Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Dynamic Soft-sensing Method Based on Impulses Response Template and Parameter Estimation with Modified DE Optimization

Wenxiang Lu, Qing Yang, Dexian Huang, Yihui Jin

Department of Automation, Tsinghua University, Beijing 100084, China, (e-mail: lwx02@mails.thu.edu.cn)

Abstract: As major experimental modeling methods, static soft-sensing methods have been widely used in modern chemical production process now. In fact, for the sampling frequency to output variable by laboratory analyzer off-line is rather low and uniform, the computed results, which gained from those methods or the existed major dynamic methods such as neural networks, are difficult to satisfy the requirements of dynamic control on-line. A dynamic soft-sensing method (DSSM) based on impulse response template (IRT) and parameter estimation using differential evolution (DE) optimization is presented in this paper. However, for a multi-variables system, learning of template parameters still takes large computational cost, and is not only slow in the convergence speed but also easy to be trapped into local optima so as to enlarge the modeling errors. To account for these problems, the original DE (ODE) is modified in the aspects of scaling factor and crossover rate, which could dynamically change with iterative loops. Subsequently, a complete implementation of the modified DE (MDE) is presented. Experiment based on hysys simulation of a primary tower system to build a three-inputs-single-output model is carried out, under various impulse response length and noise standard, and the final comparison results demonstrate the effectiveness and robustness of this method.

1. INTRODUCTION

With the increasing demanding on production control in modern chemical process industry, the problem of on-line, real-time and precise measurement to some crucial variables, which effecting the productivity and the product quality, is essential. Thus, soft-sensing methods (van den Bos, 1977) and related applied technology are put forward to solve the problem. The latest researches show that soft-sensor has become a hotspot and major developing trend in the fields of process control and process examining (McAvoy, 1992).

At present, static soft-sensing methods such as learning network (Willis et al., 1992; Assis and Filho, 2000) have been widely used as basic experimental modeling methods. The main reasons are that the sampling frequency to output variables is too low to meet dynamic modeling, for they can usually be obtained from laboratory analyzer off-line only. However, due to the dynamical behavior in practical production process, the static models using only few of samples of the input variables are easy to be over-learned for the noise and difficult to satisfy the requirements of dynamic control on-line. Meanwhile, various dynamic models such as time-delay network, recurrent neural network (Adali et al., 1997; Shaw et al., 1997; Shi et al., 2003; Fortuna et al., 2005) are also introduced into dynamical soft-sensing modeling, which basically are converted into static networks in form by adding time-delay units to both sides of network to introduce dynamic process information. Yet, in practice, these time-delay signals are hard to be obtained due to sparse or uniform sampling, especially to output. Ma et al. (2005) introduces a dynamic soft-sensing method based on impulse response (IR), but in which the IR is simplified to few broken lines, and this brings biggish errors for discarding the practical profile. To account for these problems, in this paper a novel dynamic soft-sensing method (DSSM) is present, which is based on impulse response template (IRT) and parameter estimation by modified differential evolution (MDE) optimization.

The remainder of this paper is organized as follows. In section 2, a detailed analysis of typical chemical production process system is first given. Then, on the basis of IRT and parameter optimization, a dynamic soft-sensing model is built. At last, a complete implementation of this model using MDE is shown. In section 3, numerical simulation, comparison and discussion are involved. In final section 4, we conclude with a brief summary of experimental results.

2. MODEL

2.1 Building a dynamic soft-sensing system

A typical chemical production process system with multi-inputs-multi-outputs could be decomposed into several multi-inputs-single-output sub-systems and each sub-system is made up of several control modules which are independent with each other. Then, a typical soft-sensing process system with multi-inputs-multi-outputs could be depicted in Fig. 1.

