Fractal Analysis of Time-Series Rule-Based Models and Nonlinear Model Predictive Control

Ching-Yu Peng[†] **and Shi-Shang Jang***

Chemical Engineering Department, National Tsing-Hua University, Hsin-Chu, Taiwan, R.O.C.

Abundant time-series dynamic data can be accumulated from a chemical plant during long-term operations. In our previous work, these plant data were directly implemented for the purpose of model predictive control. However, a large amount of time-series data is required to perform high-quality nonlinear model predictive control. In this work, fractal analysis is performed to reduce the size of a time-series data set for high-quality nonlinear model predictive control. Results in this study indicate that on-line identification of nonlinear models is unnecessary if the disturbances to the process satisfy the fractal-equivalence condition. Simulation examples, including the dual composition control of a high-purity distillation column, demonstrate that the nonlinear model predictive scheme is quite useful for those cases in which the linear model predictive controller has failed.

1. Introduction

Abundant time-series dynamic data can be accumulated during the long-term operation of chemical processes. In our pervious work (Peng and Jang, 1994), those plant data could be easily organized into the following input (**u**) and output (**y**) rule-based model form:

If $\mathbf{y}_k$ is in $\mathbf{Y}_k$, and if $\mathbf{y}_{k-1}$ is in $\mathbf{Y}_{k-1}$, ..., and if $\mathbf{y}_{k-n}$ is in Y_{k-n}, and if u_k is in $\mathbf{U}_k$, ..., and if $\mathbf{u}_{k-m}$ is in $\mathbf{U}_{k-m}$, then $\mathbf{y}_{k+1}$ is in $\mathbf{Y}_{k+1}$. (R1)

where k represents the current state and n and m are known process orders. The above rule set can be implemented as the plant model for nonlinear model predictive control (NMPC). However, the following problems remain unsolved:

(i) If the system is not a single-input-single-output (SISO), and/or if the system orders (n and m) are high, the required rule set that provides high-quality NMPC may be extremely large because it requires a substantially dense rule set.

(ii) On-line identification of low-frequency disturbances is not easily performed. The disturbances must first be measured off-line, and then the required data set becomes larger.

Too large of a data set may be infeasible to save in the computer memory. Further, even in the case of a sufficient amount of computer memory, an on-line search of nonempty membership rules from a large data set may consume too much computer time, thereby making on-line control infeasible. In this work, fractal analysis is applied to solve the above two restrictions.

Model predictive control (MPC) has found successful advanced control scheme applications in the chemical industry since the pioneering works by Culter and Ramaker (1979) and Richalet et al. (1978). In those works and a later substantial review work (Garcia and Morari, 1982), only empirical models such as impulse response models and dynamic matrix models were recommended for use by the model-based controllers. In linear systems, the system dynamics can be well defined by the above empirical models; this is not true for nonlinear systems. In addition, first principle physical models have been most frequently used in recent developments of nonlinear model-based control schemes (e.g., Henson and Seborg, 1991; Jang et al., 1987). However, a physical model is normally quite difficult to implement on-line because (i) a first-principle model is always highly complicated and, consequently, also difficult to implement in real time and (ii) these models also always include many under-determined parameters that should be identified on-line. In this work, the time-series rule-based models (instead of physical models) are implemented as (R1). In a separate work, Lu and Holt (1990) also implemented a look-up table model that is very similar to (R1). However, they failed to mention the rule-based model application in the area of model predictive control. This work provides a detailed description of such an application.

Fractal and chaotic analyses for nonlinear system have become an increasingly active area in chemical engineering. Many researchers have used the chaotic theory to discuss the ergodicity and bifurcation in control loops (e.g., Ydsite and Golden, 1991; Saucier et al., 1987). Some other works applied chaotic theory in process design and simulations (e.g., Fidkowski et al., 1991; Lucia et al., 1990). However, a majority of related works have performed fractal analysis in chemical systems, e.g., a fluidized bed (Fan et al., 1993), and catalysis systems (Giona, 1992; Kinoshita et al., 1992). In the above works, the fractal theory is frequently used to interpret the uncertain behavior of nonlinear systems by choosing appropriate scaling factors. In this work, fractal theory is used to analyze the similarities among the rules in the rule set of (R1). The fractal transformed model of (R1) is also verified to be capable of representing the dynamic behavior of the nonlinear systems in a wider range than the original rule set that is used to perform the fractal analysis.

