UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

Heuristically optimized RBF neural model for the control of section weights in stretch blow moulding

Jing Deng*, Ziqi Yang†, Kang Li†, Gary Menary*, Eileen Harkin-Jones *
*School of Mechanical & Aerospace Engineering
Queen's University Belfast, Belfast, BT9 5AH, UK
Email: {j.deng, G.Menary, e.harkinjones}@qub.ac.uk
†School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast, Belfast, BT9 5AH, UK
Email: {zyang06, k.li}@qub.ac.uk

Abstract—**The injection stretch-blow Moulding (ISBM) process is typically used to manufacture PET containers for the beverage and consumer goods industry. The process is somehow complex and users often have to heavily rely on trial and error methods to setup and control it. In this paper, a novel identification method based on a radial basis function (RBF) network model and heuristic optimization methods, such as particle swarm optimization (PSO), deferential evolution (DE), and extreme learning machine (ELM) is proposed for the modelling and control of bottle section weights. The main advantage of the proposed method is that the non-linear parameters are optimized in a continuous space while the hidden nodes are selected one by one in a discrete space using a two-stage selection algorithm. The computational complexity is significantly reduced due to a recursive updating mechanism. Experimental results on simulation data from ABAQUS are presented to confirm the superiority of the proposed method.**

I. INTRODUCTION

Over last few decades, the use of plastics has experienced healthy growth due to its many high performances including lightweight with high tensile/impact/tear strengths, high temperature/chemical resistance, high clarity/modulus/plasticity and low cost [1]. The injection stretch-blow moulding (ISBM) process is a kind of blow-moulding process in making thin-walled polyethylene terephthalate (PET) bottles for the carbonated soft drink and mineral water industries. In such a process, polymer granules is first extruded and injected into a hollow tube to produce structurally amorphous preforms. The resultant preforms are then loaded and conveyed in an infrared oven for reheat. Finally, the heated preforms are simultaneously stretched by a rod and blown with high pressure air to produce to the finished article shape. This stretch blowing and subsequent cooling takes around one second. Due to the fast production rate, the ability to mould complex part and some other attractive features, ISBM process has become one of the most popular methods adopted in the polymer industry.

The quality of made bottles is usually indicated by the bottle wall thickness distribution, top load, burst, and the section weights. All these variables are correlated to the process parameter settings, including preform temperature distribution, blowing air pressure and the delay between rode stretching and air blowing [2]. Unfortunately, the current ISBM process is still an open-loop system where optimal process settings are found by trial and error [3]. This is not only time consuming but also leads to wasting of materials, and the process is sensitive to both internal and external interruptions. Therefore, the design of feedback control for ISBM process is extremely urgent. The bottle wall thickness and its section weights are clearly the best options for feedback signals, but they are either difficult to measure in real-time or the cost is too high. Alternatively, the 'soft-sensor' method to infer these parameters based on the mathematical process model becomes an affordable and cost-effective approach.

In non-linear system identification, the radial basis function networks are recognized as an universal approximation model that has been widely applied in data mining, pattern recognition, signal processing, and system modelling and control [4]. The RBF network has a simple topological structure, and it is easy to be trained compared to the multilayer perceptron (MLP) neural network alternative. The construction of RBF network model mainly involves two steps, the optimization of basis function parameters and the estimation of output layer weights. The latter can be easily achieved by least-square estimation while the former is difficult to implement as it involves non-linear optimization. The conventional way to handle those non-linear parameter are either by exhaustive search or gradient-based methods, which can be computationally expensive, and the global best solution cannot be guaranteed [5].