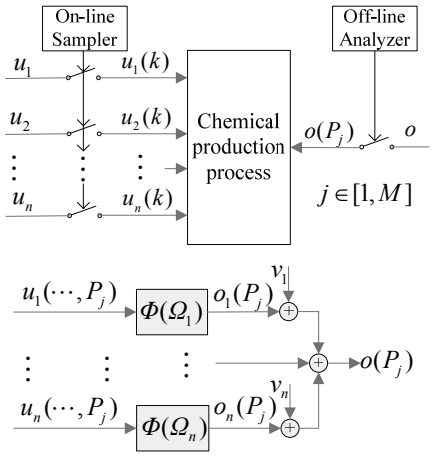

Fig. 1. A typical chemical production process soft-sensing system with multi-inputs-single-output

In Fig.1, u_1, u_2, …, u_n denote input variables, which are inherently sensitive to output, and can be measured directly by on-line sampler at the same sampling instant k (about several seconds). o denotes output variable, which is the crucial unobserved variable to be estimated and only be measured by off-line analyzer with uniform sampling interval P_j (about several hours). $\Phi(\Omega_s)$ denotes sub unit s of the system, which could be characterized by the transfer function. v_j denotes the noise. M is the sampling times to output.

Suppose unit s is a linear, time-invariant and causal system and its impulse response function is $h_s(i)$. Then, unit s could be described as the following discrete-time equation:

$$o_s(P_j) = \sum_{i=0}^{+\infty} h_s(i) u_s(P_j - i), s \in [1, n], j \in [1, M] \tag{1}$$

Considering causality and stability of the system, thus $h_s(i)$ should meet the following equations

$$h_s(i) = 0, (i < 0); \sum_{i=0}^{+\infty} |h_s(i)| < \infty; \lim_{i \to \infty} h_s(i) = 0 \tag{2}$$

For a stable system in practice, after L_s (>>1) steps, $h_s(L_s)$ is approximately approach to zero. So, (2) can be written for

$$o_s(P_j) = \sum_{i=0}^{L_s} h_s(i) u_s(P_j - i) \tag{3}$$

Similarly, we can establish equation as (3) for each unit. Then, the final model of the system can be described as follows:

$$o(P_j) = \sum_{s=1}^{n} o_s(P_j) = \sum_{s=1}^{n} \left(\sum_{i=0}^{L_s} h_s(i) u_s(P_j - i) \right) \tag{4}$$

From (4), it is easy to be seen that the dynamic relations between inputs and output are established by means of the impulse response. Equation (4) exactly represents the dynamic soft-sensing model in training mode. Assume that the value of the impulse response series have been achieved, the unmeasured variable can be estimated easily using the following model in applied mode.

$$y(j) = \sum_{s=1}^{n} \left(\sum_{i=0}^{L_s} h_s(i) u_s(j - i) \right) \tag{5}$$

Consequently, the key to implement the dynamic model is how to solve or estimate the impulse response series of (4).

2.2 Impulse response template and parameter optimization

The common method to solve the impulse response series is using system identification method based on least square (LS) principle. In the following, we will exhibit a novel solution using IRT and parameter optimization.

It is well known that the impulse response is a complete characterization of the system. From the past experiences of modeling chemical production process system, the order of each sub model of the system can be decreased to 2 approximately. Consequently, the impulse response series in (4) could be written as

$$\begin{aligned} h_s(i) &= \Phi(\Omega_s) = \Phi((T_1^s, T_2^s, K_p^s, \tau^s)) \\ &= \frac{K_p^s}{T_1^s - T_2^s} \left(\exp\left(-\frac{i - \tau^s}{T_1^s} \right) - \exp\left(-\frac{i - \tau^s}{T_2^s} \right) \right), \\ & \qquad i = \tau^s + 1, ..., L_s, s \in [1, n]; \\ h_s(i) &= 0, i = 1, ..., \tau^s, \end{aligned} \tag{6}$$

where $\Omega_s = (T_1^s, T_2^s, K_p^s, \tau^s) \in \boldsymbol{R}^{+4}$ denotes the unit's parameters, which represent the first time-constant, the second time-constant, steady-state gain, and pure delay constants in turn. These parameters should meet the following constraint

