AIChE

Process Similarity and Developing New Process Models Through Migration

Junde Lu, Ke Yao, and Furong Gao
Dept. of Chemical Engineering, The Hong Kong University of Science and Technology,
Clearwater Bay, Kowloon, Hong Kong

DOI 10.1002/aic.11822
Published online July 13, 2009 in Wiley InterScience (www.interscience.wiley.com).

An industrial process may operate over a range of conditions to produce different grades of product. With a data-based model, as conditions change, a different process model must be developed. Adapting existing process models can allow using fewer experiments for the development of a new process model, resulting in a saving of time, cost, and effort. Process similarity is defined and classified based on process representation. A model migration strategy is proposed for one type of process similarity, family similarity, which involves developing a new process model by taking advantage of an existing base model, and process attribute information. A model predicting melt-flow-length in injection molding is developed and tested as an example and shown to give satisfactory results. © 2009 American Institute of Chemical Engineers *AIChE J,* 55: 2318–2328, 2009
Keywords: batch, base model, model migration, process similarity, family similarity, modeling

Introduction

Process and quality control and the optimization of industrial processes normally require a process model. A process model is a mathematical relationship mapping from an input space, X-variables, or process conditions to an output space, response variables Y, or quality properties. Generally, a process model can be developed based on (i) first-principles, (ii) using an empirical approach, or (iii) in the case of hybrid models, through their combination. A first-principles model based on sound process knowledge can facilitate extrapolation, but developing such models is often costly in terms of time and effort or even infeasible due to high dimensionality, complexity, and/or nonlinearity in the process. Empirical approaches, such as artificial neural networks (ANNs), fuzzy logic modeling (FLM), partial least squares (PLS), or support vector machines (SVMs), fit experimental input–output data to a relationship between process inputs and product quality.[1–4] They have poor extrapolation beyond the domain of the training data. To improve extrapolation, hybrid models, a combination of a first-principles model with a traditional data-driven method, have been developed for many chemical processes.[5,6]

Data-based modeling approaches are widely adopted in process model development for their cost efficiency and low level of prior process knowledge required. In industry, operating conditions are adjusted to control costs and maximize profit. Different product grades are produced by changing process recipes. Alternative processes might involve different batch sizes, operating conditions, or equipment configurations. For example, in injection molding, different products are produced with different grades of plastics and/or different molds. A modified recipe or process might invalidate an existing data-based process model if it is outside the range of the original data. To develop a new process model for the new process conditions, experimental design, data collection, and model training must often be repeated. This is normally time-consuming and it may be inefficient or uneconomic for processes involving expensive materials or which are expensive to operate.

Although the new process may be of different size and configuration, fed with different material, or operated under

Correspondence concerning this article should be addressed to F. Gao at kefgao@ust.hk

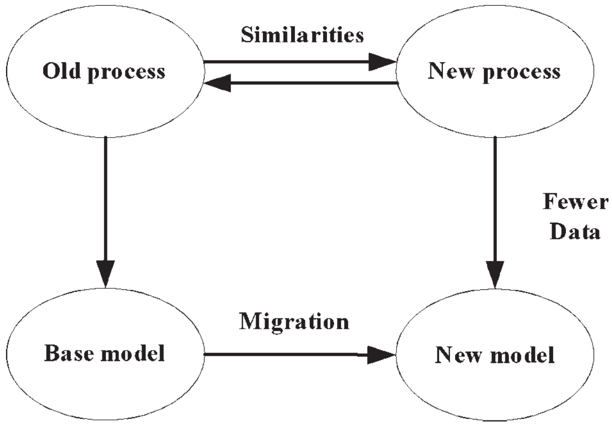


Figure 1. Development of new model.

different conditions, the underlying physical principles usually remain the same. There is hope therefore that proper use and extension of the existing process model can permit fewer experiments to develop a new model, resulting in a saving of time, cost, and effort. Such model migration needs to meet often conflicting requirements, for better generalization ability, but with fewer new experimental data at the same time. For clarity, the following discussion will define the process model describing the old process as the "base model" and the model to be developed for the new process as the "new model."

The concept of model migration has been treated recently by Lu and Gao.[7] As shown in Figure 1, the new model is developed through migration of the base model to the new process with fewer training data. Reduction of training data attributes to process similarity between the new process and old process. Lu and Gao discussed the key challenges and presented a simple demonstration case. Prior work on the subject of how to utilize existing process information in adapting an old process model to a new process has been rather limited. Multivariate calibration models have been used for extracting chemical information from spectroscopic signals. Spectral variations due to different instrument characteristics or environmental factors may render the original calibration model invalid for prediction in a new system. Feudale et al.[8] reviewed various methods of calibration model transfer to avoid time-consuming full recalibration by taking advantage of the existing model. Most calibration models are linear and one-dimensional, so such methods are not normally suitable for process modeling. Case-based reasoning (CBR) has been found to be an effective approach to find the solution to a new case by adapting solutions which solved similar cases in the past.[9] CBR is widely used in medical diagnosis, process control, planning, and product design, but CBR systems cannot propose a good solution in most situations without the assistance of domain experts or system managers.[10] On the other hand, in specific applications of practical interest, different CBR systems use different approaches which are mostly quantitative, descriptive, constructive, and experience-based. Another approach has been discussed by Jaeckle et al., who focused on analyzing historical process data to find a window of process operating conditions within which a product with a new specified quality can be produced.[11,12] The same concepts have been extended by using latent variables and joint-Y PLS models with historical process data on both sites.[13,14] The objective was to find the process conditions for a new plant instead of developing an entirely new model.

This study set out explicitly to develop a modeling strategy, called model migration, able to develop a new process model by taking advantage of the availability of a base model. It was hoped that such an approach would require fewer experimental data and make full-scale retraining unnecessary. The concept of process similarity was first proposed and defined. Similarity among complex processes was then classified into several types based on a particular process representation. Finally, for a given type of process similarity, a corresponding model migration technique was proposed. As an illustrated example, a soft-sensor model was developed for real time prediction of melt flow length in injection molding with a new mold, taking advantage of an existing model.

Process Similarity

Process representation

Generally, a particular process can be described either in terms of a finite set of its process attributes or properties, or in terms of the relationships among these properties. Attribute descriptions focus on processing equipment, materials, and conditions. This will be called an attribute-based representation. A relationship description is the viewpoint of a process model. It reflects relationships between a set of operating conditions and one or several output properties. This will be called a model-based representation. These two types of process representation can be used to categorize process similarity into several classes.

