Real-Time Qualitative Analysis of the Temporal Shapes of (Bio)process Variables

Konstantin B. Konstantinov and Toshiomi Yoshida
International Center of Cooperative Research in Biotechnology,
Faculty of Engineering, Osaka University, Osaka 565, Japan

One of the limitations of today's knowledge-based (KB) systems for diagnostics and supervision is a lack of adequate temporal reasoning mechanisms. Most of these systems are designed primarily to operate with the current values of the process variables and, sometimes, with their derivatives. Such simple capabilities, however, are not always sufficient to identify some complex dynamic phenomena, which in many cases leave their own unique "stamp" on the process behavior, expressed in the form of characteristic temporal shapes of the related variables. To detect and diagnose adequately the events of interest, the KB system should be able to reason about the temporal shapes of the process variables. Although during manual supervision process operators rely heavily on such characteristic shapes as reliable symptoms of underlying phenomena, their exploitation has not been considered seriously by the designers of KB control systems. We propose a generic methodology for qualitative analysis of the temporal shapes of continuous process variables designed to be embedded into a real-time KB environment. It is applicable to bioprocesses, as well as to other complex dynamic systems.

Introduction

During the last several years, the knowledge-based (KB) approach has found useful application in the control of fermentation processes. So far, KB systems have been successfully used in the control of various bioprocesses, such as wastewater treatment, production of yeast and bacterial biomass, biosynthesis of enzymes, antibiotics, and amino acids (Konstantinov and Yoshida, 1992). These applications have shown that by exploiting KB techniques, many of the shortcomings of conventional control methodology can be, at least partially, overcome. KB systems are capable of informal "physiological" interpretation of on-line measurements, monitoring and handling of complex phenomena which remained "invisible" to conventional systems, and flexible alteration of the control policy according to the real situation in the plant. It is expected that the KB approach will have growing importance in the field of bioprocesses control (Cooney et al., 1988; Aynsley et al., 1990).

Despite these positive results, however, KB control of fermentation processes is still a new, rather poorly developed, area of biotechnology, with many unsolved theoretical and practical problems. These problems are related to shortcomings in current (real-time) KB technology on the one hand, and to the extreme complexity of biological plants on the other. Recent studies have shown that KB systems used in bioprocess control must possess a number of specific characteristics that are nontrivial and are not supported by the KB shells available today (Konstantinov et al., 1992). Among these, of primary importance are the temporal reasoning capabilities which allow KB systems to account for the time dimension, significantly broadening their scope (see Figure 1). The availability of a consistent set of temporal reasoning features is considered to be a requisite in highly dynamic, complex environments (Moore et al., 1990; Laffey, 1991).

In the field of fermentation processes, a particular aspect of temporal reasoning that deserves special attention is the capability of the system to reason about the temporal behavior of the process variables over specific (usually recent) episodes of the process history. In many cases, the underlying phenomena that have to be identified by the KB system leave their

Correspondence concerning this article should be addressed to K. B. Konstantinov who is currently at the Department of Chemical Engineering, University of Delaware, Newark, DE 19716.

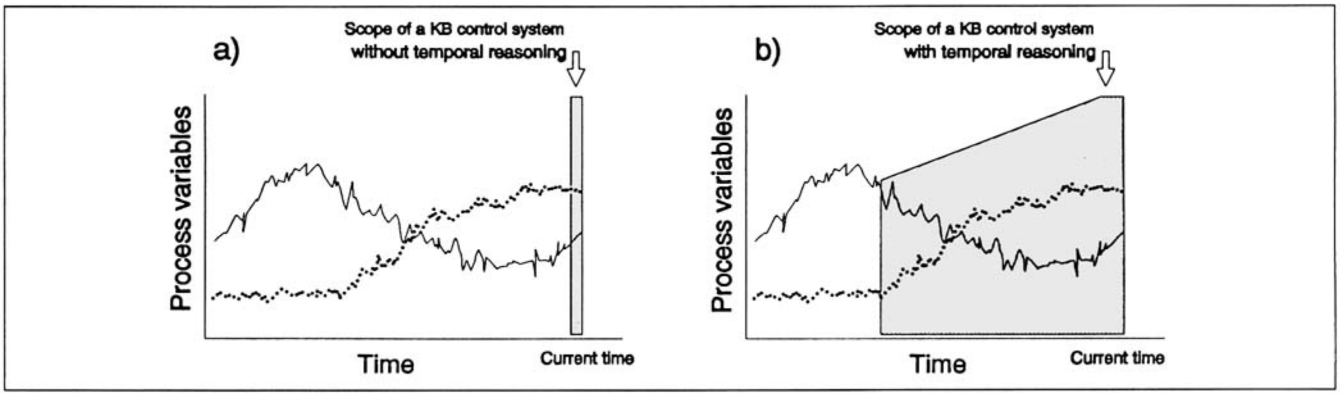

Figure 1. Scope of KB control systems with and without temporal reasoning.

KB systems provided with temporal reasoning capabilities can reason about past (and possibly future) events and phenomena, and their temporal relationships.

own unique "stamp" on the process history, expressed in the form of characteristic profiles of the related variables. To detect and diagnose adequately such events, KB systems should be able to reason about these profiles. This will result in a number of advantages: KB systems will be capable of more reliably capturing complex dynamic phenomena, extracting more information from the data available, and providing a more natural and compact form of knowledge representation. The absence of such a reasoning ability, on the other hand, reduces the situation-analyzing potential of the system; it also forces engineers to make unnatural, awkward and primitive schemes for representation of temporal dependencies that otherwise can be described in an elegant and straightforward fashion.

So far, these problems have been underestimated, even neglected. As a result, almost all KB systems suffer from serious deficiencies in this aspect of temporal reasoning. Nevertheless, there have been some publications in this area dealing with exploitation of the trends of process variables in decision-making (Perkins and Austin, 1990; Moore et al., 1990). Simple trends, however, are not always sufficient to identify complex phenomena, such as phase transitions, structural variations, and instrumentation faults, which result in intricate dynamic patterns of the process variables. Recently, Cheung and Stephanopoulos (1990a,b) have introduced a general framework for representation of process trends of higher complexity based on segmentalization and recursive simplification of the profiles of interest. This method is designed for cases when little or no knowledge of the process is available. Here, we will propose a flexible mechanism for the analysis and interpretation of temporal profiles, which is meant to enhance the performance of real-time KB systems. We are targeting processes where consistent amount of knowledge of the temporal behavior of their variables is already available, which indeed is a common case in process industry. The discussed approach is extended into a real-time software algorithm which acts as an independent front-end procedure, supplying the inference engine with information about the shapes of specified variables. Due to its efficiency, the algorithm can be used even with small computer systems. This is demonstrated by its incorporation into a KB system based on IBM-AT machine running under a real-time OS, which was successfully used on a day-to-day basis for the supervision and control of a complicated fermentation process.

In this article, we discuss the necessity of temporal shape analysis in KB systems from the viewpoint of bioprocess control. Also presented are the related real-time algorithm, as well as software implementation and embedding into a real-time KB control environment. Finally, we provide several examples of the application of the proposed methodology to the supervision of recombinant phenylalanine production.

Motivation

There are several grounds for the incorporation of reasoning about the temporal shapes of process variables into KB control systems. Although these considerations are discussed below in the light of bioprocess control, they are also relevant in other fields. Additionally, we try to explain the importance of a *qualitative* approach to the shape analysis problem.

Necessity of monitoring complex physiological phenomena

The main task of the KB system in bioprocess engineering is to act as an *intelligent monitor of physiological phenomena*. Indeed, the detection of more and more significant events is considered to be an essential goal of every real-time KB system (Edelmayer, 1990). The single chance to perform such a function is to efficiently utilize the information available on-line. Roughly, this information can be divided into two classes: current and historical. Today's KB systems are relatively good in working with the current data; they are weak in working with historical information. This may not be a problem in some cases, but in others it might be a serious limitation. Particularly demanding are continuous systems with complex dynamics. Restriction of the scope of the reasoning to current values and derivatives may lead to incorrect conclusions. Reasoning about complex phenomena, whose effect is spread dynamically over long time-intervals, requires richer representation, certainly involving the recent history of variables. The KB system should be equipped with a mechanism that allows it to locate and extract the effect of such events from the overall, often superimposed picture. As complex phenomena provoke characteristic shapes in the process variables, their detection and diagnosis might be more adequate if based on interpretation of such shapes over segments of historical data with appropriate length. KB systems that can look at the process history would be able not only to identify the current situation more reliably, but also to trace back in time how the process has actually arrived at this situation.