$$\begin{aligned} &\Omega_s \in \boldsymbol{E_s} \\ &= \{(T_1^s, T_2^s, K_p^s, \tau^s) \,|\, T_1^s \in [\xi_{s1}^{\min}, \xi_{s1}^{\max}], T_2^s \in [\xi_{s2}^{\min}, \xi_{s2}^{\max}], \\ &\qquad K_p^s \in [\xi_{s3}^{\min}, \xi_{s3}^{\max}], \tau^s \in [\xi_{s4}^{\min}, \xi_{s4}^{\max}]\} \end{aligned} \tag{7}$$

where $\boldsymbol{E_s}$ means the subspace of the corresponding model in the 4-dimention positive real number space, while $\xi_{sj}^{\min}$, $\xi_{sj}^{\max}$ (j=1,2,3,4) represent the lower and upper bounds of each parameters.

By (6) or (7), the relations between the system's parameters and the impulse response series to be estimated are built. Suppose a set of original impulse response series is produced by (6) using a set of original parameters satisfying (7). After that, another set of parameters in the given bounds is chosen to produce the corresponding set of series in the same way. As a result, it can be seen that the profile of the obtained series is similar to that of the original set, and which is just transformed by horizontal movement, horizontal scaling or vertical scaling in form. So, in a sense, the obtained series may be derived from some certain template, or from the original impulse response series. Here, we call the original parameters as template parameters (TP), and the impulse response series produced by which is called as impulse response template (IRT). Obviously, if TP and its subspace are given, the impulse response series to be identified in (4) would be confined. Consequently, to a multi-variables system, which has many impulse response series parameters to be identified, By IRT and TP, the

complex learning problem would be easily transformed into a simpler learning problem to finite template parameters and the number of the learning parameters are shorten dramatically, obviously, the derived benefits of which are that the poor generality of the model aroused by over-learning is improved efficiently, as well as the consuming time to learn parameters is decreased greatly.

From the analysis above, the key to implement dynamic soft-sensing model is to obtain excellent template parameters. From the viewpoint of optimization, it is a problem of parameter optimization. The principle of parameters estimation for the model in sense of optimization could be illustrated with the following Fig. 2.

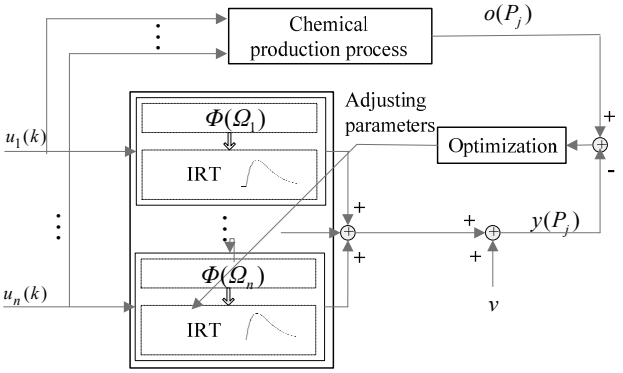

Fig. 2. The principle of parameter estimation for dynamical soft-sensing model

In Fig. 2, the problem of template parameters estimation could be formulated as the following optimization problem by searching suitable Ω_s

$$\min J = \frac{1}{M}\sum_{j=1}^{M}\left(\sum_{s=1}^{n}\left(\sum_{i=0}^{L_s}\Phi(\Omega_s,i)u_s(P_j-i)\right)-o(P_j)\right)^2 \qquad (8)$$

$$s.t.\ \ \Omega_s \in \boldsymbol{E_s}$$

where J is the minimum objective function to be optimized.

2.3 Model implementation with Modified differential evolution optimization

DE, which is gaining much research interests recently, is a novel intelligent optimization method. Various studies have shown that DE has efficient globally searching ability and robust performance (Bhat et al., 2006; Liu et al., 2007). Compared with other optimization methods such as particle swarm optimization (PSO) and genetic algorithm (GA), DE has simpler concept and easier implementation. Consequently, a DE approach is introduced into searching the best template parameters to implement the dynamic soft-sensing model in this paper.