Attribute-based representation

Definition 1. A process P is represented by a set of process attributes and corresponding attribute values in an **attribute-based representation**, defined as:

$$P = \langle A, V \rangle \tag{1}$$

$A = [A_1, A_2, \cdots A_r]$ is a finite set of process attributes that reflect the nature of the process, including the feed material, processing equipment, environmental conditions, etc. $V = [V_1, V2, \cdots, V_r]$ are the corresponding attribute values; r is the number of process attributes. V_r could be a scalar or a vector with a numerical value or a descriptive value, depending on how each process attribute is described. Taking injection molding as an example, a typical injection molding process can be described by a set of process attributes, such as screw, mold, runner, and material, etc. Consequently, an injection molding process can be represented as:

$$A_{\text{Injection}} = [\text{Screw}, \text{Mold}, \text{Runner}, \text{Material}, \cdots] \tag{2}$$

For a particular injection molding process, which operates with a reciprocating screw, a mold with fan gate, a cold runner, and material of polyethylene (PE), etc., can be described by process attribute values:

$$V_{\text{Injection}} = [\text{Reciprocating, Screw, Mold with fan gate, Cold runner, PE}, \cdots] \quad (3)$$

Model-based representation

Definition 2. A process is represented by a set of process conditions, quality properties, and their relations in a **model-based representation**:

$$P = \langle X, Y, R \rangle \quad (4)$$

Here, X is a set of operating conditions or input variables; Y is a set of quality variables, response variables, or output variables; R is relationships between X and Y governed by process principles. Generally, the R can be developed through any modeling approach, including first-principles, empirical, or hybrid methods.

Given a particular process, any change in process attributes can trigger a change in its process model. Therefore, understanding process similarity in terms of process attribute change can help explain how a process model may change when process conditions change.

Process similarity

Definition 3. Two processes that have identical structures or process attribute sets are considered to belong to one *type* and are denoted as *structurally similar*; otherwise, the processes are *structurally dissimilar*. Obviously, the processes belonging to one type have equal numbers of elements in the sets, and the attributes in both sets correspond to each other.

Formulas (5) and (6) present an example of two injection molding processes. Both have the same processing components, such as a screw, a mold, a runner, etc., so these two processes are structurally similar.

$$A^1_{\text{Injection}} = \left[\text{Screw}^1, \text{Mold}^1, \text{Runner}^1, \text{Material}^1, \cdots\right] \quad (5)$$

$$A^2_{\text{Injection}} = \left[\text{Screw}^2, \text{Mold}^2, \text{Runner}^2, \text{Material}^2, \cdots\right] \quad (6)$$

where superscripts 1 and 2 represent process 1 and 2, respectively. Obviously, that the two processes be structurally similar is a precondition for model migration.

Definition 4. Two attributes of different processes are denoted as *pair attributes* if they correspond to each other and if they stand in the same relationship with other attributes in the attribute set. According to definition 4, attribute Screw^1 and Screw^2 in formulas (5) and (6) are pair attributes.

Attribute similarity

Definition 5. Two processes are said to have *attribute similarity* if all or some part of their pair attributes have similar values. Otherwise, when none of the values match, the processes are dissimilar. Attribute similarity can be divided into scale similarity, inclusive similarity, and family similarity.

Definition 6. If all or a part of the attribute values of process i scale directly from the corresponding pair attribute values of process j, the processes have *scale similarity*.

A typical example of scale similarity is where a process is scaled up from the laboratory to a pilot plant or full scale. One or several scale relationships link attribute values such as equipment size, production capacity, mold geometry, etc.

Definition 7. If all or a some of the attribute values describing process i are a subset of corresponding pair attribute values for process j, the processes are said to show *inclusive similarity*. For example, assume size u in processes i and j is identical:

$$V^i_u = V^j_u, \quad u \in [1, 2, \cdots r] \quad (7)$$

The remaining attributes, including for example size v, are different. Equation 8 represents a situation in which some of the attribute values describing process i are a subset of those describing process j.

$$V^i_v \subset V^j_v, \quad u \neq v, \quad v \in [1, 2, \cdots r] \quad (8)$$

Inclusive similarity is in nature set relations between a class and an element of it or between a whole and its parts.

Definition 8. If all or some of the pair attributes of two processes have values that belong to certain classes or families of a specific classification, the processes have *family similarity*. Formally,

$$V^i_u = V^j_u, \quad u \in [1, 2, \cdots r] \quad (9)$$

$$V^i_v \in V_c \quad \text{and} \quad V^j_v \in V_c, \quad v \neq u, \quad v \in [1, 2, \cdots r] \quad (10)$$

Equation 9 indicates that size u in processes i and j is identical. According to Eq. 10, size v in the two processes is not the same, but the two attribute values belong to the same family of V_c.

Family similarity is common in industrial processes. Taking injection molding again as an example, if two processes use high-density polyethylene (HDPE) and low-density polyethylene (LDPE), respectively, as feed material but the other processing conditions remain the same, the two processes can be described as having family similarity, as HDPE and LDPE both belong to the PE family and both are semicrystalline.

A set of classification laws is required to determine how to divide attribute values into classes or families. The constraints on process similarity will increase with increasing specificity of the classification laws, that is, as the families become narrower they will include fewer members (eligible attribute values).

Scale similarity, inclusive similarity, and family similarity are thus defined as the three basic attribute similarities. In reality, process similarity may be complex and a combination of the three basic types. For example, using different molds and feed materials in an injection molding process will lead to a new process with a combination of scale and family similarity when part weight is the product quality of interest.

Model-based similarity

A process model describes a relationship between a set of process conditions and quality properties. With a process model, similarity can be classified in terms of the input space and the input–output relationship.

Input space

A process model may be considered mathematically as a mapping between inputs X and outputs Y, written symbolically as:

$$Y = f(X); X \in \Re^n, Y \in \Re^m \tag{11}$$

where $\Re^n$ and $\Re^m$ represent an n-dimensional input space and an m-dimensional output space and $f()$ is the function describing their relationship. Each input variable $X = \{x_1, x_2, \ldots, x_n\}$ has its own operating range (a closed interval):

$$x_k \in [x_{k,\mathrm{l}}, x_{k,\mathrm{h}}] \tag{12}$$

defined by the process, where $x_{k,\mathrm{l}}$ and $x_{k,\mathrm{h}}$ are the lower and upper operating limits of process variable x_k. For simplicity, $[x_k]$ will represent the operating range of process variable x_k.