Heuristic approaches, such as simulated annealing (SA) [6], evolutionary algorithm (EA) [7], Tabu search (TS) [8], particle swarm optimization (PSO) [9], differential evolution (DE) [10], ant colony optimization (ACO) [11] and harmony search (HS) [12] have offered the alternatives for such an optimization problem. Unlike conventional calculus-based methods, heuristic approaches randomly generate some new solutions from which the best one is selected. The employment of heuristic methods in the RBF network construction can be implemented at either the global or the local level. In the former case all the non-linear and linear parameters are optimized simultaneously while in the latter case only the non-linear parameters of a single hidden node are regarded as one solution, and the RBF model is built step by step through subset selection algorithm [13]. Simultaneous optimization

978-1-4673-1558-6/12/$31.00 ©2012 IEEE

of the whole RBF network can archive the global optimal solution. However, it can be computationally expensive, and a large number of iterations are required in the optimization process. By contrast, subset selection involves fewer parameters to be optimized at each step, thus it is more efficient. The recently proposed two-stage subset selection algorithm has also shown its ability in reaching the near global optimal solution while retaining the computational efficiency of the forward alternative [14, 15].

In this paper, heuristic optimization methods, such as the Particle Swarm Optimization, Differential Evolution and the recently developed Extreme Learning Machine [16, 17] are effectively integrated with our two-stage selection (TSS) algorithm, leading to a flexible and efficient construction scheme for RBF neural modelling. Non-linear parameters in the hidden layer will be continuously optimized at each step while the output layer weights are estimated by the least-squares. The main advantages over the conventional subset selection methods are that both the centre vector and width vector in a RBF function are optimized separately for each hidden node. Thus the model size can be significantly reduced, leading to improved generalisation performance. The experimental results on the simulation data illustrate the compactness and performances of the obtained model. It also shows that the PSO and DE have similar capabilities in non-linear parameter optimization while the extreme learning machine provides an efficient alternative with significantly reduced computational effort.

The rest of this paper is organized as follows. Section II and III give an overview of the ISBM process and the heuristic methods employed. The new two-stage selection algorithm is then introduced in Section IV with the experimental results given in V. Finally, VI concludes the paper and provides with suggested future work.

II. The ISBM process

Practically, it is difficult to generate enough data for system identification. Thus a commercial finite element (FE) package known as ABAQUS is used to simulate the ISBM process. Several conditions need to be pre-set before the simulation process. These include the geometry of the preform, stretch rod and mould, material model, pre-blow process conditions (mass flow rate, pre-blow air pressure, preform temperature) and final-blow process conditions (final-blow air pressure and cooling time).

The pre-blow process conditions have a major influence on the final thickness and mechanical properties distributions in the ISBM process. At this stage, the simulations are only focused on modelling wall thickness distribution at different pre-blow process conditions. The preform reheating stage and cooling stage are beyond the scope of this study. Four parameters of the process were identified to be important, namely mass flow rate, pressure, temperature, and timing. The mass flow rate indicates the velocity of air blowing to the pre-form while the pressure affects the total amount of air consumed. The pre-form temperature was assumed uniform and equivalent to the setting value in the simulation, and the timing reflects the delay between stretching and blowing. Each of the four parameters were given different settings as shown in table I, and totally 144 possible combinations were obtained. Meanwhile the bottle wall volume is divided into three parts: shoulder, side-wall and base. These volume distribution indicators can generally represent the weight distribution. The frictional force between mould and bottle was infinite during the simulation.

TABLE I
DOE settings of each parameter in ISBM simulation

Mass flow rate(g/s)	Pressure(MPa)	Temperature(°C)	Timing(ms)
5	0.6	95	0
17	0.8	100	50
29	1	105	100
40		110	

III. Review of RBF network model and heuristic optimization methods

A. Radial Basis function network model

A general RBF neural model can be expressed as

$$y(t) = \sum_{k=1}^{n} \theta_k \varphi_k(\mathbf{x}(t); \mathbf{c}_k; \mathbf{\Sigma}_k) + \varepsilon(t) \tag{1}$$

where $y(t)$ is the actual output at sample time t, $\mathbf{x}(t) \in \Re^p$ is the input vector, $\varphi_k(\mathbf{x}(t); \mathbf{c}_k; \mathbf{\Sigma}_k)$ denotes the nonlinear activation function, $\mathbf{c}_i = [c_{i1}, c_{i2}, \cdots, c_{ip}]^T$ is the centre vector, and $\mathbf{\Sigma}_i$ is the associated norm matrix. Finally, θ_k represents the output layer weight for each RBF node, and $\varepsilon(t)$ is the network error at sample time t. By using a set of N data samples $\{\mathbf{x}(t), y(t)\}_{t=1}^{N}$ for model training, (1) can then be re-written in a matrix form as