In this work, fractal analysis of the rule set (R1) is performed to reduce the burden of storage and also search for a large rule set. Furthermore, nonlinear model predictive control using this rule set is shown to be superior to the original approach owing to the fact that the reduced rule set actually represents the process dynamics in a larger range than the original rule set. Also, analysis results indicate that, for many cases, the equivalence condition for different values of disturbances of the original rule-based model is not true. Such a condition can be true in fractal transformed rule

* Author to whom all correspondence should be addressed.

[†] Current address: Energy and Resourse Lab., ITRI, Hsin Chu, Taiwan, R.O.C.

S0888-5885(95)00532-X CCC: $12.00

set including a very important application case: high-purity distillation systems.

2. Theory

A general nonlinear lumped system is considered.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d})$$

$$\mathbf{y} = \mathbf{g}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \qquad (1)$$

where $\mathbf{x} \in R^{l}$ are system states, $\mathbf{u} \in R^{l'}$ are control variables, $\mathbf{y} \in R^{l''}$ are on-line measurements, and $\mathbf{d} \in R^{l'''}$ are disturbances. If the computer control system has executes with a zero-order-hold control element, the following functions ϕ and γ can be obtained from $\mathbf{f}$ and $\mathbf{g}$ in (1) such that:

$$\mathbf{x}_{k+1} = \phi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k)$$

$$\mathbf{y}_k = \gamma(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) \qquad (2)$$

As indicated in our previous work (Peng and Jang, 1994), a function vector $\mathbf{F}$ exists such that:

$$\mathbf{y}_{k+1} = \mathbf{F}(\mathbf{y}_k, \mathbf{y}_{k-1}, ..., \mathbf{y}_{k-n}, \mathbf{u}_k, \mathbf{u}_{k-1}, ..., \mathbf{u}_{k-m}, \mathbf{d}_k) \quad (3)$$

However, the input–output model $\mathbf{F}$ can generally not be easily obtained explicitly, even in the optimum case in which system dynamics $\mathbf{f}$ and $\mathbf{g}$ are completely known. In our previous work, the rule sets in (R1) were used instead of the physical models (2) or (3) to perform model predictive control because the physical models are always quite difficult to obtain. However, as mentioned above, a large rule set may be required for high-quality rule-based control. Furthermore, identifying disturbances requires taking off-line measurements, which may not be available or else would result in a large rule set.

2.1. Nonlinear Model Predictive Control. Given the fact that the known physical state space model (2) or input–output model (3) can be a set of rules in (R1) and a time horizon, the most general model predictive control scheme can be as the following:

$$\min_{\mathbf{u}_k, \mathbf{u}_{k+1}, ..., \mathbf{u}_{k+N_O}} \sum_{i=1}^{N_O} \varphi_i(\hat{\mathbf{y}}_{k+i}, \mathbf{u}_{k+i-1}) \qquad (4)$$

s.t. (2) or (3)

where k represents the present time, φ_i, $I = 1,..., N_O$, is the discretized operating cost as a function of only the input and output variables, $\hat{\mathbf{y}}_{k+i}$ is the model predicted output into the immediate future, and N_O is the interesting time horizon for the immediate future. In this work, we set $\varphi_i = (\hat{\mathbf{y}}_{k+i} - \mathbf{y}_s)^T (\hat{\mathbf{y}}_{k+i} - \mathbf{y}_s)$, where $\mathbf{y}_s$ is the set point. However, in case that some unknown disturbance $\mathbf{d}$ arises in (2) or (3), $\mathbf{d}_k$ must be updated so that the estimated output $\hat{\mathbf{y}}_{k+i}$ can be accurately predicted by the model (2) or (3). To update the model, the following identification problem must frequently be solved on-line:

$$\min_{\mathbf{d}_k} \sum_{i=1}^{N_I} (\hat{\mathbf{y}}_{k-i} - \bar{\mathbf{y}}_{k-i})^2 \qquad (5)$$

s.t. (2) or (3)

Although (5) must be solved on-line, there are some exceptions. As indicated in Economou et al. (1986), if the following superposition condition is satisfied, on-line identification is not necessary, i.e.

$$\mathbf{F}(\mathbf{U}_k + \mathbf{d}_k) = \mathbf{F}_1(\mathbf{u}_k) + \mathbf{F}_2(\mathbf{d}_k) \qquad (6)$$

The optimization problem of on-line control becomes

$$\min_{\mathbf{u}_k, \mathbf{u}_{k+1}, ..., \mathbf{u}_{k+N_O}} \sum_{i=1}^{N_O} (\mathbf{y}_s - \bar{\mathbf{y}}_{k+i} + \hat{\mathbf{y}}_{k+i}) \qquad (7)$$

s.t. (2) or (3)

However, the above superposition condition is generally not true.