Definition 9. If the operating ranges of some or all of the process variables of two processes have the same range, the processes are said to have *identical inputs.*

In a practical case, the operating range of each process variable depends on equipment characteristics and material properties. Generally, similarity in operating ranges often implies inclusive similarity, shift similarity, and/or scale similarity of inputs.

Definition 10. If the operating ranges of some or all of the process variables of process i are included in or include the corresponding operating ranges of process j, the processes have *inclusive similarity of input*, described as follows:

$$[x_k^i] \subseteq [x_k^j] \quad \text{or} \quad [x_k^i] \supseteq [x_k^j] \quad k \in [1, 2, \cdots n] \tag{13}$$

where $[x_k^i]$ and $[x_k^j]$ represent the operating ranges of variable k in processes i and j, respectively.

Definition 11. If the operating ranges of some or all of the process variables in process i are shifted from the corresponding operating ranges in process j, the processes have *shift similarity of input*, described as follows:

$$[x_k^i] = [x_k^j] + c_k \quad k \in [1, 2, \cdots n] \tag{14}$$

or

$$x_k^i = x_k^j + c_k \quad k \in [1, 2, \cdots n] \tag{15}$$

where c_k is the shift in variable x_k.

Definition 12. If the operating ranges of some or all of the process variables in process i scale from the corresponding operating ranges in process j, the processes have *scale similarity of input*.

$$[x_k^i] = [x_k^j] \times b_k \quad k \in [1, 2, \cdots n] \tag{16}$$

or

$$x_k^i = x_k^j * b_k \quad k \in [1, 2, \cdots n] \tag{17}$$

where b_k is the scale factor of variable x_k.

There could exist correlations among the shift factor c_k and the scale factor b_k, depending on how the input variables correlate to each other and how similar the two processes are. This kind of correlation could be another indication to classify process similarity. Identical similarity, inclusive similarity, shift similarity, and scale similarity are four basic input similarities in input space. Practically, input similarity may often be a combination of the four basic similarities, forming a variety of complex input similarities.

Input–output relationships

Attribute and input similarities play important roles in defining and understanding process similarity, but they provide only some idea of how process characteristics change with changes in process attributes and input ranges. We cannot definitely know how a process model or input–output relationship changes in detail. All process attribute or operating range changes ultimately lead to input–output relationship changes. Therefore, an important issue is detecting input–output relationship changes and defining them with new data. Note, though, that the effort to detect process similarity will usually be less than that needed to develop a new model. Similarity in input–output relationships can generally be classified from the perspective of individual and overall factor effects.

Generally, in regression models, each input variable can be considered as a factor and each output variable is considered a response variable. A process model is multidimensional, and it is difficult to analyze process similarity in a multidimensional space, so input–output similarities must first be analyzed in terms of individual factors and responses. In other words, changes in the response of one input variable are analyzed with all the other variables fixed.

Definition 13. If, in the identical operating range of an input variable, the response curves of two process models have the same behavior but different magnitudes, the models are said to have *trend similarity* or *pattern similarity.*

Definition 14. If, when the operating range of an input variable in two process models is shifted, the response curves have the same behavior but different magnitudes, the models have *shifted trend similarity.*

Definition 15. If, when the range of an input variable scales between two process models, the response curves also scale, the models have *scaled trend similarity.*

All three kinds of trend similarity are common in similar processes. Generally, when processes follow the same physical principle, input variables have similar effects on response variables, indicating some sort of trend similarity.

Overall factor effects treat all input variable effects together. This discussion will consider only how the response of a process model changes with shifts and/or scale changes in some or all input variables.

Definition 16. Given process models

$$Y^i = f^i(X^i) \tag{18}$$

$$Y^j = f^j(X^j) \tag{19}$$

of processes i and j, if the inputs of process j are shifted and scaled from those of process i,

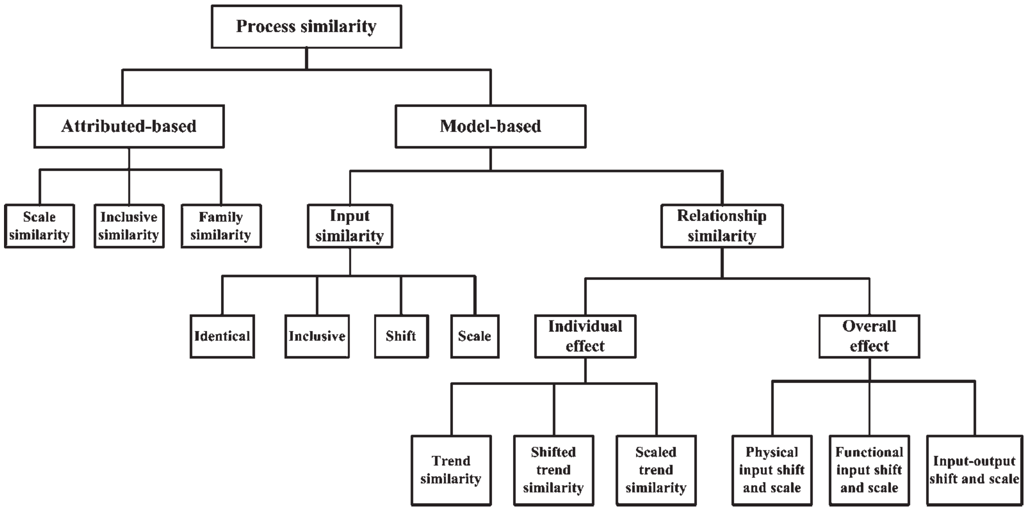


Figure 2. The hierarchical process similarity tree.

$$X^j = X^i * B + C \tag{20}$$

then,

$$Y^j = f^i(X^i * B + C) \tag{21}$$

and process j is a *physical input shift and scale* of process i. Vectors B and C represent the physical input scale and shift parameters.

Definition 17. If the shift and scale is

$$Y^j = f^i(X^i * B' + C') \tag{22}$$

process j is a *functional input shift and scale* of process i. Parameters B' and C' differ from the physical input shift and scale parameters B and C, and should be identified by new data.

Definition 18. If the shift and scale is described by

$$Y^j = D' * f^i(X^i * B' + C') + F' \tag{23}$$

process j is an *input–output shift and scale* of process i. Likewise, parameters B', C', D', and F' must be identified with new data. Lu and Gao[7] have offered a detailed analysis of how to identify these parameters.