$$\mathbf{y} = \mathbf{\Phi}\boldsymbol{\theta} + \mathbf{e} \tag{2}$$

If the regression matrix $\mathbf{\Phi}$ is of full column rank, the Least-Squares estimate of the regression coefficients in (2) is given by

$$\hat{\boldsymbol{\theta}} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y} \tag{3}$$

where $\mathbf{\Phi}^T\mathbf{\Phi}$ is sometimes called the information matrix. The associated minimal cost function is

$$J_n(\hat{\boldsymbol{\theta}}_n) = \mathbf{y}^T(\mathbf{I} - \mathbf{\Phi}_n)(\mathbf{\Phi}_n^T\mathbf{\Phi}_n)^{-1}\mathbf{\Phi}_n^T\mathbf{y} \tag{4}$$

B. Particle Swarm Optimization

PSO involves a number of particles which move through the problem search space seeking an optimal or satisfactory solution. The position of each particle is adjusted according to its velocity and the difference between its current position, the best one it has found so far, and the best position to date found by its neighbours.

Suppose $\mathbf{x}_i$ denotes the i^{th} particle in the swarm, $\mathbf{v}_i$ represents its velocity, $\mathbf{p}_i$ is its best position to date, while

$\mathbf{p}_g$ denotes the best position from the entire swarm. In inertia-weighted PSO, $\mathbf{v}_{(i+1)}$ and $\mathbf{x}_{(i+1)}$ are updated as:

$$\mathbf{v}_{(i+1)} \leftarrow w_0\mathbf{v}_i + c_1 r_1(\mathbf{p}_i - \mathbf{x}_i) + c_2 r_2(\mathbf{p}_g - \mathbf{x}_i) \quad (5)$$
$$\mathbf{x}_{(i+1)} \leftarrow \mathbf{x}_i + \mathbf{v}_i \quad (6)$$

where w_0 is the inertia weight used to scale the previous velocity term, c_1 and c_2 are acceleration coefficients, and r_1 and r_2 are two uniform random number generated between 0 and 1. The acceleration coefficients c_1 and c_2 can be fixed or varied during the iterative procedure. In order to ensure that each updated particle is still inside the search space, it is also necessary to define a value range, and check both the position and velocity for each particle at the end of an iteration.

C. Deferential Evolution

As a population based optimization technique, DE [10] also starts with some initial points which are randomly generated in the search space, and then pushes the populations toward the global optimum point through repeated operations of mutation, crossover and selection. New populations are obtained by adding the weighted difference of two vectors to a third one, where the vectors are mutually different random points from the last generation.

Suppose $\mathbf{x}_i^{(l)}$ $(i = 1, 2, \cdots, p)$ is the solution vector in generation l, the operations in the classic DE method can be summarised as follows:

- **Mutation**: A mutant vector is generated by:

$$\mathbf{v}_i^{(l+1)} = \mathbf{x}_{r1}^{(l)} + F(\mathbf{x}_{r2}^{(l)} - \mathbf{x}_{r3}^{(l)}) \quad (7)$$

  where $r1$, $r2$, $r3$ are random indices from $[1, 2, \cdots, p]$ and $F \in [0, 2]$ is a real constant which controls the amplification of the added differential variation. Larger values for F lead to higher diversity in new populations, while lower values cause faster convergence.
- **Crossover**: This operation is implemented to increase the diversity of the population. A trial vector is defined as

$$\mathbf{u}_i^{(l+1)} = [u_{i1}^{(l+1)}, u_{i2}^{(l+1)}, \cdots, u_{ip}^{(l+1)}]^T \quad (8)$$

  with elements given by

$$u_{ij}^{(l+1)} = \begin{cases} v_{ij}^{(l+1)} & rand_i(0,1) \leq C_r \ or \ i = b_r \\ x_{ij}^{(l)} & otherwise \end{cases} \quad (9)$$