Lack of sensors

This is a well-known problem in the field of bioprocesses monitoring and control. To compensate for it, the on-line measurements should be completely utilized: that is, the control system must literally "squeeze" as much information as possible from the available variables and their histories. In such a case, the informationally rich temporal shapes of the variables should not be neglected. Typical examples are the cultivation of animal cells, as well as some production-scale microbial processes, where only a few sensors (for example, for pH and for dissolved oxygen concentration) are available, whose current readings do not provide sufficient information about the process state and thus cannot serve as a reliable basis for decision-making. The situation, however, can certainly be improved if the control system is able to analyze temporal profiles from the recent process history.

Reasoning about time-profiles improves knowledge representation

Inclusion of knowledge about the time profiles of the process variables would result in the development of rules with greater expressive power. Some intricate tangles of rules using only the current values and derivatives can be replaced by single and straightforward reasoning steps. For example (see Figure 2), at the moment t_2, (current time) one may be interested to see whether or not the expected shape of the variable $x_j(t)$ has appeared during the last few hours. If we have at our disposal only the conventional trending capabilities, it would be awkward to represent the corresponding logic by rules. It would be much easier and more elegant, *if it were possible to* perform the task by a single rule which processes the $x_j(t)$ profile over the period of interest at once, looking for the expected shape as an *integral formation*. The net effect is more compact rule bases with simplified logic and shorter reasoning chains.

Qualitative reasoning about temporal profiles is natural

Biotechnology practice shows that human operators perceive the fermentation process as a set of shapes of a few key variables and reason about the physiological activities of the cell culture based on qualitative interpretation of these shapes. Studies of casual reasoning have shown that a large part of expert knowledge consists of qualitative descriptions of continuous process variables (Forbus, 1984; Kuipers, 1989). According to Edelmayer (1990), the operators usually judge the current situation through visual evaluation of the pictorial representation of the process variables in a context-sensitive way. Their decisions are not based on elaborate calculations, but on a large repertoire of incremental catalogues of qualitative patterns. Even without knowing the exact numerical values, it is often possible to make surprisingly accurate decisions. This is why operators are not satisfied to have only the numerical values displayed on the computer screen; for adequate decision-making, graphic representation of the process history is required too.

Qualitative shape analysis can handle uncertainty

A typical feature of fermentation processes is their uncertainty, resulting in poor reproducibility. They are noisy and changeable, so that their variables never follow exactly the same patterns, and any attempt to compare and analyze time profiles in a strict quantitative fashion will be inconsistent (see Figure 3). A similarity between the real and expected shapes should be sought with reduced preciseness at the appropriate level of abstraction, preserving only the underlying shape features. Only by such qualitative representation, stripping all inessential quantitative details from the temporal profiles, will we be able to estimate adequately their similarity and, for

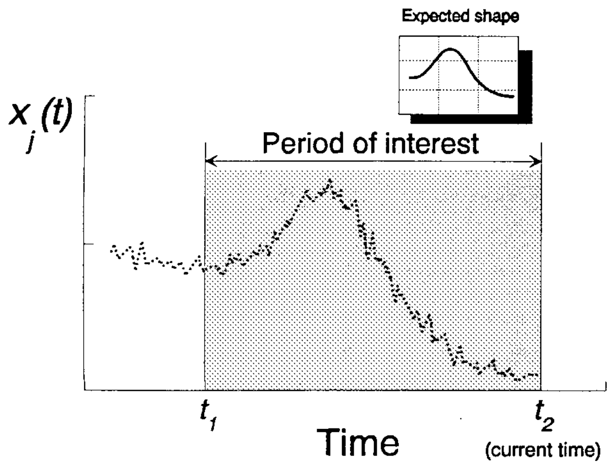

Figure 2. Reasoning about temporal profiles of process variables over specific time periods can be awkward, or even impossible, if only current derivatives are used.

It will be more efficient if the real and expected shapes are compared at once as integral formations.

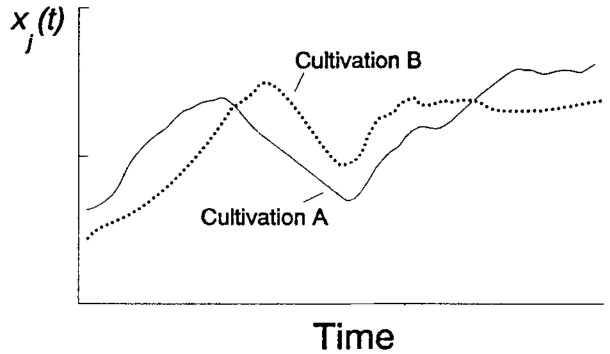

Figure 3. Because of the uncertainty of fermentation processes, time profiles of their variables do not coincide even if the processes are "physiologically" equivalent.

Only qualitative interpretation of the profiles of cultivations A and B can show their similarity.

example, to say that process A and process B, shown in Figure 3, are "physiologically" equivalent. As a qualitative description has the fundamental ability to be more *robust* than a quantitative one, the chances of developing more general knowledge bases are improved by using "qualitative" rules.

The method does not need to be expensive

The process history which is necessary for reasoning about the temporal profiles already exists in almost all control systems. It just needs to be properly used. As is shown below, efficient real-time algorithms can be developed to perform the related functions. They can be easily embedded into an existing real-time environment, without major restructuring of the KB system software.

Reasoning about Temporal Shapes As a Variation of Temporal Reasoning

Generally, temporal reasoning is defined as the capability of real-time KB systems to account for discrete or continuous phenomena, considering somehow their relationship with the time dimension (Perkins and Austin, 1990). Temporal reasoning has many different aspects, such as reasoning about the past, present or future process phenomena, temporal ordering of events, reasoning about the history of continuous process variables, and so on, some of which can be extremely useful in bioprocess monitoring and supervision. Different systems, however, require different forms of temporal logic, which can be implemented by specific mechanisms built into the KB system. Here, we will focus on one specific form—reasoning about the history of continuous process variables.

Let us assume that the knowledge in the KB system is represented as a set of standard *if-then* rules:

$$\begin{aligned}&IF\ (fact\ 1)\ and\ (fact\ 2)\ and\ (fact\ n),\\&THEN\\&(Conclusion)\end{aligned} \tag{1}$$

We are interested in developing a mechanism that can interpret rules containing facts of the following general form:

$$\begin{aligned}&IF\ldots(TimeInterval\ Variable\ ShapeDescriptor)\ldots,\\&THEN\\&(Conclusion)\end{aligned} \tag{2}$$

where *TimeInterval* defines a certain period $[t_l,t_2]$ from the process history, *Variable* is a particular process variable, and *ShapeDescriptor* specifies the expected pattern of the temporal profile. As the *Variable* might be influenced in parallel by more than one phenomena, the system should be able to discriminate between them based on considerations about their dynamics. This can be achieved by defining *TimeIntervals* with proper length. Here, *TimeInterval* introduces an appropriate time-scale for each phenomenon, which according to Cheung and Stephanopoulos (1990b) is "...the most important concept in analysis of process trends...because if we need to understand the process, we need to be able to describe the process signal at the scale of interest, which is the assumed scale of the physicochemical phenomena producing the signal." In other words, *TimeInterval* associates a specific time scale to each phenomenon, facilitating its detection from the usually superimposed signal.

According to the way of specifying the *TimeInterval*, the facts may take several forms. Usually, the right boundary t_2 coincides with the current time: that is, the reasoning is with respect to the most recent process history. The left time-boundary t_l can be specified explicitly (Figure 4a), for example:

$$IF\ (\underset{\text{TimeInterval}}{\underline{DuringTheLast2hr}}\ \underset{\text{Variable}}{\underline{SGR}}\ has\ been\ \underset{\text{ShapeDescriptor}}{\underline{Decreasing}}),\quad THEN\ (Conclusion)$$

or with respect to a certain event in the past (Figure 4b):

$$IF\ (\underset{\text{TimeInterval}}{\underline{SinceEnteringPhase3}}\ \underset{\text{Variable}}{\underline{RQ}}\ \underset{\text{ShapeDescriptor}}{\underline{>0.9}}),\quad THEN\ (Conclusion)$$

where *SGR* and *RQ* stand for the specific growth rate and respiratory quotient, respectively. In some cases the right boundary t_2 might not coincide with the current time. Instead, it can be specified with respect to a certain event in the past (Figures 4c and 4d).