The types of process similarity are summarized hierarchically in Figure 2. Of course, any practical case may display a combination of these basic similarities, forming a variety of complex process similarities.

Model Migration Strategy

Having categorized process similarity into several classes, this classification can be applied to assist the development of a general model migration strategy for each type of process similarity. Within each type of similarity, the appropriate migration strategy should allow developing new models by modifying existing ones. For example, scale similarity is common in industrial processes. Considering two processes, one of which is being scaled up from lab-scale, to pilot-scale or to full-scale; the other is injection molding with different mold sizes. Although these two groups of processes appear to be quite different, they share the common characteristics: change is a scale-up within the group. Thus, a model migration strategy for developing new models in each group may be available by modifying the general model migration strategy for scale similarity proposed by Lu and Gao.[7] For the inclusive similarity, Lu and Gao[15] discussed how to develop a new process model by aggregating outputs of a set of individual neural networks which are built based on data generated from the base model and new data. For a general case where process attributes values may be unknown, a method involving a procedure of six steps: information extraction from the base model, initial design of experiments, slope/bias correction (SBC) to the base model, outlier detection and assessment, further design of experiments, and development of the new model by combining local difference models and the corrected base model was proposed by Lu et al.[16] to deal with such case. For the model-based similarity, similar approaches may be used to explore the similarity for model migration, although specific algorithms were not given.

With the base model, there are two possibilities. One is that the base model is developed by the current user, in which then the base model structure is known to the user. This is the case that the base model is completely open to the user. In this case, certain model evolution strategies such as neural networks or fuzzy method may be used to develop new model.[17,18] However, the neural or fuzzy approach required a large number of training data. The other case is more general, in which a black-box model, for example, as a commercial package, is given to the user, so both the base model structure and parameters are not known. This black-box model maps only input–output relationship. This is a most challenge case, which is the one that we focus on in the article. So, we develop a method here for model migration regardless how the base model has been built. Of course, this poses the biggest challenge for the development of new model.

Consider a model migration strategy for family similarity. Before proposing a particular model migration method, some process assumptions are necessary. Assume first that two

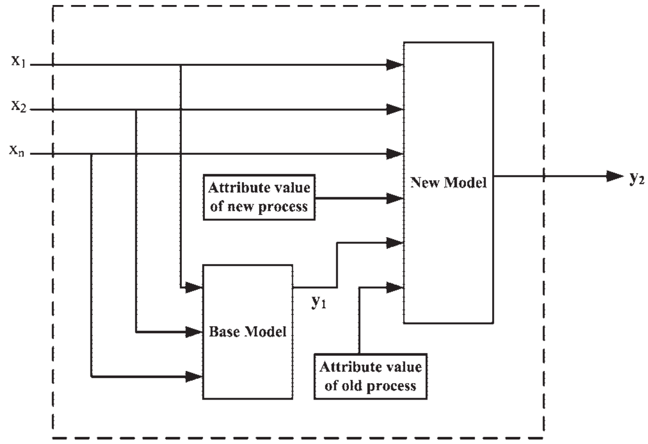


Figure 3. Schematic representation of a model migration strategy.

processes have identical inputs and operating ranges. Second, assume the process attribute values are known. Taking injection molding process as an example, we assume that the mold geometry and raw material properties are known.

Assume that the base model can be expressed as:

$$y_1 = f(x_1, x_2, ..., x_n) \tag{24}$$

where $[x_1, x_2, ..., x_n]$ is the input vector, f is a nonlinear function representing the old process, and y_1 is the output of the base model. As there is family similarity, process attributes belonging to the same family will engender similar behavior in the two processes and any new model should inherit behavior similar to that of the base model. On the other hand, changes in process attributes will lead to changes in the process model; any new model must reflect these changes. Therefore, a model migration strategy must meet the following two requirements:

(1) Inherit the behavior of the base model and

(2) Reflect process variations induced due to changes in process attributes.

To meet these requirements simultaneously, we propose a new modeling method incorporating outputs of the base model and process attribute values as additional inputs to the new model. In this method, the output of the base model is used as an additional input to the new model, as shown in Figure 3. The rationale is that because the new process is similar to the old process, the base model must be similar to the new model, so the output of the base model is a representation of the input data which should also be useful in the new model. In addition, to model process changes induced by changes in process attributes, attribute values of both processes can be incorporated into the new model to capture attribute changes. New experimental data with the new process can be used to train the resulting new model. The overall migration framework is shown in Figure 3, where y_2 represents the output of the new model.

The output of the base model, as an additional input to the new model, can represent the behavior of the base model, thereby reducing the amount of data needed for new model development. New data and incorporation of process attribute values appropriate to the new model can compensate for any bias introduced into the predictions by the presence of the base model results, and reflect process variations. Note that such a new model can be developed by any modeling approach, such as ANNs, FLM, PLS, and SVMs.

Case Study

Background introduction

To demonstrate the proposed migration method, a soft-sensor model was developed for online measurement of melt-flow-length (MFL) in an injection molding process. Injection molding, a cyclic process, operates sequentially in stages including filling, packing, holding, and cooling. During the filling stage, the screw moves forward and pushes melt flowing via the runner and gate into the mold until the cavity is completely filled. Melt development in the cavity during filling is important for the quality of the product. The melt-flow-length is defined as the distance that the melt-front has traveled in the mold from the gate during the filling phase. It is an important parameter reflecting the melt-flow status in the mold cavity, but it is difficult to measure in operation. A capacitive transducer has been developed which can monitor melt-flow-length,[19] but it is neither economical nor practical for all molds to be equipped with such a capacitive transducer. Instead, a soft-sensor measurement of the melt-flow-length during mold filling has been developed.[20] The soft sensor uses nine purpose-built mold inserts, and data from the inserts have been used to build a neural network model to predict melt-flow-length for any mold.[20] In this example, a new soft-sensor model predicting MLF for a new mold will be developed through model migration from a base model. A process operating with mold 1 (Figure 4) is considered as the old process and mold 2 (Figure 5) is used in the new process. To focus on the mold's influence on product quality, assume the same material (high-density polyethylene, HDPE), injection molding machine, and control strategy are used in these two processes. In addition, both mold geometries have gradually increasing then decreasing area. Therefore, according to definition of Eqs. 9 and 10, the process similarity between two processes can be viewed as family similarity.