  where p is the vector dimension, $C_r \in [0, 1]$ is the predefined crossover constant, $rand_i(0, 1)$ uniformly generates a scaler from $[0, 1]$ at the i^{th} evaluation, and b_r is a random index chosen from $[1, 2, \cdots, p]$ so that $\mathbf{u}_i^{(l+1)}$ contains at least one parameter from $\mathbf{v}_i^{(l+1)}$.
- **Selection**: The last step is to compare all the trial vectors $\mathbf{u}_i^{(l+1)}$ with the target ones $\mathbf{x}_i^{(l)}$ using a criterion, such as their contribution to a loss function, and then decide which one becomes a member of the next generation.

The above procedure continues until a pre-set number of iterations is reached or the desired accuracy is obtained.

D. Extreme Learning Machine

The Extreme Learning Machine (ELM) was first introduced for the training of Single-hidden Layer Feedforward neural Network (SLFN). It builds the SLFN model by randomly assigning non-linear parameters for each hidden node instead of iterative training. The target is then simply a linear combination of the hidden nodes, and the output layer weights can be easily estimated by Least-Squares. As a result, the learning speed in ELM can be several orders of magnitude faster than traditional learning. Using incremental methods, it has been proven that the ELM can be regarded as a universal approximator [18].

The SLFN has a similar structure to a RBF network. For a multi-input, multi-output (MIMO) system, it can be expressed as

$$\mathbf{y}(t) = \sum_{k=1}^{n} \boldsymbol{\theta}_k \varphi_k(\mathbf{w}_k \cdot \mathbf{x}(t) + b_k) \quad (10)$$

where $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_p(t)]$ and $\mathbf{y}(t) = [y_1(t), y_2(t), \cdots, y_m(t)]$ is the system input and output vector; $\mathbf{w}_k = [w_{k1}, w_{k2}, \cdots, w_{kp}]$ is the weight vector between the p inputs and the k^{th} hidden node; b_k is the threshold of the k^{th} hidden node; $(\cdot)$ denotes the inner product, and φ is the activation function. Finally $\boldsymbol{\theta}_k = [\theta_{k1}, \theta_{k2}, \cdots, \theta_{km}]$ is the output layer weight vector between the k^{th} hidden node and m outputs.

In ELM, the non-linear parameters $\mathbf{w}_k$ and b_k in (10) are assigned randomly, and it has been proven that the the required number of hidden nodes $n \leq N$ if the activation function φ is infinitely differentiable [16].

According to the above theorem, the ELM is also valid for the RBF network in (1). The construction process can be summarised in two steps:

1) Randomly assign the hidden nodes parameters, including the number of hidden nodes n, and non-linear parameters $\mathbf{c}_i$ and $\boldsymbol{\sigma}_i$ for $i = 1, 2, \cdots, n$;
2) Form the regression matrix $\mathbf{\Phi}$, and estimate the output layer weights using (3);

The main issue in using ELM is that the sparsity of the constructed model cannot be guaranteed due to its stochastic characteristics. Fortunately, subset selection methods can be easily integrated to solve this problem, and better generalization performance can be achieved.

IV. Two-stage selection based on Heuristic optimization

The two-stage selection algorithm includes a forward model construction stage and a backward model refinement stage. At its first stage, one RBF centre is selected and added to the model at each step. The significance of each centre is measured by its contribution to the cost function. This process continues until some pre-defined modelling criteria are met (such as Akaike's information criterion or a pre-defined number of hidden nodes is reached), the algorithm then moves to the second stage where the importance of previously selected centres are reviewed, and any insignificant ones are replaced. The computational efficiency involved in such process is achieved by defining a residual matrix $\mathbf{R}_k$ which can be updated recursively. Moreover, PSO and DE are used to find the best centre at each step while ELM is used to form the centre pool before the selection process.