According to the *ShapeDescriptor*, the facts can again be classified into several types. The *ShapeDescriptor* might specify patterns related to constraint violations, simple trends, or more complicated forms. Respective examples are:

$$IF\ (TimeInterval\ RQ\ \underset{\text{ShapeDescriptor}}{\underline{<0.95}}),\ THEN\ (Conclusion)$$

$$IF\ (TimeInterval\ SGR\ has\ been\ \underset{\text{ShapeDescriptor}}{\underline{Increasing}}),\quad THEN\ (Conclusion)$$

$$IF\ (TimeInterval\ OUR\ has\ \underset{\text{ShapeDescriptor}}{\underline{PassedOverMaximum}}),\quad THEN\ (Conclusion)$$

where *OUR* is the oxygen uptake rate of the cell culture. Each of the above types of facts requires a specific interpretation scheme, which must be built into the interference mechanism. In all cases, however, the interpretation of such facts should finally yield their degree of certainty (*dc*), which shows to what extent the given fact is true at the current moment. The *dc* is usually considered to be two-valued, with possible levels of 0 (false) and 1 (true). In more sophisticated systems, the *dc* can take any value from the interval [0, 1]. Whether discrete or continuous, the calculation of the *dc* requires the introduction of an appropriate procedure *P* that maps the fact being interpreted into its *dc*:

$$dc = P(fact) \tag{3}$$

The estimated *dc* is further used by the inference mechanism to calculate the certainty of the condition of the rule(s) where the fact is used. To this end, simple operators are usually applied, for example, min, max, product, or weighted sum. This problem has already been discussed in the literature (Konstantinov and Yoshida, 1989; Krijgsman et al., 1991) and will not be considered here. Our goal will be to construct a pro-

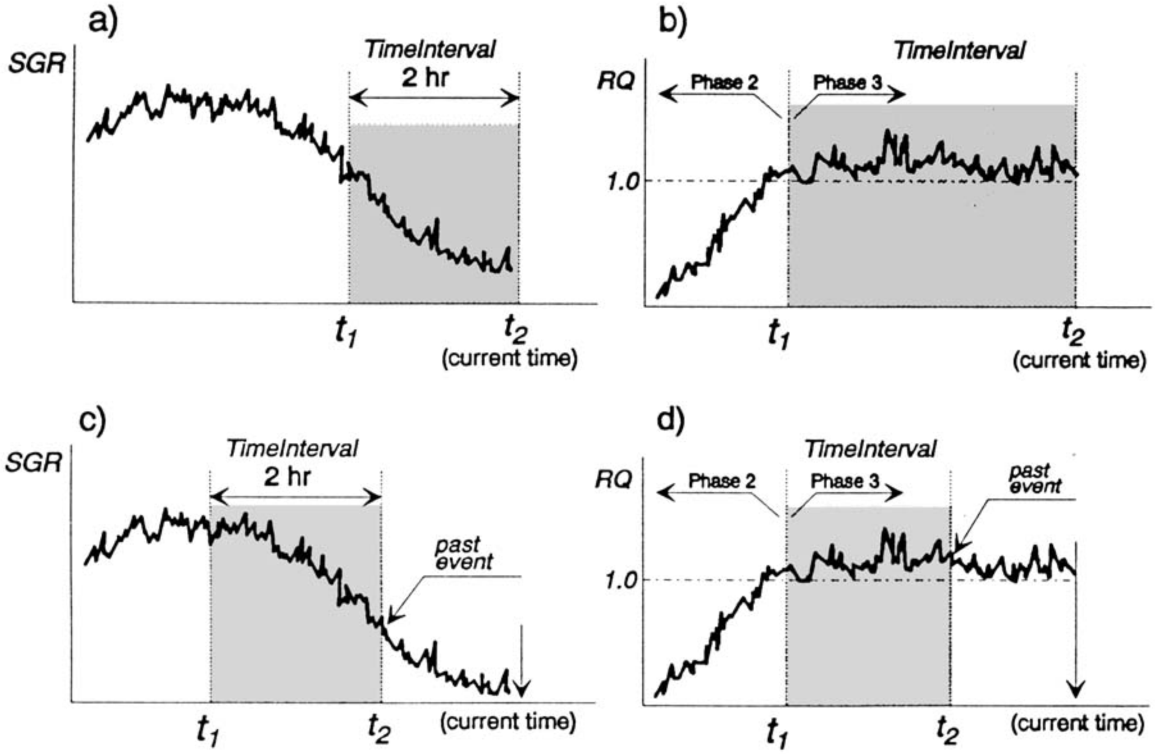

Figure 4. Different ways of *TimeInterval* specification.

a) t_2 coincides with the current time, t_1 is specified explicitly;
b) t_2 coincides with the current time, t_1 is specified in respect to a certain past event;
c) t_2 is specified in respect to a certain past event, t_1 is specified explicitly;
d) t_2 is specified in respect to a certain past event, t_1 is specified in respect to another past event.

cedure P, and the corresponding real-time algorithm, that can calculate the *dc* of facts of type (Eq. 2), where the *TimeInterval* is specified in any of the above-mentioned ways, and the *ShapeDescriptor* defines qualitatively the temporal shape of the given *Variable*.

Description of the Method

While calculation of the *dc* of facts related only to the current values or derivatives of the process variables is a relatively simple task (Konstantinov and Yoshida, 1991b), its evaluation in the case of facts of type (Eq. 2) concerned with the process history is much more complicated and does not have a universal solution. In this section, we will describe one possible approach that relies on evaluation of the similarity between the time-profile of a given process variable $x_j(t)$ over the time interval $[t_1,t_2]$, and the expected shape represented linguistically by the *ShapeDescriptor*. The *ShapeDescriptor* itself takes its value from a library of template shapes (for example, *"Increasing," "Decreasing," "IncreasingConvexly," "Decreasing-Concavely," "PassedOverMaximum,"* and so on) stored in the computer memory. Part of the current contents of our expandable shape library is shown in Figure 5. Such representation of the shapes reflects the so-called "relevance principle" of qualitative reasoning (Kuipers, 1986). According to this, qualitative reasoning about continuous scale objects requires the definition of a discrete set of symbols which must be *relevant* to the reasoning being performed.

It is apparent that the entries in the shape library do not represent a minimal set of primitive forms that in combination can construct more complicated profiles (Rengaswamy and Venkatasubramanian, 1992). While the minimal set would reduce the size of the library, it may require more sophisticated temporal logic for description of complex patterns. To avoid this, in addition to the set of primitives, we have included several "composite" shapes that can completely describe various dynamic phenomena. Thus, the detection of the latter can be achieved by a single fact, which simplifies considerably the construction of the rule base. In addition, the shape library is completely open, and if necessary, new application-specific entries can be defined. The incremental updating of the shape library is a natural process because with the accumulation of knowledge, new phenomena and their corresponding shapes are usually discovered.

The similarity between the library and the real-time profiles is searched for qualitatively by representing and comparing the real and library shapes in a symbolic form as ordered compositions of elementary shape features. Similar to the method of Cheung and Stephanopoulos (1990a,b), for such features we use the signs of the first and the second derivatives of the variables over the interval $[t_1,t_2]$ (Figure 6). They are selected as shape features for several reasons. First, these derivatives have clear visual representation, which can be easily perceived and described linguistically, while derivatives of higher order remain "invisible." Second, for profiles of normal complexity, their estimation is straightforward. As shown