Base model description

Chen et al. have explained that, for a given material and mold, the melt-flow-length at any time is mainly determined

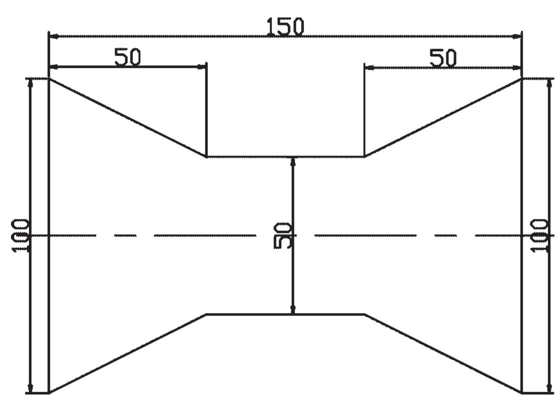


Figure 4. Mold geometry of the old process.

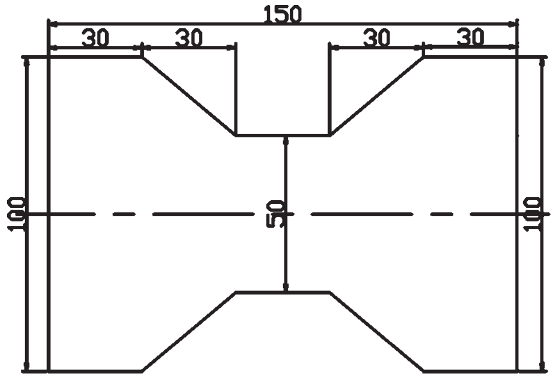


Figure 5. Mold geometry of the new process.

by nozzle pressure (NP), the screw displacement (SD), the injection velocity (IV), and the nozzle temperature (NT).[20] The relationship can be represented as:

$$\mathrm{MFL}_n = f(\mathrm{MFL}_{n-1}, \mathrm{NP}_n, \Delta\mathrm{NP}_n, \mathrm{SD}_n, \Delta\mathrm{SD}_n, \mathrm{IV}_n, \Delta\mathrm{IV}_n, \mathrm{NT}_n, \Delta\mathrm{NT}_n) \quad (25)$$

where the subscript n denotes the current time; $n - 1$ represents the last time interval; MFL_{n-1} is the preceding melt-flow-length; NP_n is the current NP representing the driving force for mold filling; $\Delta\mathrm{NP}_n$ represents the changes in the force required to generate the current melt-flow-length increment; SD_n is the current SD, which represents the total amount of melt in the mold; $\Delta\mathrm{SD}_n$ denotes the amount of melt entering the mold during the current time increment; IV_n is the current screw IV, which will affect directly the melt-front velocity and thus the melt-flow-length; $\Delta\mathrm{IV}_n$ is the IV change in the current increment; NT_n and $\Delta\mathrm{NT}_n$ are the current NT and its change, respectively. Although the melt-flow-length is a process variable related to time, the injection time is excluded as an explicit input, as it can be inferred from the SD and IV. The correlation between the melt-flow-length and the process variables can be treated as a ''black box'' input–output relation. In Eq. 25, the melt-flow-length of the nth sample, MFL_n depends not only on the current situation, but also on the previous status as reflected by the measurements at the last time interval, the preceding melt-flow-length, MFL_{n-1}.

To quantify the base model, a number of experiments were conducted using a Chen Hsong reciprocating screw injection-molding machine (model MK-III J88). With mold 1 (the old process), 11 different IV profiles were designed to cover as widely as possible practical molding conditions and generate training data for base model development. The IV profiles were constant profiles at 10, 15, 20, 25, and 30 mm/s; step change profiles with steps from 10 to 30 mm/s and 15 to 25 mm/s and two step-down profiles from 25 to 15 mm/s and 30 to 10 mm/s; and ramp profiles ramping up from 10 to 30 mm/s and ramping down from 30 to 10 mm/s. Function described by Eq. 25 is dynamic relation, a recurrent neural network therefore should be adopted for model development. However, direct training of such network is difficult, as there are interactions among the parameters over time. Chen et al. used algorithm to train the recurrent network but taking the form of a feedforward network.[20] In this article, the same training algorithm and neural network architecture with 15 neurons in the first layer and 20 neurons in the second (15) are adopted for base model development. Validation data showed that the predictions of the base model were in good agreement with the observations (Figure 6). The relative root-mean-square error (R-RMSE) was 0.115%, where R-RMSE is defined as:

$$\text{R-RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\left(\frac{Y_i-\hat{Y}_i}{Y_i}\right)^2}{N}} \quad (26)$$

where Y_i is the validation data, $\hat{Y}_i$ is the estimated value from base model, and N is number of validation data, respectively.

Model migration and results

As stated before, the old process and the new process are different only in terms of one process attribute (mold geometry), other process attributes, such as screw, runner, raw material, are assumed same. Furthermore, mold geometry has significant impact on prediction of MFL. Consequently, applying the model migration strategy shown in Figure 3, process attribute values (mold geometry information of two processes), and the output of the base model should be incorporated as additional inputs of any new model, as illustrated in Figure 7. In this example, two different molds are being considered, so their differing geometries need to be incorporated into the new model as additional inputs. A two-dimensional static matrix can be used to represent mold geometry. However, it is difficult to take such a static matrix as an input for model development. Moreover, the MFL is a process variable related to time, so MFL changes cannot be reflected by a static mold geometry matrix. Therefore, dynamic mold geometry information should be incorporated as inputs to the new model.