A. First stage - forward selection

Refer to (4), a recursive matrix $\mathbf{M}_k$ and a residual matrix $\mathbf{R}_k$ are defined to simplify the computation.

$$\mathbf{M}_k \triangleq \mathbf{P}_k^T \mathbf{P}_k \qquad k = 1, \cdots, n \tag{11}$$

$$\mathbf{R}_k \triangleq I - \mathbf{P}_k \mathbf{M}_k^{-1} \mathbf{P}_k^T \qquad \mathbf{R}_0 \triangleq I \tag{12}$$

where $\mathbf{P}_k^T \in \Re^{N \times k}$ contains the first k columns of the regression matrix $\mathbf{\Phi}$ in (2).

By substituting (12) into (4), the cost function becomes

$$J(\mathbf{P}_k) = \mathbf{y}^T \mathbf{R}_k \mathbf{y} \tag{13}$$

At this forward stage, the RBF centres are optimized one at a time, and given by the best solution from the entire population of the heuristic methods after several iterations. Suppose at the k^{th} step, one more centre $\mathbf{p}_{k+1}$ is to be added. The net contribution of $\mathbf{p}_{k+1}$ to the cost function can then be calculated as [14]:

$$\Delta J_{k+1}(\mathbf{P}_k, \mathbf{p}_{k+1}) = \frac{(\mathbf{y}^T \mathbf{p}_{k+1}^{(k)})^2}{\mathbf{p}_{k+1}^T \mathbf{p}_{k+1}^{(k)}} \tag{14}$$

where $\mathbf{p}_{k+1}^{(k)} \triangleq \mathbf{R}_k \mathbf{p}_{k+1}$. According to [14], an auxiliary matrix $\mathbf{A} \in \Re^{n \times n}$ and a vector $\mathbf{b} \in \Re^{n \times 1}$ need to be defined to reduce the computational complexity. Their elements are given by:

$$a_{i,j} \triangleq (\mathbf{p}_i^{(i-1)})^T \mathbf{p}_j, \quad 1 \le i \le j \tag{15}$$

$$b_j \triangleq (\mathbf{p}_j^{(j-1)})^T \mathbf{y}, \quad 1 \le j \le n \tag{16}$$

where $(\mathbf{p}_j^{(0)} = \mathbf{p}_j)$. The efficiency of the this forward stage then follows from updating these terms recursively as:

$$a_{i,j} = \mathbf{p}_i^T \mathbf{p}_j - \sum_{l=1}^{i-1} a_{l,i} a_{l,j} / a_{l,l} \tag{17}$$

$$b_j = \mathbf{p}_j^T \mathbf{y} - \sum_{l=1}^{j-1} (a_{l,j} b_l) / a_{l,l} \tag{18}$$

By substituting (15) and (16) into (14), the net contribution of a new RBF centre $\mathbf{p}_{k+1}$ to the cost function can then be expressed as:

$$\Delta J_{k+1}(\mathbf{p}_{k+1}) = \frac{b_{k+1}^2}{a_{k+1,k+1}} \tag{19}$$

In heuristic methods, (19) provides the formula to evaluate each solution in the population. For instance, the selection of local best and global best particles in PSO or the selection between trial vector and target vector in DE.

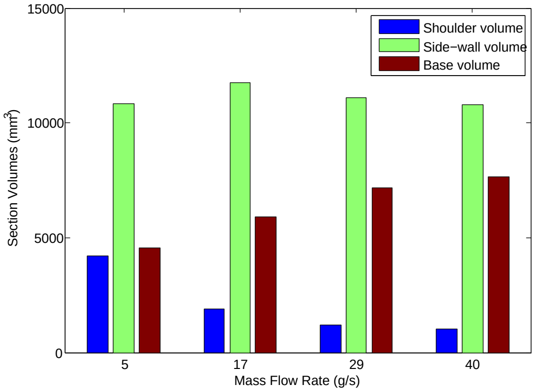

Fig. 1. Effect of mass flow rate on the section volumes (The other three parameters were kept constant, and set as Pressure = 10bar, Temperature = 100°C, Timing = 50ms