However, for a complicated system with multi-variables, learning of lots of template parameters still takes large computational cost, and is easy to be trapped into local optima to enlarge the modeling errors. To account for the problems, the original DE (ODE) is modified in the aspects of scaling rate and crossover rate, which are substituted by variables changing dynamically with iterative loops. A complete solution of MDE is formulated as follows:

1) Randomly initialize the population of individual. Suppose the size of the population is *NP*, the *k*th individual X_k is a vector of 4*n*-dimension and denotes a potential solution of Ω_s to be decided. Here, X_k is written as

$$\begin{aligned} X_k &= \{x_{k1}, x_{k2}, ..., x_{k,4n}\} = \{\Omega_1, \Omega_2, ..., \Omega_n\}_k \\ &= \{T_1^1, T_2^1, K_p^1, \tau^1, T_1^2, T_2^2, K_p^2, \tau^2, ..., T_1^n, T_2^n, K_p^n, \tau^n\}_k \end{aligned} \qquad (9)$$

Each element in the trial vector should satisfy the limitation $x_{kr} \in [x_r^{\min}, x_r^{\max}]$ (r=1,2,…,4n), where $x_r^{\min}, x_r^{\max}$ denote the lower and upper bounds in search space respectively, and their values have been given by (7).

Set *maxiter* as the maximum iterative loops. Take the iterative loop *iter* as 0 and generate randomly the initial population by

$$\begin{aligned} x_{kr}^{iter} &= x_{kr}^{\min} + rand(0,1)\times(x_{kr}^{\max} - x_{kr}^{\min}) \\ &(k = 1,2,...,NP;\ r = 1,2,...,4n); \end{aligned} \qquad (10)$$

2) By (8), calculate the objective function values J_k^{iter} for each individual at generation *iter*, determine the best objective value in all generations, which is recorded as $J_{\min}^{iter} = \min(J_k^{iter})$, and *gBest* is the optimal individual which produces the best objective value.

3) If the stopping criterion $J_{\min}^{iter} < \varepsilon$ or *iter* > *maxiter* is met, then output *gBest*, otherwise continue.

4) Mutation. For each individual to perform mutation operation according to the following equation

$$X_k^* = X_{a1}^{iter} + F\times(X_{a2}^{iter} - X_{a3}^{iter}) \qquad (11)$$

where X_k^* is the mutant counterpart of X_k^{iter} at generation *iter*, *a*1, *a*2, *a*3 are three integers chosen randomly and be mutually different, *F* denotes scaling factor which controls amplification of the differential variation $(X_{a2}^{iter} - X_{a3}^{iter})$. As a general rule, *F* is a constant chosen from (0, 2). In MDE, *F* is substituted by a variable which could dynamically changing with iterative loops, that is

$$F = FF_{\max} - [(FF_{\max} - FF_{\min})\times iter]/maxiter \qquad (12)$$

where $FF_{\max}$, $FF_{\min}$ represent the maximum and minimum scaling factors respectively. Obviously, in the initial phase, *F* is bigger and is beneficial to search globally so as to diversify the population, while in the latter phase, *F* is smaller, which is advantageous to enhance local searching so as to improve the precision and quicken the convergence speeding.

5) Crossover. In order to further increase the diversity of the population, perform binary-distribution crossover operation at each element of between the mutant vector X_k^* and X_k^{iter} according to

$$x_{kr}^{\Delta} = \begin{cases} x_{kr}^{iter}, & if\ rand(0,1) < C \\ x_{kr}^{*}, & otherwise \end{cases} \qquad (13)$$

where C denotes crossover factor which controls the differential degree of each element, and is usually a constant chosen from (0, 1). Similarly, in MDE, C is substituted by

$$C = CC_{\min} + [(CC_{\max} - CC_{\min}) \times iter] / maxiter \tag{14}$$

where $CC_{\max}$, $CC_{\min}$ represent the maximum and minimum crossover factors respectively. Apparently, in the initial phase, C is smaller, thus the opportunity of crossover is greater and it is beneficial to increase the diversity of the population and improve the performance of searching globally, In the latter C is bigger, so the chance of crossover is smaller and is advantageous to improve the precision and quicken the convergence speeding.

