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

Evolutionary Improvement Algorithm with Statistical Learning Method for Process Real-Time Optimization

Seungjune Choi, Seunghyok Kim , En Sup Yoon*

* *Department of Chemical and Biological Engineering, Seoul National University*
Shillim-dong 56-1, Kwanak-gu, Seoul, Korea

(e-mail:esyoon@pslab.snu.ac.kr)

Abstract: This study proposes an effective framework for process real-time optimization and data-driven modeling method. The proposed RTO framework with evolutionary improvement algorithm does not wait for the steady-state and it corrects the set-point continuously through the similar way which genetic algorithm exploit to find optimal points. It can deal with higher frequency disturbances and is less influenced by control system performance. Moreover, it is able to address the convergence to suboptimal.

Also, this study proposes statistical learning model (modified Support Vector Machine) that is used in RTO framework. It is able to handle highly-nonlinearity and carry out parameter tuning easily.

The performance of proposed method was successfully illustrated by means of RTO example.

1. INTRODUCTION

As the competition and fast-changing market conditions increase, it is critical for chemical plants to be able to operate optimally over a wide range of conditions. Real-time optimization (RTO), or on-line operations optimization, has a wide appeal in the process industries because of its promise for improving process profitability. The objective of RTO systems is to operate the plant at every instant as near to the optimum operating condition as possible.

Process optimization technology has been maturing over the last decade, and RTO is currently practiced. It is between the planning and control level. The process optimizer gets the constraints and objective function from the planning level. After optimization calculations, it provides a set of set-points for the control level. The plant-wide control system will then move the plant to these set-points. Then the conventional RTO loop can be considered as an extension of a feedback control system. It consists of subsystems for measurement validation, steady-state detection, process model updating, model-based optimization, and command conditioning.

The conventional RTO system has following drawbacks:

(1) Steady state must be achieved for optimization to be performed. Nothing can be done until this condition is satisfied. And the RTO scenario cannot be contemplated, when the frequency of disturbance is higher than the process time constraints.

When several decision variables are involved, the usual optimization procedure might require too much time because it uses the objective function derivative or Hessian matrix.

To implement the set points resulting from the optimization, the plant must still be in the original steady state. The magnitude of set point changes must be bounded for safety reasons.

And RTO is a model-based process control approach that uses current process information (i.e. process model and economic data) to predict the optimal operating policy for a process unit during the next RTO interval. Although most RTO systems attempt to improve model accuracy through model updating, there is always problem such as plant/model mismatch. Both economic and process models require a deep understanding of the system in order to successfully implement and achieve the benefits of on-line optimization. Most process models are incomplete because they use principle model with many assumptions. So the alternate approach using artificial intelligence techniques, such as Artificial Neural Networks (ANN). This statistical learning model should be able to handle highly-nonlinearity and carry out parameter tuning easily.

2. HYBRID MODELLING METHOD

The most common task of the statistical model such as neural network is to perform a mapping from an input space to an output space. It has shown that, under some mild assumption, can approximate any nonlinear continuous function arbitrarily accurately. These models have ability to 'learn' the frequently complex dynamic behavior of a physical system. Learning is the process where the network approximates the function mapping from system inputs to outputs, given a set of observations of its inputs and corresponding outputs. This is done by adjusting the statistical model's internal parameters, typically in such a way as to minimize the squared error between the network's outputs and the desired outputs. The ability to approximate unknown functions through presentation of their instances makes statistical models a useful and potentially powerful tool for modeling in engineering applications.

Neural networks have typically been used as 'black-box' tools, that is, no prior knowledge about the process was assumed; the goal was to develop a process model based only on observations of its input-output behavior. Modeling without using a priori knowledge has often proved successful and is the only possible method when no process knowledge is available. The ability of neural networks to learn nonparametric (structure-free) approximations to arbitrary functions is their strength, but it is also a weakness. A typical neural network involves hundreds of internal parameters, which can lead to 'overfitting'-fitting of the noise as well as the underlying function-and poor generalization. Furthermore, interpretation of such models is difficult.

Since redundancy (excess degree of freedom) may result in poor models, one approach has been to decrease the redundancy of the neural network model by developing algorithms that 'prune' the weights that have no significant effect on the network's performance. These methods either penalize model complexity or examine the sensitivity of the prediction error to the network's weights, and eliminate theses weights (connections) that least effect the fit. However, they do not address the issue of lack of internal model structure and do not use prior knowledge about the process being modeled.