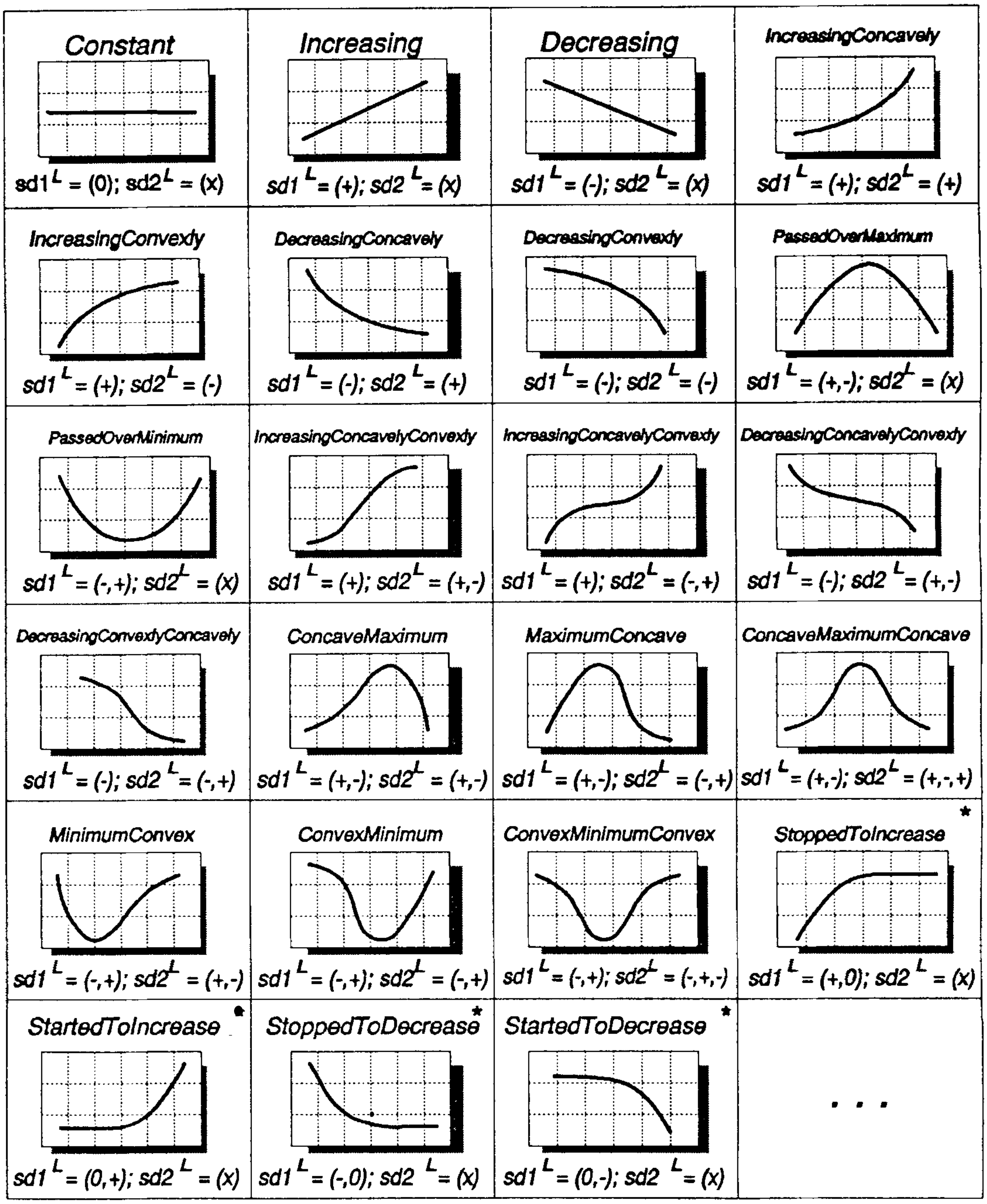

Figure 5. Elements of our expandable shape library.

Each entry is stored in the computer memory in a symbolic form represented by the strings $sd1^L$ and $sd2^L$ (see Eq. 4). The shapes denoted by an asterisk require an approximating function different from the one described in the text.

in Figure 5, the sequential combination of these features provides a rich set of profiles, which can cover a large number of real bioprocess situations. Formally, the extraction of the sequence of the derivative signs from the real-time profiles is described by the operators:

$$SD1[x_j(t)] = sd1 = (+,-,\ldots)$$
$$SD2[x_j(t)] = sd2 = (+,-,\ldots), \quad t\epsilon[t_1,t_2] \qquad (4)$$

which transform the continuous variable $x_j(t)$ into discrete symbolic strings $sd1$ and $sd2$, respectively. The qualitative shape of $x_j(t)$ is represented by the combination of these strings (Figure 6):

$$qshape[x_j(t)] = \{SD1[x_j(t)];\ SD2[x_j(t)]\}$$
$$= \{(+,-,\ldots);(+,-,\ldots)\}, \quad t\epsilon[t_1,t_2] \qquad (5)$$

Hence, two temporal shapes are considered qualitatively equivalent if their *qshapes* coincide. In accord with such representation, to assay the *dc* of the fact (Eq. 2), the analyzing procedure should extract in real-time $sd1$ and $sd2$ of $x_j(t)$ over $[t_1,t_2]$ and compare them with those of the library shape spec-

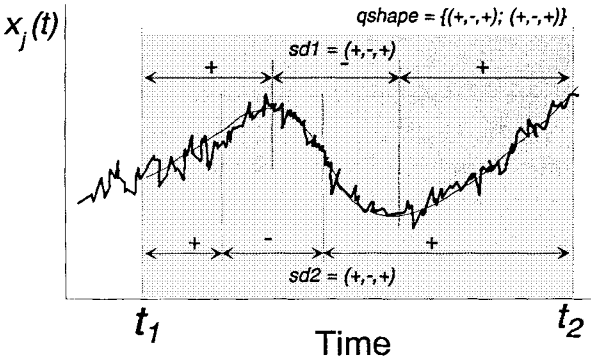

Figure 6. Example of transformation of the temporal profile of the variable $x_j(t)$ over the interval $[t_1,t_2]$ into the qualitative form *qshape* $[x_j(t)] = \{sd1;sd2\}$.

The thin line represents an approximation of $x_j(t)$ from which the feature strings *sd1* and *sd2* can be extracted.

ified by the *ShapeDescriptor*. The feature strings $sd1^L$ and $sd2^L$ of the library shapes are stored in the computer memory in the same symbolical form (see Figure 5). The *dc* represents a measure of the similarity of this qualitative representation of the real and library shapes. As shown in Figure 7, the procedure (Eq. 3) for calculation of *dc* is composed of three modules—approximation, transformation into qualitative form, and *dc* calculation—which are invoked sequentially, one after another.

Approximation under real-time constraints

The first step of the procedure consists of an approximation of the variable $x_j(t)$, whose history is stored in a circular buffer with sufficient length by a proper analytical function $x_j^*(t)$ over the time interval $[t_1,t_2]$. This is necessary to: 1) provide a convenient model for the subsequent analysis; 2) cut the noise; and 3) eliminate nonessential details from the real profile.

Among the functions that are candidates for such approximation, we have found the polynomials to be particularly convenient, because they lead to efficient approximation schemes. Below, we will consider this type of approximation of $x_j(t)$, expressed by the function $x_j^*(t)$:

$$x_j^*(t)=c_0t^0+c_1t^1+\ldots+c_mt^m,\quad t\epsilon[t_1,\ t_2]$$

Its order m may differ depending on the *ShapeDescriptor* and the complexity of the real profile. As one of our main concerns was to guarantee fast real-time response of the inference engine in heavily loaded multitasking environments even on small computers, it was essential to speed up the approximation, which is the most time-consuming part of the shape analyzing procedure. To this end, the well-known approximation equation (James et al., 1977)

$$F^T\cdot F\cdot\{c\}=F^T\cdot\{x\} \tag{6}$$

has been modified and optimized. Here, $\{c\}$ is the vector of the unknown coefficients c_k, $k=1,\ldots m$ and $\{x\}$ is the n-dimensional vector of the discrete values of $x_j(t)$ over the time interval $[t_1,t_2]$. The matrix F takes the form

$$F=\begin{vmatrix}[t_1+\ 0.\Delta T]^0 & [t_1+\ 0.\Delta T]^1 & \ldots & [t_1+\ 0.\Delta T]^m\\ [t_1+\ 1.\Delta T]^0 & [t_1+\ 1.\Delta T]^1 & \ldots & [t_1+\ 1.\Delta T]^m\\ \ldots\ldots\ldots & \ldots\ldots\ldots & \ldots & \ldots\ldots\ldots\\ [t_1+(n-1).\Delta T]^0 & [t_1+(n-1).\Delta T]^1 & \ldots & [t_1+(n-1).\Delta T]^m\end{vmatrix}$$

where $[t_1+(n-1).\Delta T]=t_2$, and $\Delta T=const$ is the sampling interval. The context of the problem in our particular case allows us to consider $t_1=0$ and $\Delta T=1$. Then, F can be simplified to the following numerical form:

$$F=\begin{vmatrix}0^0 & 0^1 & \ldots & 0^m\\ 1^0 & 1^1 & \ldots & 1^m\\ \ldots & \ldots & \ldots & \ldots\\ (n-1)^0 & (n-1)^1 & \ldots & (n-1)^m\end{vmatrix}$$

which can be calculated in advance and depends only on the

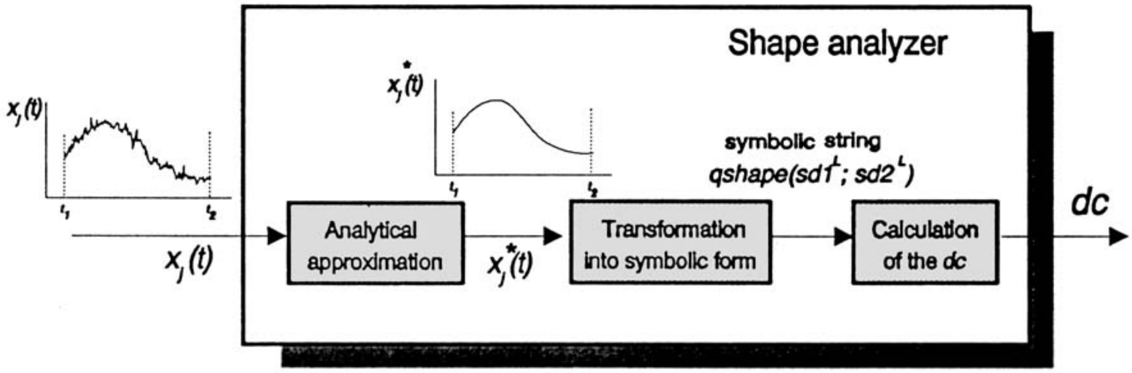

Figure 7. The shape-analyzing procedure (Eq. 3) consists of three modules: analytical approximation, transformation into symbolic form, and *dc* calculation.