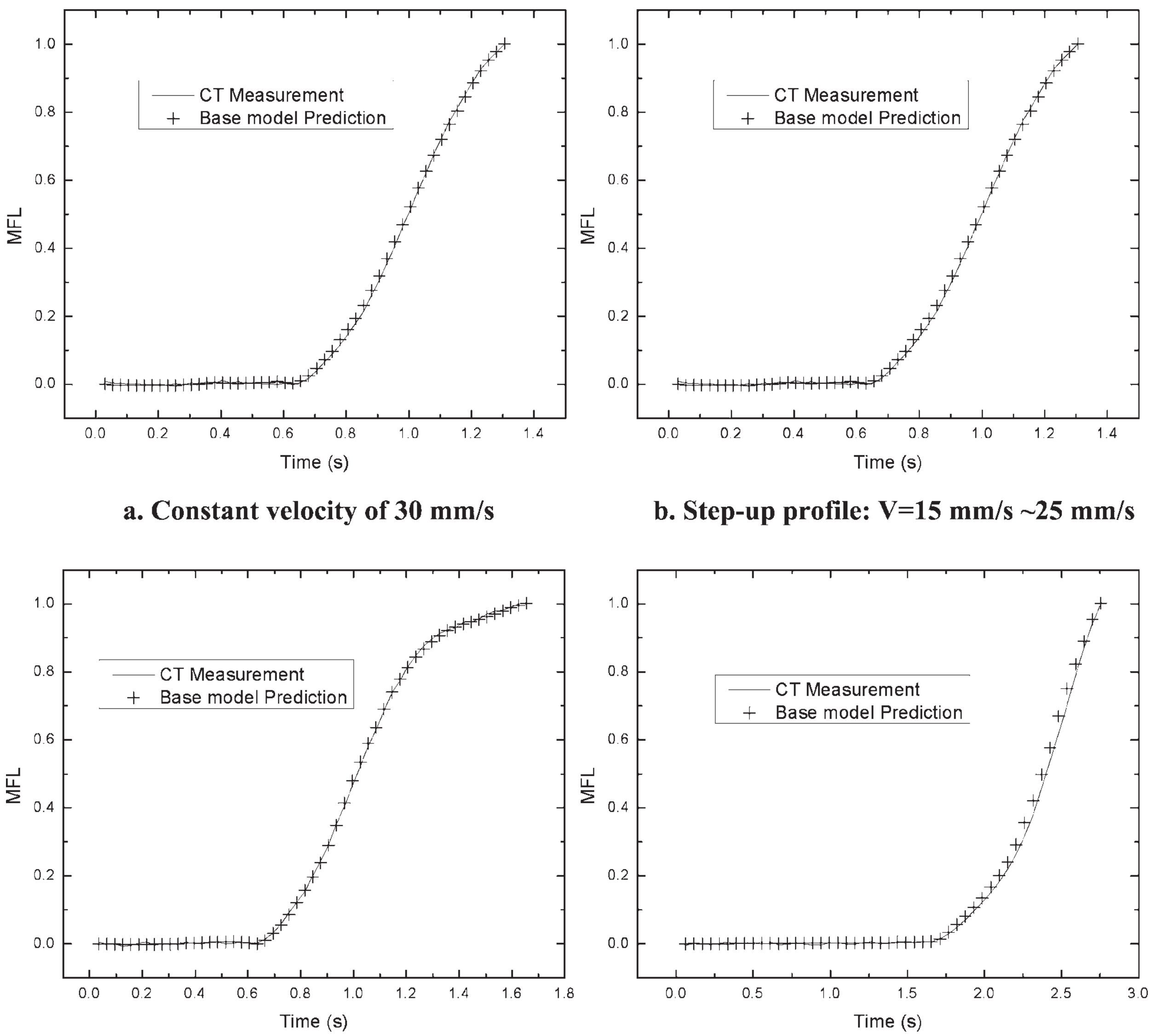
To represent mold geometry dynamically, imagining a moving line scanning the mold geometry along the flow direction. This line will intersect the edges of mold at points p_1 and p_2, as shown in Figure 8. The height (h) and slope (s) of the two points can be used to represent the mold geometry information at the current time. A series of those heights and slopes can represent the whole mold:

$$M_n = [h_{1,n}\ s_{1,n}\ h_{2,n}\ s_{2,n}] \quad (27)$$

where M_n is the current mold geometry information, represented by the height and slope information vector $[h_{1,n}\ s_{1,n}\ h_{2,n}\ s_{2,n}]$, which incorporate the current height and slope of points p_1 and p_2. The position of the moving line (or points p_1 and p_2) is determined by MFL_n. This breaks the static mold geometry into a series of measurements compatible with the time series function of Eq. 25. The representation can then be expanded to:

$$\mathrm{MFL}_n = f(\mathrm{MFL}_{n-1}, \mathrm{NP}_n, \Delta\mathrm{NP}_n, \mathrm{SD}_n, \Delta\mathrm{SD}_n, \mathrm{IV}_n, \Delta\mathrm{IV}_n, \mathrm{NT}_n, \Delta\mathrm{NT}_n, \mathrm{MFL}_n(y_1), h_{1,n}, s_{1,n}, h_{2,n}, s_{2,n}) \quad (28)$$

where $\mathrm{MFL}_n(y_1)$ represents the output of the base model. New experimental data collected using mold 2 are then required for training the new model with different IV profiles, such as constant profile (20 mm/s), step change profile (10–30 mm/s and 25–15 mm/s), and a ramp profile (30–10 mm/s).



a. Constant velocity of 30 mm/s

b. Step-up profile: V=15 mm/s ~25 mm/s

c. Step-down profile: V=30 mm/s ~10 mm/s

d. Ramp-up profile: V=10 mm/s ~30 mm/s

Figure 6. A comparison of measurements and base model predictions with constant velocity (a), step-up profile (b), step-down profile (c), ramp-up profile (d).

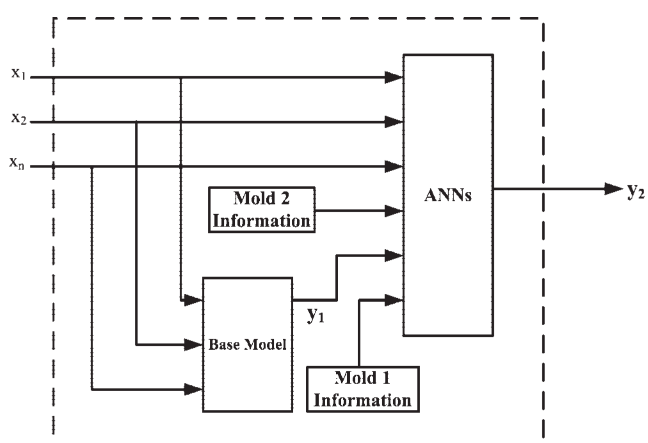


Figure 7. Schematic of model migration.

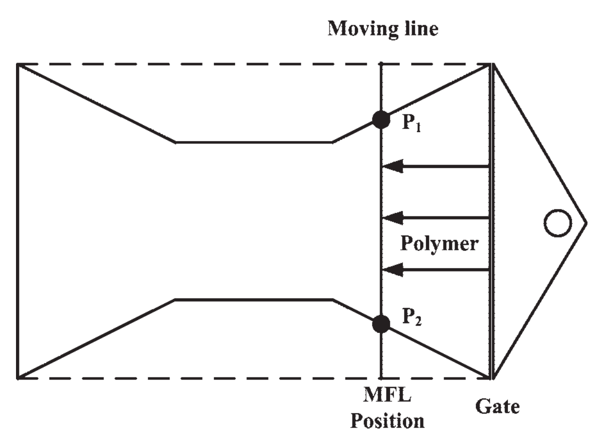


Figure 8. Schematic of scanning mold geometry.