In this work, a hybrid modeling approach has been developed. The approach attempts to maximize the value of domain-specific knowledge. The foundational model structure for the proposed approach is a hybrid or semi-parametric model. Figure 1 shows that this structure applies parametric 'default' model in parallel with a nonparametric support vector machine (SVM). These components are combined in series with a parametric output model. The default model accounts for parametric model behavior that holds in the absence of data. The SVM captures unknown functional relationship between the inputs and outputs. The output model enforces the explicit functional relationships that exit between the variables.

This approach offers significant advantages over a 'black-box' neural network modeling methodology. The hybrid model has internal structure which clearly determines the interaction among process variables and process parameters, and as a result is easier to analyze than standard neural networks. The first principles partial model specifies process variable interactions from physical considerations; the neural network complements this model by estimating unmeasured process parameters in such a way as to satisfy the first The first principles partial model specifies process variable interactions from physical considerations; the neural network complements this model by estimating unmeasured process parameters in such a way as to satisfy the first principles constraints; nonparametric estimation is needed since no knowledge is variable about theses parameters. Such structured models are expected to perform better than 'black-box' neural network models in process identification tasks, since generalization and extrapolation are confined only to the uncertain parts of the process while the basic model is always consistent with first principles and does not allow a physical variable interaction.

As shown the schematic representation (figure 1), the SVM component receives as inputs the process variables and provides an estimate of the current parameter values. The SVM's outputs serves as an input to the first principles component, which produces as output the values of the process variables at the end of each sampling time. The combination of these two building blocks yields a complete hybrid SVM model of the system.

However, for the hybrid model target outputs are not directly available. In this case, the known partial process model can be used to calculate a suitable error signal that can be used to update the SVM parameters. The observation error between the structured model's predictions and the actual state variable measurements can be 'back-propagated' through the known set of equations, essentially by using the partial model's Jacobian, and translated into the error signal for the SVM component. The intuition behind this is that the process parameters should be changed proportionally to their effect on the state variable predictions, multiplied by the observed error in the state predictions.

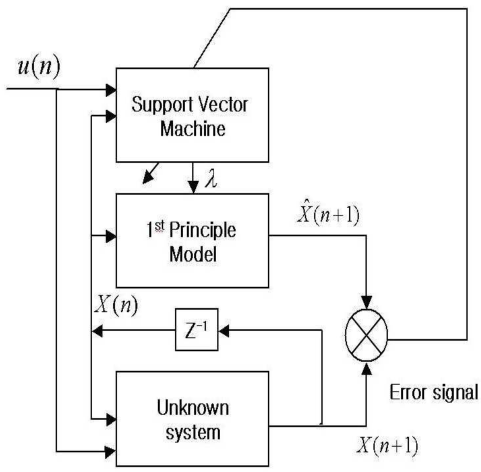

figure 1. Schematic representation of proposed model

3. EVOLUTIONARY IMPROVEMENT RTO

3.1. OUTLINE OF THE ALGORITHM

In order to overcome the drawbacks of not only conventional RTO frameworks but the RTE framework with direct search [6], an effective improvement algorithm has been proposed. The proposed improvement algorithm imitates the way that GA(Genetic Algorithm) finds the optimal solution with given constraints, then the proposed framework has the following merits: (1) the real-time framework with the proposed improvement algorithm is independent to the problem complexity because it does not use the objective function derivative or Hessian matrix and only use the objective function value itself, (2) can provide a global search method, and (3) its computation time increase just linearly with problem size.

As GA finds the optimum points through evolutionary procedure, so the proposed framework implements evolutionary improvement algorithm for real time optimization. The following outline summarizes how the algorithm works:

1) The algorithm begins by creating a random initial population within specific region around current process operating condition.

2) The algorithm then creates a sequence of new populations, or generations. At each step, the algorithm uses the individuals in the current generation to create the next generation. To create the new generation, the algorithm performs the following steps:

a. Scores each member of the current population by computing its fitness. The fitness value is calculated using pseudo steady-state at current individuals.

b. Scales the raw fitness scores to convert them into a more usable range of values.

c. Selects parents based on their fitness.

d. Produces children from the parents. Children are produced either by making random changes to a single parent (e.g. mutation) or by combining the vector entries of a pair of parents (e.g. crossover).

e. Replaces the current population with the children to form the next generation.

3) When the algorithm generates specific generation step size, the average value of current improved population is implemented as new adjusted set-point of process. The value is sent to control system.