polynomial order m and the length of the time interval $[t_1,t_2]$ expressed with the number of discrete samples n. Consequently, Eq. 6 can be reduced to:

$$\{c\} = (F^T \cdot F)^{-1} \cdot F^T \cdot \{x\} = Q \cdot \{x\} \tag{7}$$

where $Q = (F^T \cdot F)^{-1} \cdot F^T$ is a constant matrix for given m and n, which also can be calculated in advance. Practically, it is economical to prepare and store in the computer memory a set of matrices Q for several reasonable combinations of m and n, ready for use with different *TimeIntervals* (determining n) and *ShapeDescriptors* (determining m). In this way, the approximation is simplified to an extremely fast procedure—single multiplication (Eq. 7) of the matrix Q with the vector of historical data points x. In addition to the speed improvement, up to a certain number of combinations of m and n, the procedure uses less memory than its conventional alternative (Eq. 6).

After calculation of the unknown coefficients, the procedure checks the deviation of $x_j^*(t)$ from $x_j(t)$. If it exceeds a certain value, the approximation is automatically restarted with $m = m+1$. This recursive procedure continues until the deviation enters the allowed limits, or m reaches predefined maximum value.

Conversion of the shape into qualitative form

Once the analytical, noise-free model $x_j^*(t)$ is available, it is subjected to mathematical analysis. Beginning with this moment, the real curve $x_j(t)$ is not necessary any more, and all further operations refer to its approximation $x_j^*(t)$. The procedure extracts analytically from $x_j^*(t)$ the feature strings *sd1* and *sd2* composing the qualitative model (Eq. 5) of $x_j(t)$. As some simple patterns are represented adequately only by *sd1*, the second feature string *sd2* remains empty. It is used when the shape of interest is specified more detailedly.

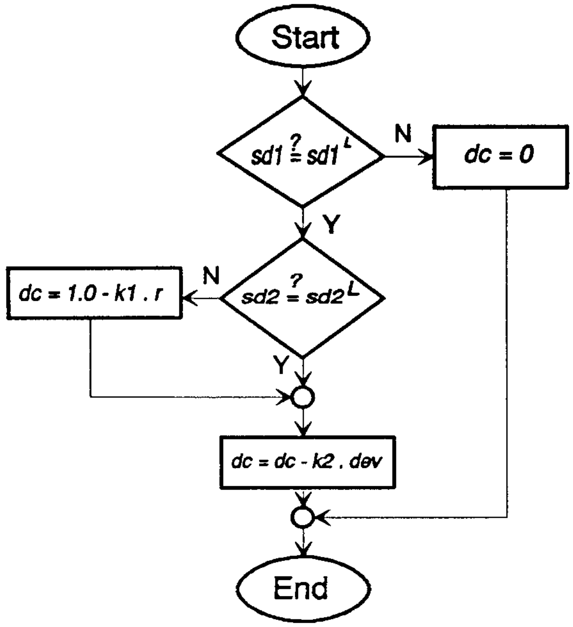

Figure 8. Algorithm for calculation of *dc*.

$k1$ and $k2$ are tuning parameters, r is the number of symbols in *sd2* and $sd2^L$ that do not match, and *dev* is the deviation of $x_j^*(t)$ from $x_j(t)$.

Evaluation of the similarity index dc

The final purpose is the calculation of the degree of certainty *dc* of the fact being interpreted. The *dc* estimation logic is given by the following formula (see also Figure 8):

$$dc = cpm\ 1(sd1, sd1^L) * \left(1.0 - k1 * \frac{cmp\ 2(sd2, sd2^L)}{R} - k2 * \frac{dev}{dev_{\max}}\right) \tag{8}$$

where

$$cmp\ 1(sd1, sd1^L) = \begin{vmatrix} 0 \text{ if } sd1 \neq sd1^L \\ 1 \text{ if } sd1 = sd1^L \end{vmatrix},$$

$cmp2(sd2, sd2^L)$ gives the number of the symbols in $sd2, sd2^L$ (from totally R) that do not match, $k1$ and $k2$ are weights, and $dev_{\max}$ is a constant representing the maximal allowed deviation. First, the generated sequences of features are compared with those of the library shapes, giving higher priority to the first derivative. If the strings *sd1* and $sd1^L$ do not coincide, *dc* will be zeroed automatically. Lack of coincidence between *sd2* and $sd2^L$ results in a reduction of the *dc*. The procedure also accounts for the deviation of $x_j^*(t)$ from $x_j(t)$, and lowers *dc* proportionally to the value of the deviation (see Figure 9). Only in the ideal case, when this value is zero, can *dc* reach its maximum of 1.0. The shape analyzing procedure terminates after passing the calculated *dc* to the inference engine.

Integration of the Procedure in a Real-time KB Environment

Control and software architectures of a KB control system

The structure of the KB system for the control of fermentation processes, which has been designed in our laboratory,

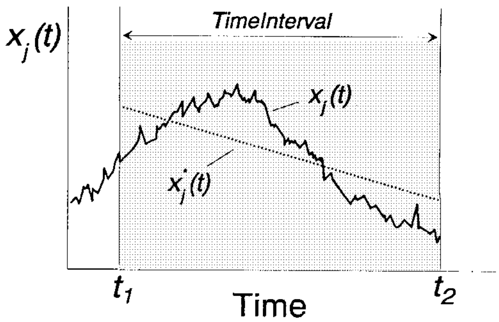

Figure 9. Approximation by low-order polynomials (corresponding to simple *ShapeDescriptors*, such as *"Increasing"* or *"Decreasing"*) may result in large deviations and inadequate decisions.

For example, if the deviation of $x_j^*(t)$ from $x_j(t)$ is not considered in the *dc* calculation, the above $x_j(t)$ profile will be wrongly interpreted as *"Decreasing."*

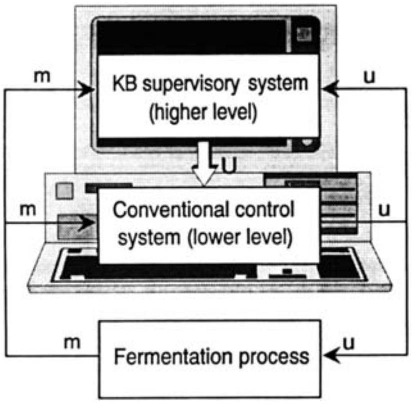

Figure 10. Control architecture of the KB system for control of fermentation processes.

m and u are measurement and control vectors, respectively; U is a high-level command for supervision of the conventional control module. Both parts of the system work in parallel on a single computer.

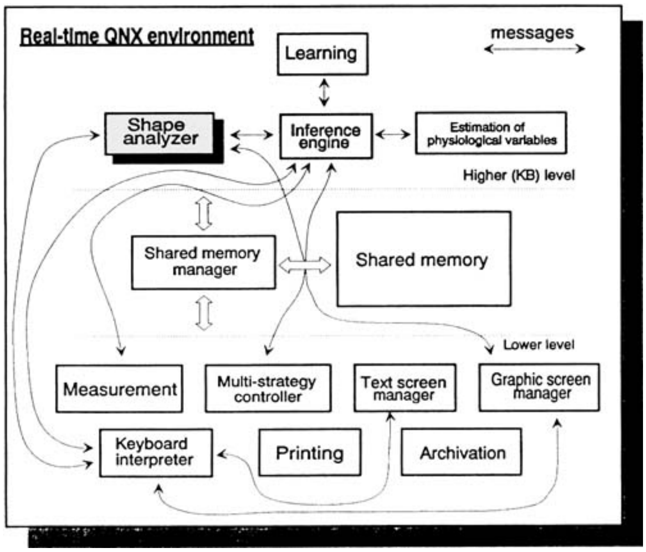

Figure 11. Software architecture of the KB control system.