B. Second stage - backward network refinement

The model from the first stage is not optimal due to the correlations between selected terms, that is the prior selected centres introduced a constraint on the latter selections. This second stage is therefore adopted to eliminate such constraint and replace any insignificant centres by new one generated from the population. Clearly, the last selected centre in the forward construction has always been maximally optimized for the whole network, the backward refinement can be divided into two main steps: Firstly, a selected centre $\mathbf{p}_k$, $k = 1, \cdots, n-1$ is shifted to the n^{th} position as if it were the last selected one; then the optimization is implemented to find an alternative centre at the n^{th} position based on the re-ordered $n-1$ centres. If the shifted one is less significant than the new centre from the population it will be replaced, leading to a reduced training error and potential improvement in the generalisation capability. This review is repeated until a pre-defined number of check loops is reached. The detailed algorithm and its computational analysis can be found in [13].

V. Experimental results

According to table I, a total number of 144 experiments need to be carried out to generate the required data. However, running these experiments and measuring the section weights are practically difficult. Therefore, a manufacturing simulation software, known as ABAQUS/standard version 6.10 [2], was adopted as a substitute of the real process. Each simulation lasted around 20 minutes, and when all finished, four experiments were regarded as failure with their associated results being removed from the final data set. For the remaining 140 data points, 100 were used for RBF model training, and another 40 were reserved for model validation. The effect of mass flow rate is illustrated in Fig. 1 while other three parameters were kept constant.

The two-stage selection algorithm integrated with PSO, DE, and ELM was then applied on the training data. For heuristic optimization methods, their algorithm parameter settings are shown in table II. From the implementation, it is found that

TABLE II
PARAMETER SETTINGS FOR THE HEURISTIC APPROACHES (THE INITIAL CENTRE VECTORS WERE RANDOMLY SELECTED FROM THE TRAINING DATA)

Method	Parameter	Value	Description
PSO	$\mathbf{x}_i$	$[c_i; \sigma_i]$	i^{th} particle in the swarm
	σ_i	$\in [0.1, 4]$	Range of the width of i^{th} RBF centre
	S	20	Number of particles
	G	30	Number of updating cycles
	w_0	0.8	Inertia weight in velocity updating
DE	$\mathbf{x}_i$	$[c_i; \sigma_i]$	i^{th} solution vector in the population
	σ_i	$\in [0.1, 4]$	Width of i^{th} RBF centre randomly generated from the specified range; if $\sigma_i < 0$ in the mutation step, $\lvert\sigma_i\rvert$ is used
	S	10	Population size
	G	20	Maximum number of generations
	F	0.8	Weight of vector difference
	C_r	0.6	Crossover constant
ELM	σ_i	$\in [0.1, 4]$	Width of i^{th} RBF centre randomly generated from the specific range;

TABLE III
COMPARISON OF BOTTLE SHOULDER VOLUME MODELLING PERFORMANCES (RMSE)

Algorithm	Model size	Training error	Test error
PSO+FRA	4	0.2395	0.2770
PSO+TSS	4	0.1687	0.1998
DE+TSS	4	0.1780	**0.1869**
ELM+TSS	4	0.2531	0.2784
PSO+FRA	5	0.2333	0.2319
PSO+TSS	5	0.1441	0.1575
DE+TSS	5	0.1448	**0.1216**
ELM+TSS	5	0.2354	0.2408
PSO+FRA	6	0.1791	0.1972
PSO+TSS	6	0.1297	**0.1272**
DE+TSS	6	0.1089	0.1293
ELM+TSS	6	0.1896	0.2261

the increase of swarm size normally affects the performance more than the increase of updating cycles in PSO, while in differential evolution, these two control parameters have the similar effects.

The resultant RBF models are usually evaluated by the Root Mean Squared Error (RMSE) on validation data, table III - V illustrate such performances under different algorithms. Due to the stochastic characteristics in heuristic optimization, the best results from five runs are chosen for comparison. The fast recursive algorithm (first stage of TSS) usually gives worse results than the TSS alternative, so it was only tested in the modelling of bottle shoulder volume. The results also indicate that DE works better than PSO in the shoulder volume modelling. However, PSO outperforms DE in the other two section volume modelling. Extreme learning machine is definitely the most efficient methods which runs much faster than PSO or DE, however the corresponding model performances are less favourable. The prediction of bottle base volume is also shown in Fig. 2, while the associated RBF model parameters are given in table VI.