The new model structure is chosen to be an ANN. A typical recurrent neural network has at least one feedback loop. For example, a recurrent neural network consists of a layer of neurons, some of which feed their outputs back as the inputs to neurons in previous layers, as depicted in Figure 9. Owing to its particular structure, the recurrent network can store information for future reference and thus be able to learn temporal as well as spatial patterns. For comparison, we also only use these new data to train a network to predict MFL. The prediction results from the new model, the trained network, and the base model are compared in Figures 10 and 11. The validation data were collected with the new mold and were not used for training. The solid line represents the actual MFL values, the stars represent the MFLs predicted by the new model, the circles are the MFLs predicted using the trained network, and the cross lines are the MFLs predicted by the base model. All the results suggest that the new model gave better prediction results than those of the network. As the network was trained using data from only four velocity profiles, its network predictions were poor, especially for the step-up and ramp-up profiles (Figures 10 and 11). The poor predictions from the base model indicate that there were significant changes in process behavior due to the changes in mold geometry. Table 1 summarizes the prediction results for the new process from the new model, the trained network, and the base model in terms of relative root-mean-square error (R-RMSE). Based on prediction results of Figures 10 and 11 and Table 1, the new model had better prediction results than either the network or the base model. With only four IV profiles for training, the new model had predictions similar to those of the base model calibrated with 11 IV profiles. This is a large reduction in training data without loss of prediction ability. The good results can be attributed to the new model's ability of taking advantage of the existing base model and incorporating the new mold geometry information effectively.

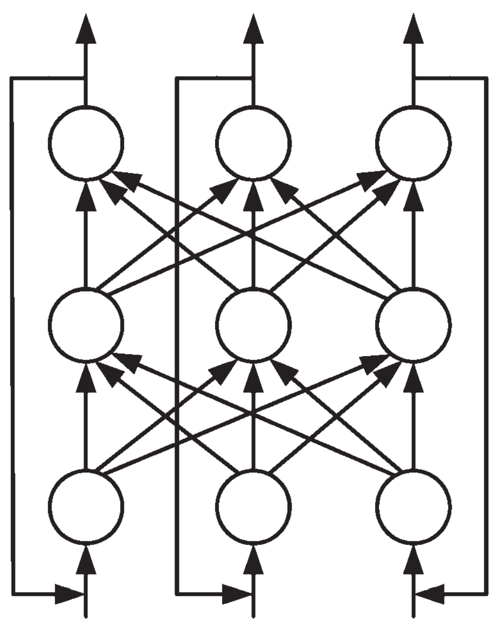
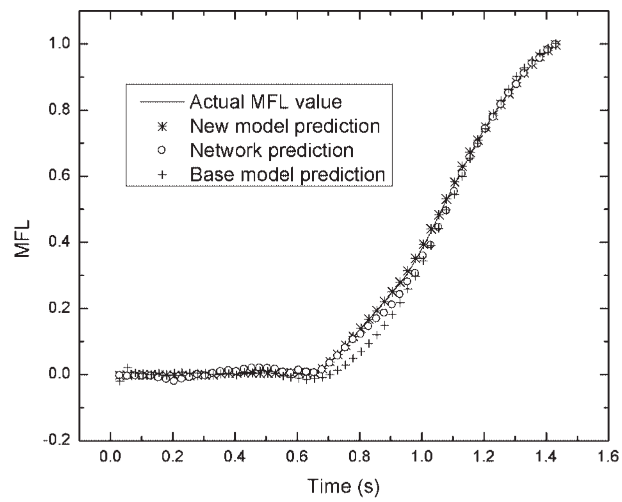
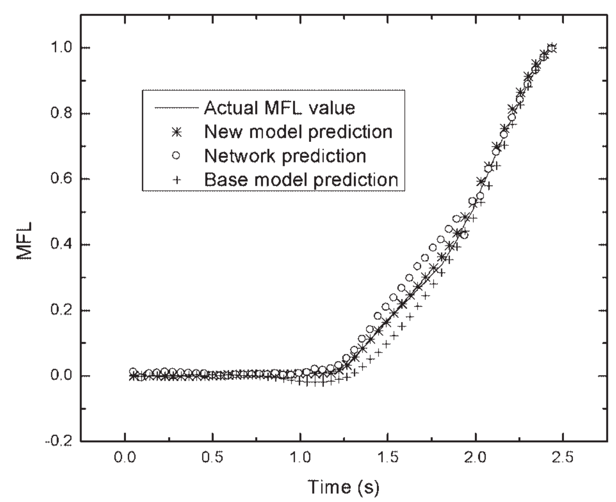


Figure 9. The architecture of recurrent neural networks.



a. Constant velocity of 30 mm/s

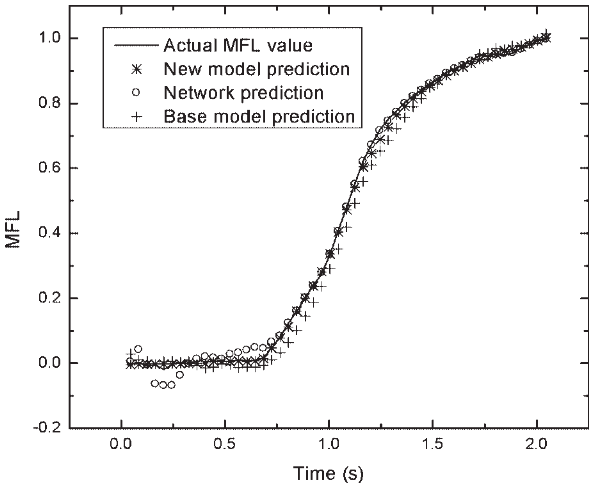


b. Step-up profile: V=15 mm/s ~25 mm/s

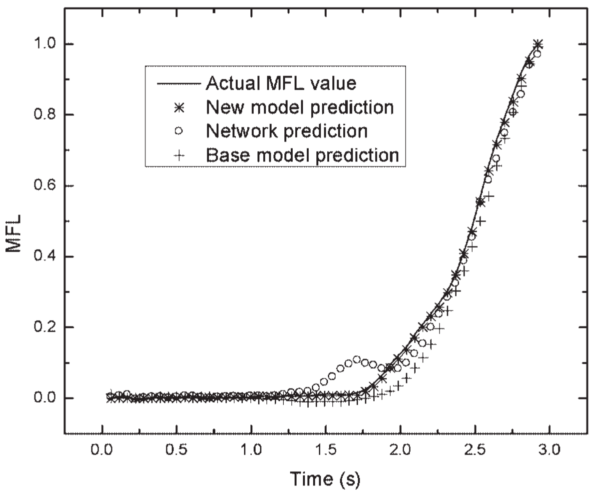
Figure 10. A comparison of measurements and the predictions from the new model, trained network and base model with constant velocity (a) and step-up profile (b).

