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

FUZZY SOFT SENSORS FOR CHEMICAL AND OIL REFINING PROCESSES

Natalia Bakhtadze, Evgeny M. Maximov, Ravil T. Valiakhmetov

Institute of Control Sciences, 65, Profsoyuznaya st., Moscow,117997, Russia
Tel.: +7.095.334.9201 E-mail: bahfone@ipu.rssi.ru

Abstract: The paper offers an approach to soft sensors design for chemical and oil refining processes. The approach proposed is based on virtual models and associative search techniques. Takagi-Sugeno fuzzy model is applied in combination with production knowledgebase to compensate for the lack of lab data.

Keywords: Process identification, knowledgebase, virtual models, associative search models, soft sensors.

1. INTRODUCTION

Soft sensors (Albertos and Goodwin, 2002) are becoming increasingly popular in oil refining, chemical, petrochemical, and other process industries where product qualities cannot be measured directly. A soft sensor (SS) is a mathematical relationship between a certain product quality measured periodically by plant laboratory or by stream analyzer and one or more process variables, such as temperatures, pressures, flow rates, etc., measured directly at the process.

Plant laboratories and analytical services are typically not able to provide the operations with timely and consistent information on product qualities, because sampling, sample transportation, and analysis are usually long-term and laborious, and can be hardly executed more than once or twice per shift. This often results in either off-spec product manufacture or large giveaways and, hence, lost benefits in either case.

Automatic stream analyzers were believed to overcome the challenge, but their high ownership costs hamper their extensive dissemination. At the same time, SS demonstrate comparable accuracy in many applications, higher reliability, and by orders of magnitude lower costs, thus providing an attractive alternative to stream analyzers. At present, SS are being developed using ANN techniques, hybrid neuron technologies (genetic algorithms), adaptive algorithms used for real-time model adjustment, and other techniques.

The paper discusses an approach to SS development based on fuzzy virtual models (Chadeev, 2004) and adaptive associative search algorithms (Bakhtadze *et al*., 2007). These algorithms ensure quick adjustment to specific process even in case of significant nonlinearities. Moreover, the fuzzification of certain model parameters using process operator's expertise enables the compensation for the lack of lab data necessary for building a valid process model.

The developed SS software interacts both with process and workshop databases and with LIMS. The software is compatible with DCS from various vendors such as ABB, Foxboro, Honeywell, Yokogava, etc., with SACADA, such as GE Fanuc, AdAstra, Wonderware, etc., with realtime databases such as OSISoft PI System, and other industrial information systems.

2. NONLINEAR PREDICTION ALGORITHM BASED ON VIRTUAL MODELS DEVELOPMENT

An identification algorithm for complex nonlinear dynamic objects such as continuous and batch processes was presented in (Chadeev, 2004). The identification algorithm with continuous real-time self-tuning is based on *virtual models* design. The algorithm enables product quality adjustment in advising mode by statistical treatment of process and lab data.

At every time step, a new virtual model is created. To build a model for a specific time step, a temporary "ad hoc" database of historic and current process data is generated. After calculating the output forecast based on object's current state, the database is deleted without saving.

The linear dynamical prediction model looks as follows:

$$y_t = a_0 + \sum_{i=1}^{r} a_i y_{t-i} + \sum_{j=1}^{s} \sum_{p=1}^{P} b_{jk} x_{t-j,p} , \qquad (1)$$

where y_t is the object's output forecast at the t-th step, x_t is the input vector, r is the output memory depth, s is the input memory depth, P is the input vector length.

The original dynamic algorithm consists in the design of an approximating hypersurface of input vector space and the related one-dimensional outputs at every time step (see Figure 1). To build a virtual model for a specific time step, the points close in a manner to the current input vector are selected. The output value at the next step is further calculated using least squares method (LSM).

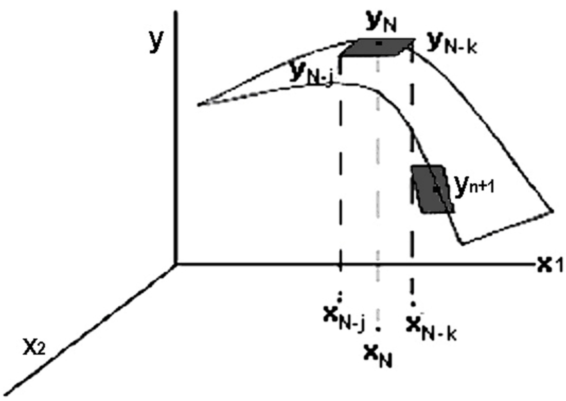

Fig. 1. Approximating hypersurface design

3. ASSOCIATIVE SEARCH TECHNIQUE FOR VIRTUAL MODELS DESIGN

High-speed approximating hypersurface design algorithms enabling the usage of fuzzy models for various process applications were offered in (Bakhtadze *et al*., 2007).

The following quantity

$$d_{t,t-j} = \sum_{p=1}^{P} \left| x_{tp} - x_{t-j,p} \right| , \; j = 1,\ldots,s , \qquad (2)$$

was introduced as distance (metric in $\Re^P$) between points of P-dimensional input space, where, generally, $s < t$, and x_{tp} are the components of the input vector at the current time step t.

Assume that for the current input vector $\boldsymbol{x}_t$:

$$\sum_{p=1}^{P} \left| x_{tp} \right| = d_t . \qquad (3)$$

To build an approximating hypersurface for $\boldsymbol{x}_t$, we select such vectors $\boldsymbol{x}_{t-j}$, $j = 1,\ldots,s$ from the input data archive that for a given D_t the following condition will hold:

$$d_{t,t-j} \le d_t + \sum_{p=1}^{P} \left| x_{t-j,p} \right| \le d_t + D_t , \; j = 1,\ldots,s . \quad (4)$$

The preliminary value of D_j is determined on the basis of process knowledge, e.g., product quality specifications. If the selected domain does not contain enough inputs for applying LSM, i.e., the corresponding SLAE has no solution, then the chosen points selection criterion can be slackened by increasing the threshold D_t.

To increase the speed of the virtual models-based algorithm, an approach is applied based on employing a model of process operator's associative thinking for predicting.

For modeling the *associative search* (Bakhtadze *et al*., 2007) procedure imitating intuitive prediction of process status by operator, we assume that the sets of process variable values, which are the components of an input vector, as well as the system outputs at previous time steps altogether create *a set of symptoms, making an image of the plant's output at the next step.*

The associative search process consists in the recovery of all symptoms describing the specific object based on its images. Denote the image initiating the associative search by P and the corresponding resulting image of the associative search by R. A pair of images (P,R) will be further called *association A* or $A(P,R)$. The set of all associations on the set of images forms the *memory* of the *knowledgebase* of the intelligent system.

At the system learning phase, an archive of images is created. In our case, a set of n input vectors selected form the process history according to the algorithm described in Section 1 will be considered as an image.

At the prediction stage, the input vector x_t will be considered as an initial image P^a of the associative search, while approximating hypersurface formed by the input vectors from the process history built with the help of the algorithm from Section 1 will be the final image R^a of the associative search. This means that to build a virtual model, one should select the existing hypersurfaces stored in the archive at the learning phase rather than individual vectors close to x_t. The selected hypersurface is an image of the current input vector which is used for output