE performance since it does not explore the non-linearity of the data at all. It is interesting to observe that the RBF mode with fixed node only slightly better than the MRLS approach, while the GAP-RBF with adaptive structure can obviously further improve the performance. This clearly indicates that the structure of the RBF should be adapted especially for non-stationary data. The proposed SI-MRLS algorithm performs significantly better than the GAP-RBF algorithm, making it an attractive approach in practice.

IV. Conclusion

In this paper, we propose the on-line SI-MRLS algorithm for the RBF network. Based on the swarm intelligence, the SI-MRLS algorithm adjusts both the RBF weight vector and structure parameters, while keeping the network size limited. The proposed approach is compared to the linear MRLS, RBF with fix nodes and the GAP-RBF algorithm with node adaptation. Numerical results show that the SI-MRLS outperforms the existing approaches.

Acknowledgement

This research is sponsored by the UK Engineering and Physical Sciences Research Council and DSTL under the grant number EP/H012516/1.

References