Conclusions

Process similarity has been classified into several types. For a particular type, a model migration strategy has been proposed to incorporate the output of a base model and process attribute values as inputs to a new model. To demonstrate the proposed strategy, a model predicting melt-flow-length in injection molding was developed taking advantage of family similarity, data on an existing model and old and new mold geometry information. The prediction results showed the proposed migration method to be effective.



a. Step-down profile: V=30 mm/s ~10 mm/s



b. Ramp-up profile: V=10 mm/s ~30 mm/s

Figure 11. A comparison of measurements and the prediction from the new model, trained network and base model with step-down profile (a) and ramp-up profile (b).

Table 1. Prediction Results of Different Models with the New Process

Model	Training Data	R-RMSE
New model	4 velocity profiles	0.121%
Network	4 velocity profiles	0.283%
Base model	No training data	0.453%

Notation

P = process
A = process attribute
V = process attribute value
R = relationship between input and output variables
V_u^i = attributes which have same value in two processes
V_v^i = attributes which have similar value in two processes
V_c = the same family attributes
$\Re^n$ = n-dimensional input space
$\Re^m$ = m-dimensional output space
$f()$ = the function describing relationship between input and output variables
$x_{k,l}$ = the lower operating limits of process variable x_k.
$x_{k,h}$ = the upper operating limits of process variable x_k.
$[x_k]$ = the operating range of process variable x_k.
c_k = the shift in variable x_k
b_k = the scale factor of variable x_k.
Y = the output variables
X = the input variables
B = the physical input scale parameters
C = the physical input shift parameters.
B' = the identified input scale parameters
C' = the identified input shift parameters
D' = the identified scale parameters
F' = the identified shift parameters
MFL = melt flow length
R-RMSE = relative root-mean-square error
Y_i = the ith validation data
$\hat{Y}_i$ = the ith estimated value from base model
N = the number of validation data
M_n = the current mold geometry information
h = height information mold geometry
s = slope information of mold geometry
p_1 = point 1
p_2 = point 2

Superscripts

1 = Process 1
2 = Process 2
i = Process i
j = Process j

Subscripts

u = Number of attributes which have same value in two processes
v = Number of attributes which have similar value in two processes
k = the kth process variable
l = lower operating limits
h = upper operating limits
n = the current time
$n-1$ = the last time interval
1 = the point p_1
2 = the point p_2

Acknowledgments

This work is supported in part by Hong Kong Research Grant council, project 613107 and China National 973 project, 2009CB320603.

Literature Cited

1. Lotti C, Ueki MM, Bretas RES. Prediction of the shrinkage of injection molded iPP plaques using artificial neural networks. *J Inject Mold Technol.* 2002;6:157–176.
2. Li E, Li J, Yu J. A genetic neural fuzzy system and its application in quality prediction in the injection process. *Chem Eng Commun.* 2004;191:335–355.
3. Lu N, Gao F. Stage-based process analysis and quality prediction for batch processes. *Ind Eng Chem Res.* 2005;44:3547–3555.
4. Lee DE, Song JH, Song SO, Yoon ES. Weighted support vector machine for quality estimation in the polymerization process. *Ind Eng Chem Res.* 2005;44:2101–2105.
5. Michael L, Thompson MAK. Modeling chemical processes using prior knowledge and neural networks. *AIChE J.* 1994;40:1328–1340.
6. van Lith PF, Betlem BHL, Roffel B. Combining prior knowledge with data driven modeling of a batch distillation column including start-up. *Comput Chem Eng.* 2003;27:1021–1030.
7. Lu J, Gao F. Process modeling based on process similarity. *Ind Eng Chem Res.* 2008;47:1967–1974.
8. Feudale RN, Woody NA, Tan H, Myles AJ, Brown SD, Ferre J. Transfer of multivariate calibration models: a review. *Chemom Intell Lab Syst.* 2002;64:181–192.

9. Kolodner JL. *Case-Based Reasoning*. San Mateo: Morgan Kaufmann, 1993.
10. Watson ID. *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco: Morgan Kaufmann, 1997.
11. Jaeckle CM, MacGregor JF. Product design through multivariate statistical analysis of process data. *AIChE J*. 1998;44:1105–1118.
12. Jaeckle CM, MacGregor JF. Industrial applications of product design through the inversion of latent variable models. *Chemom Intell Lab Syst*. 2000;50:199–210.
13. Jaeckle CM, MacGregor JF. Product transfer between plants using historical process data. *AIChE J*. 2000;46:1989–1997.
14. Garcia Munoz S, MacGregor JF, Kourti T. Product transfer between sites using Joint-Y PLS. *Chemom Intell Lab Syst*. 2005;79:101–114.
15. Lu J, Gao F. Model migration with inclusive similarity for development of a new process model. *Ind Eng Chem Res*. 2008;47:9508–9516.
16. Lu J, Yao Y, Gao F. Model migration for development of a new process model. *Ind Eng Chem Res*. 2009.
17. Angelov PP, Filev DP. An approach to online identification of Takagi-Sugeno fuzzy models. *IEEE Trans Syst Man Cybern Part B*. 2004;34:484–498.
18. Alexandridis A, Sarimveis H, Bafas G. A new algorithm for online structure and parameter adaptation of RBF networks. *Neural Networks*. 2003;16:1003–1017.
19. Chen X, Chen G, Gao F. Capacitive transducer for in-mold monitoring of injection molding. *Polym Eng Sci*. 2004;44:1571–1578.
20. Chen X, Gao F, Chen G. A soft-sensor development for melt-flow-length measurement during injection mold filling. *Mater Sci Eng*. 2004;384:245–254.

Manuscript received Sep. 26, 2008, and revision received Dec. 31, 2008.