TABLE IV
COMPARISON OF BOTTLE SIDE WALL VOLUME MODELLING PERFORMANCES (RMSE)

Algorithm	Model size	Training error	Test error
PSO+TSS	4	0.4084	**0.4023**
DE+TSS	4	0.4201	0.4531
ELM+TSS	4	0.4506	0.5505
PSO+TSS	5	0.3761	**0.3761**
DE+TSS	5	0.3481	0.3935
ELM+TSS	5	0.4194	0.4996
PSO+TSS	6	0.3523	0.4926
DE+TSS	6	0.3379	**0.4049**
ELM+TSS	6	0.4156	0.4425

TABLE V
COMPARISON OF BOTTLE BASE VOLUME MODELLING PERFORMANCES (RMSE)

Algorithm	Model size	Training error	Test error
PSO+TSS	4	0.1300	**0.1281**
DE+TSS	4	0.1518	0.1508
ELM+TSS	4	0.2500	0.2959
PSO+TSS	5	0.1148	**0.1021**
DE+TSS	5	0.1363	0.1260
ELM+TSS	5	0.1907	0.2215
PSO+TSS	6	0.0898	**0.0938**
DE+TSS	6	0.1054	0.0990
ELM+TSS	6	0.1834	0.2104

TABLE VI
OPTIMIZED PARAMETERS OF RBF MODEL FOR THE BOTTLE BASE VOLUME PREDICTION (REFER TO (1))

Parameters	Optimized values
θ_1	1.5419
θ_2	-2.6327
θ_3	2.5896
θ_4	-3.0357
θ_1	3.0150
$\boldsymbol{c}_1$	[-0.051, 0.244, 1.635, 1.499]
$\boldsymbol{\sigma}_1$	[0.100, 4.000, 4.000, 0.100]
$\boldsymbol{c}_2$	[-1.605, 1.475, -0.112, 1.499]
$\boldsymbol{\sigma}_1$	[0.244, 4.000, 4.000, 4.000]
$\boldsymbol{c}_3$	[1.575, 1.475, 1.635, 0.451]
$\boldsymbol{\sigma}_3$	[0.487, 4.000, 2.986, 0.276]
$\boldsymbol{c}_4$	[0.946, -1.475, -1.574, 1.499]
$\boldsymbol{\sigma}_4$	[4.000, 4.000, 2.008, 4.000]
$\boldsymbol{c}_5$	[1.185, 0.632, -0.332, 1.499]
$\boldsymbol{\sigma}_5$	[2.966, 4.000, 2.454, 1.030]

VI. CONCLUSION AND FUTURE WORK

Injection stretch-blow moulding is a typical process to produce plastic bottles in the industry. The process is usually controlled by several adjustable parameters, such as the mass flow rate, blow pressure and preform temperature. The quality of resultant bottles can be measured by its top loads, or the section weights distribution. Due to the lack of closed-loop control, large variations can be observed on the quality indicators.

Section weights are found to be the most prospective variable for feed-back control, but cannot be measured directly in a typical process. Thus, a soft-sensor approach based on

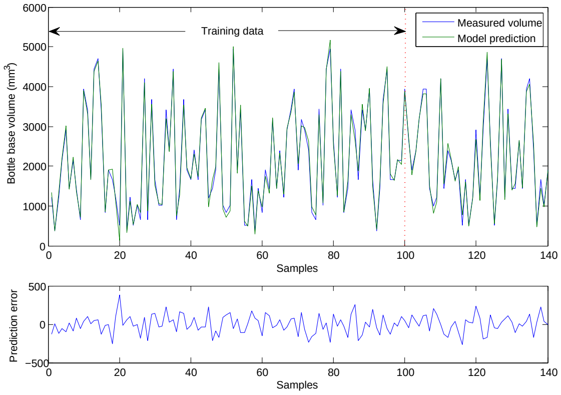

Fig. 2. The bottle base volume model prediction by TSS+PSO (5 hidden nodes were used, the first 100 samples show the training performance while the rest 40 points are model prediction)