6) Bounds check (Bhat et al., 2006). For the values of partial elements of the new individuals may be out of the prescribed bounds, then we should force it to lie within the given bounds according to the following judgments

$$\begin{aligned} x_{kr}^{\Delta} &= x_{kr}^{\min} + 2.0(p/q)(x_{kr}^{\max} - x_{kr}^{\min}), \\ &\quad if\ x_{kr}^{\Delta} > x_{kr}^{\max} \\ x_{kr}^{\Delta} &= x_{kr}^{\min} + 2.0(q/p)(x_{kr}^{\max} - x_{kr}^{\min}), \\ &\quad if\ x_{kr}^{\Delta} < x_{kr}^{\min} \end{aligned} \tag{15}$$

where $p = x_{kr}^{\Delta} - x_{kr}^{\max}$ and $q = x_{kr}^{\Delta} - x_{kr}^{\min}$.

7) Competition. The objective function value of the new individual X_k^{Δ} is not always better than that of the former X_k^{iter}. So both of them should compete the opportunity of producing the next generation by the following competitive operation

$$X_k^{iter+1} = \begin{cases} X_k^{\Delta},\ if\ \ J_k^{\Delta} < J_k^{iter} \\ X_k^{iter},\ otherwise \end{cases} \tag{16}$$

8) *iter* = *iter*+1, and goto step 2.

3. EXPERIMENTAL

3.1 Experimental design

In simulation, HYSYS, well-known simulation software for chemical production process, is used to produce the datasets. A primary tower of the crude distillation unit is simulated as the object, and the inputs are the reciprocal of top absolute pressure, top temperature and the character factor of the tower, while the output is the 100% cut point (ASTM D86) of the top product, naphtha. To test the ability of anti-disturbance of the model, we add white noises to all the measurements, including the inputs, according to the precision of common instruments.

In the following, we produce three training datasets as Train1, Train2 and Train3, to train model, and one testing dataset as Test, to test the performance of generalization of the model. Before training, all the inputs and output is zero-mean transformed and the transformation parameters are parts of the final model. All datasets are shown in Table 1:

Table 1. Experimental datasets

Data sets	Input	Output		
	Available	Type	Cycle	Available
Train1	48000	Sparse	480	100
Train2	10000	Sparse	100	100
Train3	100000	Sparse	100	1000
Test	300	Continuous	1	300

Three experimental models are included for comparison: (1) DSSM based on IRT and MDE, which simply marked as MDE_IRT; (2) DSSM based on IRT and ODE, simply marked as ODE_IRT; (3) DSSM based on LS and IR, simply marked as LS_IR. To distinguish between MDE, ODE and LS, MDE and ODE are uniformly defined as DE.

The common parameters in both MDE and ODE are: *NP*=50, *maxiter*=500, ε=0.5, and the bounds of parameters to be estimated are $T_1^s \in [0,90]$, $T_2^s \in [0,90]$, $K_p^s \in [0,100]$, $\tau^s \in [0,15]$ (s=1,2,3). In practice, they could be determined on the basis of the mechanism analysis of producing process or by experiences. In ODE, set F=1.0, C=0.5. In MDE, set $FF_{\min}$=0.1, $FF_{\max}$=1.8, $CC_{\min}$=0.1 and $CC_{\max}$=0.9.

To evaluate the error of the models, the mean square error (*MSE*) is used as the error criterion

$$MSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}[o(i) - y(i)]^2} \tag{17}$$

where $o(i)$ is the output without normalization, while $y(i)$ is the predictive output after inverse normalization.