All tasks work cooperatively in the real-time QNX environment, communicating through the shared memory, or by messages.

is shown in Figure 10. It is based on the concept of supervisory expert control (Verbruggen and Astrom, 1989), according to which the system is composed of two hierarchical levels—a KB one and a conventional one. The KB module is not explicitly involved in low-level control; it does not generate signals to the control plant, but helps the lower level to perform its job better. To this end, the higher level issues supervisory commands which tell the lower level when to do what (Smuts and MacLeod, 1989). The potential of this structure is rooted in the flexible combination of the traditional control approach with the KB methodology. It allows enhancement of the control system by the capability of intelligent decision-making based on informal interpretation of the complex behavior of the plant. This structure is capable of covering various problems that usually remain outside the scope of conventional systems for control of bioprocesses.

Each of the system levels looks at the biological plant in a different way. The higher level considers the fermentation process as a sequence of qualitative (physiological) states of the cell culture, that is, as a kind of discrete event system (Ramadge and Wonham, 1987). It realizes event-driven supervisory control, by solving the classical detection-diagnosis task and issuing supervisory commands to modify the lower level according to the current situation. The main functions of the higher level are the monitoring of physiological phenomena, identification of the physiological state of the cell culture, detection of instrumental failures, supervision of the conventional control part, and advanced communication with the user (Konstantinov et al., 1992). The temporal shape analysis procedure is meant to work at this system level. It can be useful particularly in the implementation of the first, second and third of the above functions.

The lower level, which actually represents a conventional control system, treats the process as a continuous quantitative system and utilizes conventional error-driven control logic. It performs standard algorithmic functions, such as measurement, filtration, control, and data acquisition. Among these, the control function, which is implemented as a multistrategy controller, possibly based on the multimodel control concept (Badr et al., 1989), deserves special attention. Each of its strategies is responsible for handling the plant in a particular qualitative state (Konstantinov and Yoshida, 1989, 1990).

The software architecture of the control system is shown in Figure 11. It consists of several tasks divided into two groups—KB and conventional—corresponding to system hierarchy levels. They run concurrently or with different priorities, and talk to each other using two distinct communication mechanisms—message passing and shared memory. The higher system level is made up of one main task (the inference engine) and several additional satellite tasks, which provide the former with the information required for the real-time inference. As the KB modules co-exist and work together with the conventional control algorithms, integrated in the same real-time single-computer environment, such schemes are called embedded. They are considered to be more advanced and very economical compared to the commonly used "interfaced" scheme, in which the KB part is implemented on a separated computer (Arzen, 1989; Konstantinov and Yoshida, 1991a; Odette, 1991).

The KB control system was developed to work in a modest hardware and software environment, which suits our main design criteria—compactness, modularity, low-price, and real-time performance. Presently, the system works on an IBM-AT 6-MHz machine under the QNX (Quantum, Canada) real-time multitasking OS. For the sake of speed improvements, all programs were written in C (ANSI C and C++).

Embedding of the shape analyzing procedure

Following the design philosophy of the overall system, the shape analyzing procedure has been implemented as a separated task (see Figure 12) running at a priority equal to that of the inference mechanism. This task has also been embedded into the same environment as a member of the higher level group. Indeed, this was done two years after the control system has been set in regular operation. The highly modular, message-passing organization of the control system allowed us to accomplish this easily and efficiently.

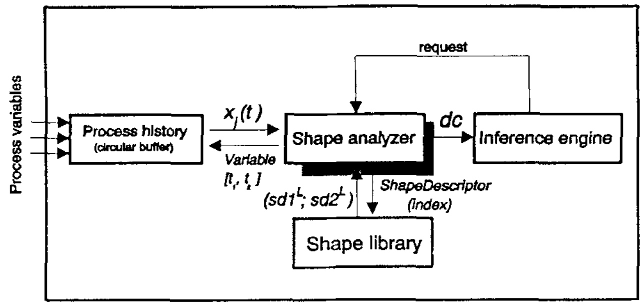

Figure 12. Communication between the shape analyzer and the inference engine.

As for all other tasks, the shape analyzer was designed as a "server" which remains in a blocked state most of its time and runs only after receiving a request (in the form of a message) for some job. Such a message is always sent by the inference engine when the reasoning process reaches a rule containing a fact of type (Eq. 2) requiring identification of the time-profile of a particular process variable. After identifying the fact encoded in the rule base, the inference engine prepares a message containing the values of the *TimeInterval* (the time-boundaries t_l and t_2), *Variable* (the variable index) and *ShapeDescriptor* (the index of the library shape), sends the message to the shape analyzer, and blocks while awaiting a reply. This message unblocks the later task; it unpacks the message, retrieves from the shared memory the historical data for the specified variable over the *TimeInterval*, and runs the procedure. After completing the calculations, the shape analyzer replies to the inference engine with the evaluated *dc*. This unblocks the inference engine, which resumes its work having been provided with the required information.

Although during run time all these activities of the control system remain invisible, we have enhanced the shape analyzer with a convenient graphic interface which allows tracking of what this task is currently doing or what it has done recently. Independent of the inference engine, the user himself can request on-line shape analysis of any piece of historical data, and observe the results and the evaluated *dc*. This capability facilitates the development, testing and debugging of rules containing facts of type (Eq. 2).

Application to the Supervision of Recombinant Amino Acid Production

The described system has been intensively exploited for the control of a complicated amino acid production process. This is a highly effective cultivation which uses a genetically engineered *Escherichia coli* strain to transform glucose into phenylalanine (Konstantinov et al., 1991). All experiments were conducted in a 14-L laboratory fermentor (Chemap, Switzerland), linked to the above-described control system through analog and serial lines. As this fermentation is accompanied by a large number of physiological phenomena that require timely and adequate intervention of the control system, one of our main goals was to provide its KB part with the capability to consistently monitor such events.

The organization of the knowledge base is shown in Figure 13. The inference cycle begins with an assessment of the cultivation stage. This is accomplished by a group of rules which

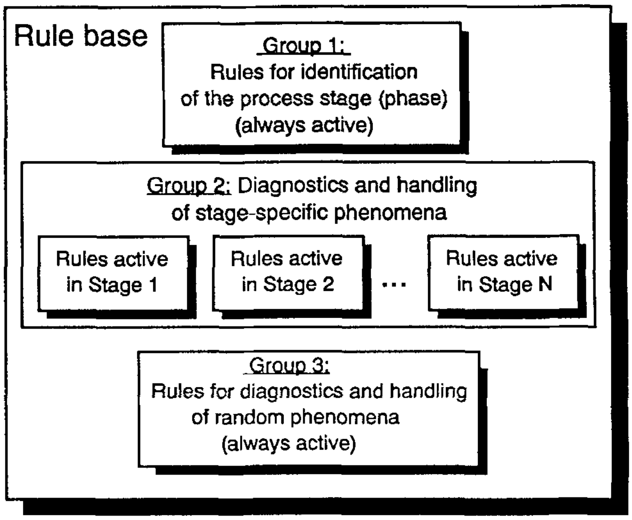

Figure 13. Structure of the rule base.

are always active. Because of their key position, these rules are of great importance for the system. Afterwards, the inference engine focuses its attention on a group of rules that are specific to the inferred process stage. They are meant to enrich the current picture by additional details, allowing for more accurate selection of the supervisory commands. The stage-specific groups of rules are also responsible for detection of other events, possibly of nonbiological origin that require intervention of the control system. There is one more group of rules meant to handle random phenomena, including faults in process equipment. Since they may appear in every stage, these rules are always active. The inference engine processes the events in the order of their detection. If several events happen simultaneously, they will be queued and processed sequentially.

Initially, the knowledge base was composed of about 35 rules that did not include facts about the process history, and we have tried to describe all situations only by using current values. Although the system could adequately detect some of the phenomena of interest, there were several situations when its reactions were nonconsistent, due to long delays or completely wrong decisions (Konstantinov and Yoshida, 1991b). A related problem was that in some cases it was difficult, even impossible, to express our knowledge for phenomena with complex dynamics in a reasonable form using such simple, time-less representation. To improve the situation, the KB system was enhanced with the capability of analyzing temporal shapes by addition of new facts to nearly 20% of the rules. Although our experimental work is still in progress, the results achieved so far clearly show improvement in the system performance in two main directions. First, the control system was capable of more reliable identification of some dynamic phenomena. Second, aided by the shape analyzing procedure, the system could detect some events that were outside its scope before. Another positive result is the elimination of some complex rulesets used before in certain process situations. Thus, addition of the new types of facts or rules did not result in a substantial net increase of the size of the knowledge base. Below, a few simple examples from the application of the discussed approach are given. They are only illustrative and certainly do not exhaust the capabilities of the method. To concentrate on the essential points, the exemplary rules have been stripped from the logic that is not directly related to the discussion.