an inferential mathematical model becomes an affordable alternative for control implementation. This paper uses a radial basis function network model to predict the bottle section weights. The main issue involved in RBF model construction is the determination of non-liner parameters in the hidden nodes. As the gradient-based approaches require large computational effort, heuristic optimization methods, such as particle swarm optimization, deferential evolution, and extreme learning machine become appropriate alternatives. In this paper, these heuristic optimization methods are effectively integrated with our recently proposed two-stage subset selection algorithm, leading to an efficient RBF model construction algorithm. Experimental results on simulation data has successfully verified the effectiveness of the proposed method. Future work will use practical data to build the models and implement iterative learning control for the ISBM process based on the soft-sensors.

Acknowledgment

This work was financially supported by Engineering and Physical Sciences Research Council (EPSRC) under grant number EP/G059489/1 and partly supported by EPSRC grant EP/F021070/1, EP/H020756/1, and EP/G042594/1, and Science and Technology Commission of Shanghai Municipality (11ZR1413100).

References

[1] C. Abeykoon, K. Li, M. McAfee, P. J. Martin, and G. W. Irwin, “Extruder melt temperature control with fuzzy logic,” in *Proceedings of the 18th IFAC World Congress*, 2011, pp. 8577–8582.

[2] Z. Yang, E. Harkin-Jones, G. Menary, and C. Armstrong, “A non-isothermal finite element model for injection stretch-blow molding of pet bottles with parametric studies,” *Polymer Engineering & Science*, vol. 44, no. 7, pp. 1379–1390, 2004.

[3] G. Menary, C. Tan, C. Armstrong, Y. Salomeia, M. Picard, N. Billon, and E. Harkin-Jones, “Validating injection stretch-blow molding simulation through free blow trials,” *Polymer Engineering & Science*, vol. 50, no. 5, pp. 1047–1057, 2010.

[4] S. Chen and S. Billings, “Neural networks for nonlinear dynamic system modelling and identification,” *International Journal of Control*, vol. 56, no. 2, pp. 319–346, 1992.

[5] S. McLoone, M. Brown, G. Irwin, and G. Lightbody, “A hybrid linear/nonlinear training algorithm for feedforward neural networks,” *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 669–684, 1998.

[6] S. Kirkpatrick, “Optimization by simulated annealing: Quantitative studies,” *Journal of Statistical Physics*, vol. 34, no. 5, pp. 975–986, 1984.

[7] Z. Michalewicz, *Genetic algorithms + data structures = evolution programs*. Springer, 1996.

[8] F. Glover and R. Marti, “Tabu search,” *Metaheuristic Procedures for Training Neutral Networks*, vol. 36, pp. 53–69, 2006.

[9] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proc. IEEE International Conference on Neural Networks*, vol. 4, Perth, Australia, 1995, pp. 1942–1948.

[10] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[11] M. Dorigo and L. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 2002.

[12] G. Loganathan, “A New Heuristic Optimization Algorithm: Harmony Search,” *SIMULATION*, vol. 76, no. 2, pp. 60–68, 2001.

[13] J. Deng, K. Li, G. Irwin, and M. Fei, “Two-stage RBF network construction based on PSO,” *Transactions of the Institute of Measurement and Control*, vol. 0, no. 0, pp. 1–9, 2011.

[14] K. Li, J. Peng, and G. Irwin, “A fast nonlinear model identification method,” *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1211–1216, 2005.

[15] J. Deng, K. Li, and G. Irwin, “Locally regularised two-stage learning algorithm for RBF network centre selection,” *International Journal of Systems Science*, no. 1, pp. 1–14, 2011.

[16] G. Huang, Q. Zhu, and C. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, pp. 489–501, 2006.

[17] J. Deng, K. Li, and G. Irwin, “Fast automatic two-stage nonlinear model identification based on the extreme learning machine,” *Neurocomputing*, vol. 74, no. 16, pp. 2422–2429, 2011.

[18] G. Huang, L. Chen, and C. Siew, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.