3.2 Experimental results and discussion

In simulation, we build various DE_IRT and LS_IR models, of which the forced truncation length of impulse response, $L(=L_1=L_2=L_3)$, is chosen respectively from 20, 40, …, 240 (the step is 20), both with and without noise. Then, under the particular L and noise standard, we use three different training datasets to train DE_IRT and LS_IR models. The *MSE* results of both train and test datasets are shown from Table 2 to Table 4. In these tables, the figures with "□", "▒", "_" denote the minimum *MSE* value in the located column.

Table 2. Under different noise standards and L, the *MSE* results in "Train1/Test"

L	Without Noise			With Noise		
	IRT		IR	IRT		IR
	MDE	ODE	LS	MDE	ODE	LS
60	3.77/3.83	4.13/4.10	4.80/4.81	4.07/4.12	4.13/4.09	5.91/6.02
120	3.29/3.32	3.73/4.80	4.26/4.30	5.40/5.41	5.86/5.97	5.95/6.01
180	1.37/1.40	1.39/1.42	4.73/4.84	4.21/4.28	4.41/4.39	5.83/6.05
240	1.07/1.09	1.50/1.52	5.12/5.41	3.16/3.20	3.80/3.77	5.65/5.81

Table 3. Under different noise standards and L, the *MSE* results in "Train2/Test"

L	Without Noise			With Noise		
	IRT		IR	IRT		IR
	MDE	ODE	LS	MDE	ODE	LS
60	4.08/4.02	4.19/4.16	5.09/5.18	4.95/4.96	4.96/5.01	5.61/5.85
120	3.22/3.22	3.31/3.38	5.53/5.65	5.92/5.94	5.94/6.03	5.44/5.59
180	1.30/1.32	1.40/1.42	5.57/5.63	3.39/3.44	3.77/3.80	5.75/6.11
240	0.97/0.8	1.46/1.49	5.42/5.72	2.46/2.45	2.53/2.51	5.33/5.59

Table 4. Under different noise standards and L, the *MSE* results in "Train3/Test"

L	Without Noise			With Noise		
	IRT		IR	IRT		IR
	MDE	ODE	LS	MDE	ODE	LS
60	4.01/3.99	4.14/4.15	3.80/3.79	4.19/4.22	4.23/4.30	5.69/5.72
120	2.96/2.96	3.06/3.11	3.31/3.37	2.87/2.89	3.36/3.34	5.64/6.71
180	0.69/0.70	0.71/0.72	3.94/4.15	1.14/1.16	1.42/1.43	5.73/6.79
240	0.62/0.63	0.87/0.87	4.22/4.48	2.30/2.31	2.40/2.38	5.39/6.01

From Table 2 to Table 4, it can be seen that the *MSE*s of the three models are great when L is small and the *MSE*s of DE_IRT models are decreasing gradually with the increasing L, while the *MSE*s of LS_IR models are always big and the rule is not clear. The decreasing rate of the *MSE*s of DE_IRT models is also decreasing with the increasing L. When L and available output number is fixed, the *MSE*s with noise of all are bigger. For LS_IR models, their *MSE*s are relatively smaller when available samples are more and there is no noise. But, compared with DE_IRT models under equal condition, they are much bigger.

Comparing Table2 with Table3, it can be seen that, when available output numbers are fixed and the sampling cycle is decreased, although the available inputs are smaller, the predictive precisions of DE_IRT models are somewhat improved, and a better predictive result can be find in any event either with or without noise. Comparing Table 3 with Table 4, it can be seen that, when the sampling cycle is fixed, the predictive precisions of DE_IRT models are somewhat improved with the increasing available output numbers. But the degree of improvement is not obvious when there is noisy.

Note that the proper length L_s of the impulse response series is determined by characteristic parameters of the given system such as time constants and pure delay constant, and which is crucial to the performance of the models. In theory, with the increase of L, the dynamic process information to be used for modeling is more, and then the predictive results of the models are relatively better. When L is big enough to be able to show all the impulse response series, the effect is not obvious.