Handling of process phase transition

A typical example of the application of the discussed procedure is the automatic detection and handling of the transition from the first to the second cultivation phase. During the first several hours, the culture grows under batch conditions and high glucose concentration. After depletion of the initial amount of glucose, the process enters its second phase of continuous feeding. To eliminate the negative effects of glucose starvation, it is essential to detect the glucose depletion as soon as possible (in terms of seconds) and to initiate continuous feeding immediately. As these operations are critical and sensitive, the control system has not only to carry them out, but also to confirm that the transition has been successfully accomplished.

The most informative variable, closely related to the underlying physiological processes, is the dissolved oxygen (*DO*) concentration. During the phase transition, it exhibits a typical characteristic shape, shown in Figure 14. Initially, the following rule was used to detect the glucose depletion, that is, the end of the batch phase:

IF (*DOincrement* > *5%*),

THEN (*Report: Glucose depletion*) *and* (*Activate glucose feeding*)

However, this rule, which is based only on the momentary *DO* increment, does not always work reliably due to unpredictable disturbances to which the *DO* is exposed. To compensate for this, it has been replaced by the rule:

IF (*DOincrement* > *5%*) *and* (*DuringTheLast30sec DO has been Increasing*),

THEN (*Report: Glucose depletion*) *and* (*Activate glucose feeding*)

which combines the quantitative fact about the current *DO* increment with the qualitative fact related to the recent *DO* history. This correction practically eliminated the errors observed previously.

To confirm that the transition has passed smoothly, a few minutes after activation of the continuous glucose feeding, the inference engine invokes another rule which checks for the expected *DO* shape:

IF (*SinceActivationOfTheGlucoseFeeding DO has PassedOverMaximum*),

THEN (*Report: Normal transition to Phase 2*)

If this pattern is not found, this signals that something is going wrong. Then, the system will call other rules, trying to identify the problem and isolate its cause.

Detection of foaming

During the first stage of the cultivation, usually between the 4th and the 8th hours, the culture produces foam, which should be eliminated by the addition of an antifoam agent. Since it is impossible to use a special foaming sensor, this event has to be detected using the available information. Again, we have found application of the shape-analyzing facility convenient. Detailed observation of the *DO* showed that during this stage of the process, its value decreases following convex pattern, due to the growing oxygen uptake potential of the culture. In the case of foaming, however, this pattern changes to concave (Figure 15), which can be efficiently detected by a proper shape-analyzing rule:

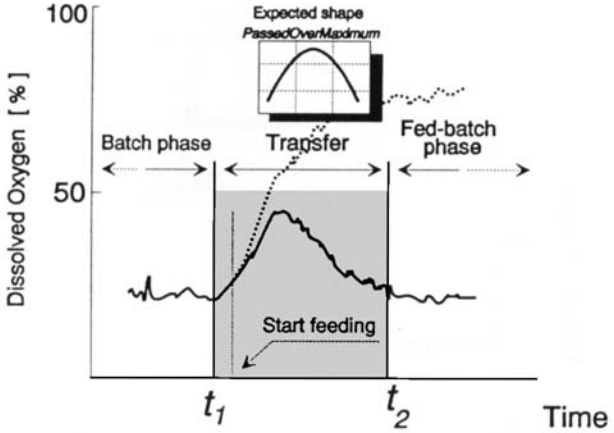

Figure 14. Typical time profile of *DO*(—) upon transfer from batch to continuous glucose-limited feeding and *DO* profile (---) of incorrect phase transfer.

IF (*DuringTheLast1hr DO has been DecreasingConcavely-Convexly*)

THEN (*Report: Foaming*) *and* (*Feed antifoam agent*)

It is worth noting that only a single fact is necessary to accomplish the task.

Automatic termination of the process

During the final process stage, cells progressively lose their oxidative potential and change to fermentative glucose metabolism. As a result, they start to excrete acetic acid, which accumulates in the broth. When this excretion becomes intensive, the process must be terminated because the product is not formed any more. The concentration of acetic acid is directly unmeasurable, but the phenomena can be monitored indirectly, using information about the specific variable $R_{a/g}$ (ratio of the ammonia feed rate to the glucose feed rate). During this period $R_{a/g}$ increases, following the characteristic concave profile (Figure 16). The process has to be terminated when its value exceeds a certain threshold. The rule used initially was based on this single static condition:

IF ($R_{a/g} > 0.03$),

THEN (*Report: Intensive acetic acid excretion*) *and* (*Terminate the process*)

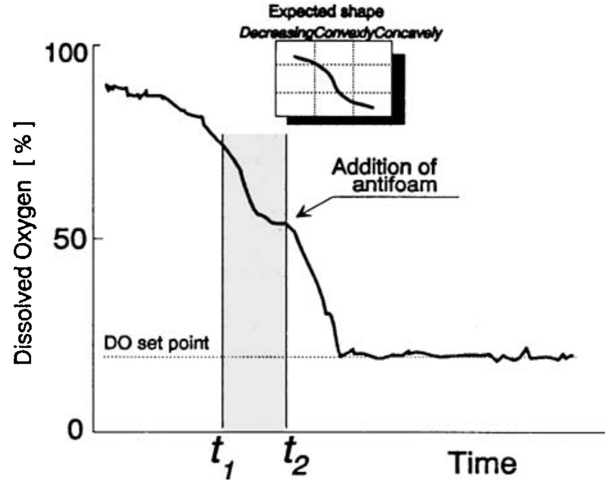

Figure 15. *DO* profile during the initial cultivation stage.
A characteristic *DO* pattern can be used as a reliable symptom of foaming.

However, in some cultivations violation of the threshold value is not unique, but happens repeatedly during earlier process stages. The problem was solved by enhancement of the rule by a qualitative fact:

IF ($R_{a/g}$ > *0.03*) *and* (*DuringTheLast2hr* $R_{a/g}$ *has been IncreasingConcavely*),

THEN (*Report: Intensive acetic acid excretion*) *and* (*Terminate the process*)

If $R_{a/g}$ is increasing, but not concavely, the *dc* of the fact will still be high (see Figure 8), and the rule condition will be eventually satisfied. However, the detailing of the expected pattern adds an important touch of preciseness, which helps distinguish the completely normal cultivation from a slightly deviant one.

The last rule illustrates once again an effective combination between qualitative and quantitative reasoning. As has been reported elsewhere (Grantham and Ungar, 1991; Konstantinov and Yoshida, 1991b), such constructions have high expressive power and can be used to create rules with high efficiency.

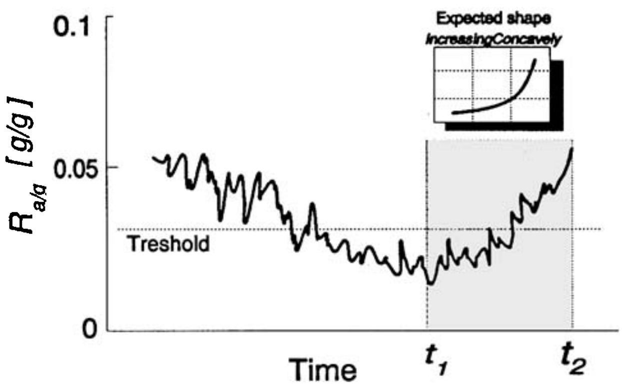

Figure 16. Characteristic time profile of $R_{a/g}$ indicating intensive excretion of acetic acid.

Monitoring of instrumental failures

In some cases it is possible that both the derivative and the value of a particular variable are normal, and no malfunction will be suspected if only these are considered. However, if we look at the time profile as a whole, this may indicate suspicious behavior. For example, a slow, but continuous, buildup of pressure in the bioreactor may be a reliable symptom of a malfunction of the gas outlet system. This was a common problem in our case, caused by gradual clogging of the gas outlet filter. In such a case, the pressure increases with a very small derivative, the value of which is quite possible in a trouble-free system (Figure 17). During the drift, the value of the pressure might also be still in the normal region. Thus, neither the value nor the derivative provides useful information about the progressing phenomenon. What is dangerous here is the long-term, unidirectional, steady drift of the variable. However, this can be seen only from the variable time profile. A rule with very simple logic, which is scanned infrequently, might be helpful in such a case

IF (*DuringTheLast4hr P has been Increasing*),

THEN (*Report: Pressure build up; Outlet gas filter clogging suspected*)

Such logic can be very useful for the early detection of slow phenomena, either of physical or physiological origin (Moore et al., 1990).