The schematic diagram is shown in Figure 2. With this evolutionary improvement algorithm, the proposed RTO system is able to adapt external disturbance continuously.

3.2. CONSTRAINTS

Recently, several techniques have been proposed to handle constraints with genetic algorithm. The existing techniques can be roughly classified as follows: rejecting methods, repairing methods, and penalty methods. A rejecting method discards all infeasible chromosomes created throughout evolutionary process. It is simplest but least effective way to handle the problem. Repairing methods involve taking an infeasible chromosome and generating a feasible one through a repair procedure. For many combinatorial optimization problems, it is relatively easy to create a repair procedure.

The penalty technique is perhaps the most common technique used in genetic algorithms for constrained optimization problems. In essence, this technique transforms a constrained problem into an unconstrained problem by penalizing infeasible solutions, in which a penalty term is added to the objective function for any violation of constraints. In proposed framework, the *dynamic penalty* method was used. This method increases the penalty pressure along with growth of the evolutionary process.

$$f' = f - f_{avg} \times penalty$$
$$penalty = \exp\left(\max\left(\frac{\max\_gen}{\mathrm{ref}\_gen}, gen - \frac{\max\_gen}{\mathrm{ref}\_gen}\right)\right) \times \left(\sum_{i=1}^{no\_h} \Phi_{hi} + \sum_{i=1}^{no\_g} \Phi_{gi}\right) \quad (1)$$

where;

f: the value of scaled fitness function
f_{avg}: average value of f of current generation
f': fitness value with penalty term
gen : number of current generation
max_gen: number of termination generation
ref_gen: reference generation

The function, Φ_{hi} and Φ_{gi}, represent intensity of constraints violation, and it is defined as equation (2).

$$\Phi_{hi} = \begin{cases} \dfrac{h_i}{h_{i,\min}}, & \text{if } h_i < 0, h_{i,\min} \neq 0 \\ 0, & \text{if } (h_i < 0, h_{i,\min} = 0) \vee (h_i > 0, h_{i,\max} = 0) \\ \dfrac{h_i}{h_{i,\max}}, & \text{if } h_i > 0, h_{i,\max} \neq 0 \end{cases}$$
$$\Phi_{gi} = \begin{cases} 0, & \text{if } g_{i,\max} = 0 \\ \dfrac{\max(g_i, 0)}{g_{i,\max}}, & \text{if } g_{i,\max} \neq 0 \end{cases} \quad (2)$$

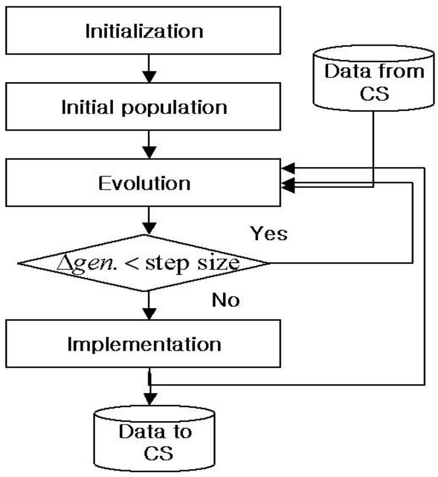

figure 2. Procedure of proposed framework

4. CASE STUDY

4.1. WILLIAMS-OTTO REACTOR

The process considered is the continuous-stirred tank reactor (CSTR) from the Williams-Otto plant (Williams and Otto, 1960) which is describe in figure 3. This case has been used for performance test of RTO system in some previous work.

Scenario

The objective is to maximize the instantaneous objective function (IOF), the corresponding profit is as follow:

$$profit(T_r, F_b) = 5554 \times F_r \cdot X_p + 125.9 \times F_r \cdot X_e - 370.3 \times F_a - 555.4 \times F_b \quad (3)$$

The main disturbance is the flow of stream *A*, F_a. The decision variables (e.g. set-points) are the reactor temperature,

T_r and the flow of stream B, F_b.

The profit surface for the plant equations is shown in figure 4. Starting from the optimum operating conditions, a step is simulated in F_a (from F_a =1.83 *kg/s* to F_a =1.0 *kg/s* at t = 300*s*). The RTO system will react after steady state detection, while the proposed framework is tuned to actuate every 5s (e.g. every 5 generation of genetic calculation).