Fig. 3 gives a MDE_IRT model in the Train1 without noise and L=240. The corresponding comparison of calculations and original values of Test is shown in Fig. 4.

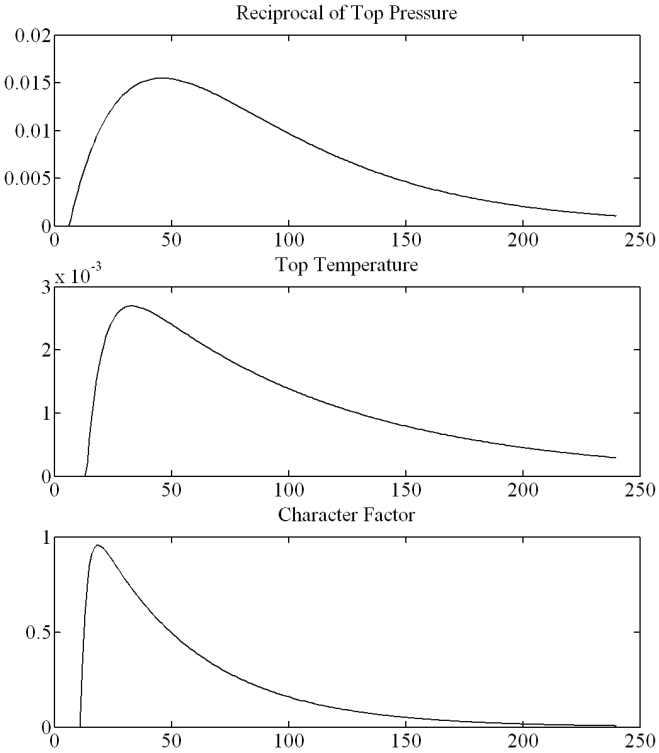

Fig. 3. MDE_IRT model in Train1 without noise and L=240

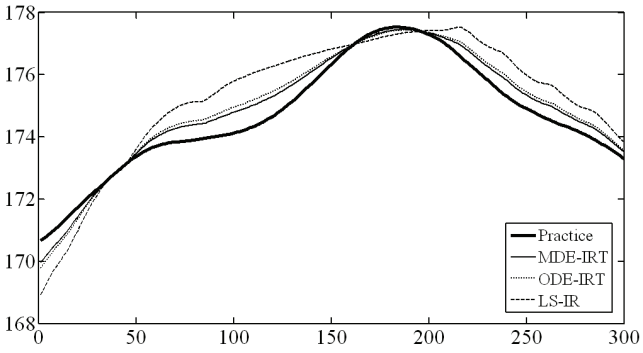

Fig. 4. Comparison of calculations and original value of Test

From Fig. 3 and Fig. 4, the dynamics character of the MDE_IRT model is shown expressly, and this result accords with the general experience of chemical process. From Fig. 4, it can be seen that all the calculations of three models have similar trends to the practice and the DE_IRT model is much better than LS_IR model.

Comparing the DE_IRT and LS_IR models in the experiment, the results are distinct with increasing L. For DE_IRT, the *MSE*s of all models are smaller comparatively when L is bigger. On one hand, for the impulse response series are confined to the given template, its performance is completely decided by the corresponding template parameters and bounds. When L is larger and available samples are more, more efficient dynamic information is

provided to DE_IRT modeling exactly. On the other hand, owing to better performance of searching ability and anti-disturbance of DE, so only if the training time is enough, the final DE_IRT models would, all the time, give a better result. However, the results of the LS_IR models do not have the same characters, and the reasons lie in that, LS_IR is based on the least square principle, and the solution is absolutely depend upon the mutual related or correlated information between input and output of sampling system. When the sampling cycle is greater or there is noise, the accuracy of samples is low so that the LS_IR models obtained are distinct from the norm one and have enlarged the errors of prediction. Especially, when the increasing cumulative computational error results in the divergence of the least square solution due to larger L, the obtained model is completely different from the true so that the results of which is invalid.