Conclusion

Although the concept that when making control decisions, one should look back carefully at the recent process history is intuitively clear and consistent, it has not been exploited efficiently in real-time KB control systems. In this article, we have introduced a method for reasoning about the form of the recent temporal profiles of process variables, which in many cases carry important information about the process state and the underlying process phenomena. This allows the use of temporal shapes together with the current values and the derivatives of the process variables in the construction of rules with greater expressive power. KB shape analysis provides for

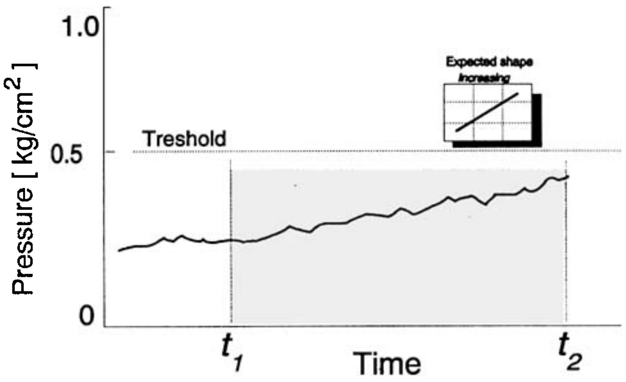

Figure 17. Slow pressure buildup in the bioreactor due to a problem in the gas outlet system can be detected from the long-term, steady drift of the variable.

natural representation of knowledge about the dynamics of complex systems and the capturing of behaviors that otherwise are difficult to detect. Although the discussion has remained within the context of the control of fermentation processes, the proposed methodology can be useful in other highly dynamic environments, such as complex chemical processes. Similarly, KB shape analysis is not limited only to supervisory KB control; it can be incorporated into systems for direct KB control to analyze overshoots, oscillations, stable/unstable behaviors, and so on.

The proposed approach has a qualitative character which to a large extent, corresponds to the natural way of human perception of fermentation processes. Elimination of nonessential quantitative details allows for more abstract and robust knowledge representation, compensating for the weak reproducibility in complex systems. Indeed, we believe that application of qualitative process theory to fermentation processes modeling (development of state-transition qualitative models) and control is very promising.

The performance of the proposed procedure can be enhanced in several ways. Most important are an expansion of the shape library, the addition of other approximating functions, the introduction of shape validation hystereses, fuzzification of the time boundaries, development of adaptive approximation, and improvement of the algorithm for calculation of the *dc* from the qualitative form of the shape. Further research and experimentation in these directions are the only way to transform the discussed method into "tried-and-true" technology.

Literature Cited

Arzen, K. E., "Knowledge-Based Control Systems—Aspects of the Unification of Conventional Control Systems and Knowledge-Based Systems," *Proc. Amer. Control Conf.*, Pittsburgh, 2233 (1989).

Aynsley, M., A. G. Hofland, G. A. Montague, D. Peel, and A. J. Morris, "A Real-Time Knowledge Based System for the Operation and Control of a Fermentation Plant," *Proc. Amer. Control Conf.*, San Diego, 1992 (1990).

Badr, A., Z. Binder, and H. Yassin, "Expert Multimodel Control System," *Proc. IFAC Conf. on Advanced Information Processing in Automatic Control*, Nancy, France, 243 (1989).

Cheung, J. T., and G. Stephanopoulos, "Presentation of Process Trends: I. Formal Representation Framework," *Computers Chem. Eng.*, **14**, 495 (1990a).

Cheung, J. T., and G. Stephanopoulos, "Presentation of Process Trends: II. The Problem of Scale and Qualitative Scaling," *Computers Chem. Eng.*, **14**, 511 (1990b).

Cooney, C. L., G. M. O'Connor, and F. Sanches-Riera, "An Expert System for Intelligent Supervisory Control of Fermentation Process," *Preprints of Int. Biotechnol. Symp.*, Paris, 563 (1988).

Edelmayer, A., J. Bokor, P. Gaspar, and M. Tanyi, "A Knowledge-Based Signal-Symbol Processing Approach for Change Detection," *Preprints IFAC World Cong.*, Tallin, USSR, 7, 168 (1990).

Forbus, K. D., "Qualitative Process Theory," *Artificial Intelligence*, **24**, 85 (1984).

Grantham, S. D., and L. H. Ungar, "Comparative Analysis of Qualitative Models When Model Changes," *AIChE J.*, **37**, 931 (1991).

James, M. L., G. M. Smith, and J. C. Wolford, *Applied Numerical Methods for Digital Computation*, Harper and Row, New York (1977).

Januz, M., and V. Venkatasubramanian, "Automatic Generation of Qualitative Description of Process Trends for Fault Detection and Diagnosis," *Eng. Applic. Artif. Intell.*, **4** (1991).

Konstantinov, K. B., and T. Yoshida, "Physiological State Control of Fermentation Processes," *Biotechnol. Bioeng.*, **33**, 1145 (1989).

Konstantinov, K. B., and T. Yoshida, "An Expert Approach for Control of Fermentation Processes As Variable Structure Plants," *J. Ferment. Bioeng.*, **70**, 48 (1990a).

Konstantinov, K. B., and T. Yoshida, "On-Line Monitoring of Representative Structural Variables in Fed-Batch Cultivation of Recombinant *Escherichia Coli* for Phenylalanine Production," *J. Ferment. Bioeng.*, **70**, 421 (1990b).

Konstantinov, K. B., and T. Yoshida, "Software Architecture of a Knowledge-Based System for Diagnosis and Control of Fermentation Processes," *Preprints IFAC Workshop on Computer Software Structures Integrating AI/ES Systems in Process Control*, Bergen, Norway, 198 (1991a).

Konstantinov, K. B., and T. Yoshida, "A Knowledge-Based Pattern Recognition Approach for Real-Time Diagnosis and Control of Fermentation Processes As Variable Structure Plants," *IEEE Trans. Syst., Man, Cybern.*, **SMC-21**, 908 (1991b).

Konstantinov, K. B., N. Nishio, T. Seki, and T. Yoshida, "Physiologically Motivated Strategies for Control of the Fed-Batch Cultivation of Recombinant *Escherichia Coli* for Phenylalanine Production," *J. Ferment. Bioeng.*, **71**, 350 (1991).

Konstantinov, K. B., and T. Yoshida, "Knowledge-Based Control of Fermentation Processes (Review)," *Biotechnol. Bioeng.*, **39**, 479 (1992).

Konstantinov, K. B., R. Aarts, and T. Yoshida, "Expert Systems in Bioprocesses Control: Requisite Features," *Adv. in Biochem. Eng./Biotechnol.*, **48** (1992).

Konstantinov, K. B., R. Aarts, and T. Yoshida, "On the Selection of Variables Representing the Physiological State of the Cell Cultures," *Proc. IFAC Symp. on Modeling and Control of Biotechnical Processes*, Keystone, CO (1992).

Krijgsman, A. J., R. Jager, H. B. Verbruggen, and P. M. Bruijn, "DICE, a Framework for Intelligent Real-Time Control," *Preprints IFAC Workshop on AI in Real-Time Control*, Sonoma, CA (1991).

Kuipers, B. J. "Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge," *Automatica*, **25**, 571 (1989).

Kuipers, B., "Qualitative Simulation," *Artificial Intelligence*, **29**, 289 (1986).

Laffey, T. J., "The Real-Time Expert," *Byte*, 259 (Jan. 1991).

Moore, R., H. Rosenof, and G. Stanley, "Process Control Using Real-Time Expert System," *Preprints IFAC World Cong.*, Tallin, USSR, 7, 234 (1990).

Odette, L. L., *Intelligent Embedded Systems*, Addison-Wesley, Reading, MA (1991).

Perkins, W. A., and A. Austin, "Adding Temporal Reasoning to Expert-System-Building Environments," *IEEE Expert*, **5**, 23 (1990).

Ramadge, P., and W. Wonham, "Supervisory Control of a Class of Discrete Event Processes," *SIAM J. Control and Optimization*, **25**, 206 (1987).

Rengaswamy, R., and V. Venkatasubramanian, "An Integrated Framework for Process Monitoring, Diagnosis and Control Using Knowledge-Based Systems and Neural Networks," *Proc. IFAC Symp. on On-Line Fault Detection and Supervision in the Chemical Process Industries*, 325 (1992).

Smuts, W. B., and I. M. MacLeod, "Using Discrete AI Techniques for Designing a Real-World Control Supervisor," *Proc. IFAC Workshop on AI in Real-Time Control*, Shenyang, China, 31 (1989).

Verbruggen, H. B., and K. J. Astrom, "Artificial Intelligence and Feedback Control," *Proc. IFAC Workshop on AI in Real-Time Control*, Shenyang, China, 1 (1989).

Manuscript received Dec. 20, 1991, and revision received June 12, 1992.