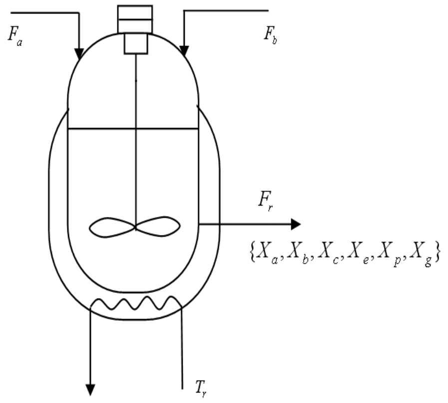

figure 3. Williams-Otto CSTR

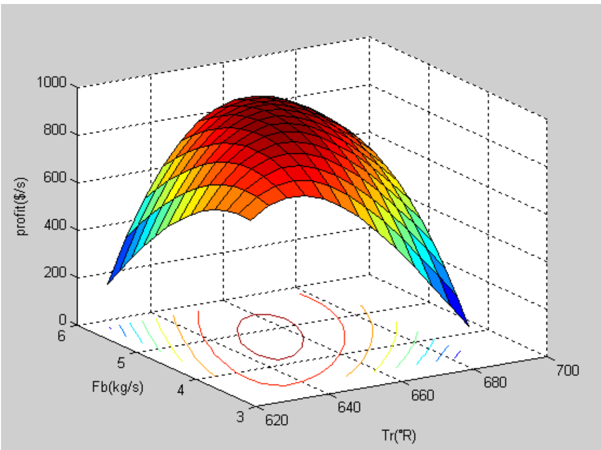

figure 4. Profit surface for plant equations (F_a = 1.83 *kg/s*)

Results

RTO response: Temporal values of F_a, T_r, F_b and the instantaneous objective function (*IOF*) using the RTO system are shown in the figure 5. It can be seen that at a time near to *t*=1500*s*, the RTO system implements the optimal set-points. The delay is mainly due to the steady state detection and also to the time consumed by the subsequent optimization procedure. The *IOF* (profit) graph in figure 5 clearly reflects this behavior and the improvement reached due to the RTO system.

RTE response: Temporal values of F_a, T_r, F_b and the instantaneous objective function (*IOF*) using the RTE system are shown in the figure 6. It can be seen that the RTE system does not wait for the steady state, but immediately reacts and changes set-points. The evolution of both decision variables is to proceed step by step to their optimum values (figure 6). In the case of T_r, RTE tries at first to increment its value, but later changes the direction towards the final optimum value. This mean that RTE will let T_r increase until a good enough product quality can be guaranteed according to the new F_a and F_b values, and only thereafter T_r will be allowed to reach its optimum value.

Profit graph in figure 6 shows comparative results of RTO, RTE and no optimizing action, with *IOF* as performance criterion. For both RTO and RTE final *IOF* approximates to 700$/s. It can be seen, that after the process comes to a true steady state, the IOF values for RTO eventually approximates to that of RTE as the time goes.

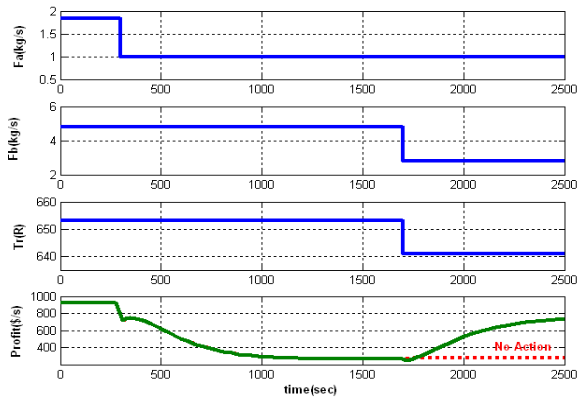

figure 5. Step disturbance and RTO response over the set-points (T_r, F_b and Objective function)

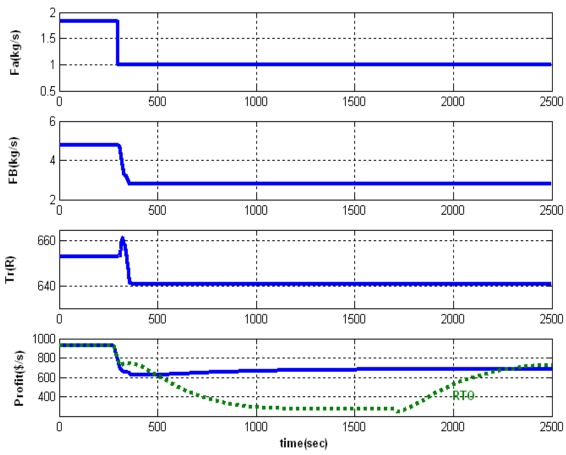

figure 6. Step disturbance and RTE response over the set-points (T_r , F_b and Objective function)

A more meaningful comparative plot (figure 7) is obtained using the mean objective function (*MOF*), which contemplates the history of the process and is defined as:

$$MOF(t) = \frac{\int_{t_0}^{t} IOF(\theta)d\theta}{(t-t_0)} \qquad (4)$$

Where t represents the current time and t_0 is a reference instant.