Seen from Table 2 to Table 4, the *MSE*s of MDE_IRT models are somewhat smaller than those of ODE_IRT models under the same conditions, for the MDE method can increase the searching speed and the ability of preventing the optimization object from being trapped into local optima. Thus, it is seen that MDE_IRT model has improvements against ODE_IRT in predictive precision.

4. CONCLUSIONS

To the best of our knowledge, this is the first report of applying IRT and DE method to the problem of dynamic soft-sensing modeling in chemical production process, from the viewpoint of optimization. By simulation we could draw the following conclusions:

1) Using IRT and DE in DE_IRT model, the impulse response series achieved are more similar to real ones, and the estimation error is decreased apparently which is stirred by irregularity of the least square solution, especially for the system suffering from much disturbances.

2) The selection of the length L of the impulse response series is crucial. L should not be too small in the impulse response model. The small one makes much foregone input information lost and causes the increase of estimation error. Similarly, the length should not too large. On one hand, the large one will prolong the training time, and on the other hand, for DE_IRT model with larger L, the effect of degrading the errors is not obvious under noisy environment. Inversely, some efficient information would be submerged by the increasing cumulative computational error as well as decreasing the precision of the model. The realistic range of length is determined by time constants and pure delay constant of the given system. But, in practice, it could be determined by trial-and-error approach or the previous optimal method automatically.

3) The model performance is relevant to the amount of available modeling data and the sampling frequency. Generally speaking, if the sampling frequency is higher, the modeling data demanded is more, and vice versa. Only considering the side of modeling data, the amount of data demanded to train the model based on IRT is smaller than those to train the model based on LS and IR, yet the predictive precision is higher. This point is of significance to practical application.

ACKNOWLEGEMENTS

This work was supported by NSFC (Grant No. 60574072), and the national 863 program (Grant No. 2006AA04Z168, 2007AA04Z193) of China.

REFERENCES

Adali, T., Bakal, B., Sonmez, M.K. and et al. (1997). Modeling nuclear reactor core dynamics with recurrent neural networks. *Neurocomputing*, **15**, 363-381.

Assis, A.J. and Filho, R.M. (2000). Soft sensors development for on-line bioreactor state estimation. *Computer & Chemical Engineering*, **24**, 1099-1103.

Fortuna, L., Graziani, S. and Xibilia, M.G. (2005). Soft sensors for product quality monitoring in debutanizer distillation columns. *Control Engineering Practice*, **13**, 499-508.

Liu, B., Wang, L., Jin, Y.H. and et al. (2007). Control and synchronization of chaotic systems by differential evolution algorithm. *Chaos, Solitons and Fractals*, **34**, 412-419.

Ma, Y., Huang, D.X. and Jin, Y.H. (2005). Discussion about dynamic soft-sensing modelling. *Journal of Chemical Industry and Engineering* (*China*), **56**(8), 1516-1519

McAvoy, T.J. (1992). Contemplative stance for chemical process control. *Automatica*, **28**(2), 441-442.

Shaw, A.M., Doyle, F.J. and Schwaber, J.S. (1997). A Dynamic neural network approach to nonlinear process modeling. *Computer & Chemical Engineering*, **21**(4), 371-385.

Shi, D., Zhang, H.J. and Yang, L.M. (2003). Time-delay neural network for the prediction of carbonation tower's temperature. *IEEE Transactions on Instrumentation and Measurement*, **52**(4), 1125-1128.

van den Bos, A. (1977). Application of statistical parameter estimation methods to physical measurement. *Journal of Physics E: Scientific Instruments*, **10**, 753-760.

Bhat, T.R., Venkataramani, D., Ravi, V. and Murty, C.V.S. (2006). An improved differential evolution method for efficient parameter estimation in biofilter modeling. *Biochemical engineering journal*, **28**, 167-176.

Willis, M.J., Montague, G.A, Di Massimo, C. and et al. (1992). Artificial neural network in process estimation and control. *Automatica*, **28**(6), 1181-1187.