A deeper analysis of this graph reveals that during a few seconds the *MOF* values for RTO and RTE systems are lower than those for no optimizing action. This happens because RTO and RTE systems are based on steady state models, which only guarantee long-term improvement, when the steady state is achieved. Certainly, for the step disturbance case, both RTO and RTE reach steady state and thus the same IOF and MOF final values are expected when time tends to infinity. However, disturbances can arrive at any time, rather than infinity. Given that RTE produces faster improvement of MOF than RTO, the latter should always lead to better overall performance if further disturbances occur. This fact suggests testing the RTE performance when the disturbance consists in a continuous change, which is the subject of the following experiment.

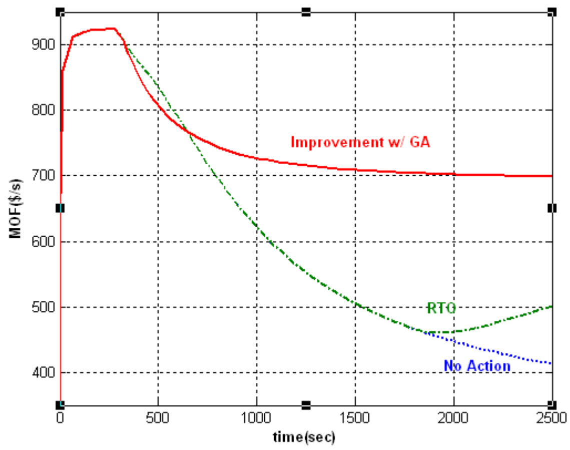

figure 7. MOF profile comparison between RTO, RTE and no RT system

5. CONCLUSIONS

The procedure of the proposed framework can be summarized as follows. When disturbance information is available, proposed framework keeps adjusting of set-point values, according to current disturbance measurements, present operating conditions and a steady state model. The adjusted set-points are obtained with genetic algorithm concept, which does not need the process to be at steady state. The performance of the proposed framework was successfully illustrated by means of real-time optimization example. The proposed method has the following advantage:

- The proposed method is very practical real-time optimization framework that is simple and fast enough to be applied in any kind of process despite its size.
- The proposed method can address the convergence to suboptimal because it uses the genetic algorithm.
- With continuous set-points correction, the proposed method is able to deal with higher frequency disturbance.
- The proposed method is much independent of control system performance

REFERENCES

Bhat, N. and McAvoy, T., "Information Retrieval From a Stripped Back-propagation Network", AIChE Meeting (1990)

Forbe, J. F. and Marlin, T. E., "Model accuracy for economic optimizing controllers: the bias update case", Ind. Eng. Chem. Res. 33, 1919-1929 (1994)

Forbe, J.F. et al., "Model accuracy requirements for optimizing plant operations", Com. Chem. Eng. 18, 497-510 (1994)

Karnin, E. D., "A Simple Procedure for Pruning Back-Propagation Trained Neural Networks", IEEE Trans. on Neural Networks, 1, 239 (1990)

Mavrovouniotis, M. L. and Chang, S., "Hierachical Neural Networks for Process Monitoring", Comp. Chem. Eng. 16(4), 347 (1992)

Miletic, I. And Marlin, T. E., "On-line statistical results analysis in real-time operations optimization", Ind. Eng. Chem. Res., 37, 3670-3684 (1998)

Mozer, M. C. and Smolensky, P., "Skeletonization: A Technique For Trimming The Fat Form A Network via Relevance Assessment, in Advances in Neural Information Processing", D. S. Toureszky, ed., Morgan Kaufmann, San Mateo, CA (1989)

Psichogios, D. C. and Ungar, L. H., "Direct and Indirect Model Based Control Using Artificial Neural Networks", Ind. Eng. Chem. Res. 30, 2564 (1991)

Sequeira, S. E. et al., "Real time evolution for on-line optimization of continuous processes", Ind. Eng. Chem. Res., 41(7), 1815-1825 (2002)