2.2. Model Prediction Using the Rule-Based Model. Given a rule set model in (R1), and current state ($\mathbf{y}_k$, $\mathbf{y}_{k-1}$, ..., $\mathbf{y}_{k-n}$,$\mathbf{u}_k$, ..., $\mathbf{u}_{k-m}$), how can the future states $\hat{\mathbf{y}}_{k+1}$, $\hat{\mathbf{y}}_{k+2}$, ... be estimated? In our previous work, fuzzy mathematics were used to find "similar" rules from (R1). The measurements $\mathbf{y}_k$, $\mathbf{y}_{k-1}$, ..., $\mathbf{y}_{k-n}$ are assumed here to be noisy. We assume the fuzzy membership of an element of $\mathbf{y}_k$, $\mathbf{y}_{k-1}$, ..., $\mathbf{y}_{k-n}$ to be in trapezoidal form and the uncertainty of a rule can be expressed by a triangular membership. The fuzzification of a rule for a set of measurements is as follows:

$$\mu_{R_j, \mathbf{y}_k}(\mathbf{Y}_{k+1}) = \min[\mu_{\mathbf{y}_k}(\mathbf{Y}_k), ..., \mu_{\mathbf{y}_{k-n}}(\mathbf{Y}_{k-n}), \mu_{\mathbf{y}_k}(\mathbf{U}_k), ..., \mu_{\mathbf{y}_{k-n}}(\mathbf{U}_{k-m})] \qquad (8)$$

where R_j is the jth rule in the rule set. The defuzzication of a prediction of the future state is as follows:

$$\hat{\mathbf{y}}_{k+1} = \frac{\sum_{j=1}^{M} \mu_{\mathbf{R}_j, \mathbf{y}_{k+1}}(\mathbf{Y}_{k+1}) \cdot \mathbf{Y}_{k+1}}{\sum_{j=1}^{M} \mu_{\mathbf{R}_j, \mathbf{y}_{k+1}}(\mathbf{Y}_{k+1})} \qquad (9)$$

Given (9), problem (4) can be solved. However, as indicated in the previous sections, (a) the rule set (R1) can be too large to store and search and (b) the identification problem (5) may be required to be solved before the predictions of (9) can be made. In the next section, fractal approach is implemented to relieve the burden of a large rule set. In some cases, the NMPC using the fractal-transformed rule-based model of the rules may bypass the on-line identification and go directly to the problem (4).

2.3. Fractal Analysis of Rule-Based Model. The rule-based model set in the form of (R1) is next considered. The rule set is simplified into the following form:

If $\Delta\mathbf{y}_k$ is in $\Delta\mathbf{Y}_k$, and if $\Delta\mathbf{y}_{k-1}$ is in $\Delta\mathbf{Y}_{k-1}$, ..., and if $\Delta\mathbf{y}_{k-n}$ is in $\Delta\mathbf{Y}_{k-n}$, and if $\Delta\mathbf{u}_k$ is in $\Delta\mathbf{U}_k$, ..., and if $\Delta\mathbf{u}_{k-m+1}$ is in $\Delta\mathbf{U}_{k-m+1}$, then $\Delta\mathbf{y}_{k+1}$ is in $\Delta\mathbf{Y}_{k+1}$. (R2)
where $\Delta\mathbf{y}_{k-i} = \mathbf{y}_{k-i} - \mathbf{y}_{k-i-1}$, and $\Delta\mathbf{u}_{k-i} = \mathbf{u}_{k-i} - \mathbf{u}_{k-i-1}$. Given known $\mathbf{y}_{k-n}$ and $\mathbf{u}_{k-m}$, the rule can be easily reconstructed back to (R1) and, hence, $\mathbf{y}_{k+1}$ can be predicted. Figure 1 shows a set of rules of (R2) for the CSTR example (also the example No. 1 in this study) given in our previous work in the above simplified rule-based model (R2). In Figure 1, the magnitudes of $\Delta\mathbf{y}_1$, $\Delta\mathbf{y}_2$, and $\Delta\mathbf{y}_3$ are plotted on the three axes for the same control actions $\Delta\mathbf{u}_1$, $\Delta\mathbf{u}_2$, and $\Delta\mathbf{u}_3$, respectively. This figure indicated that the rules are quite similar to each other except for that the triangle sizes are different.

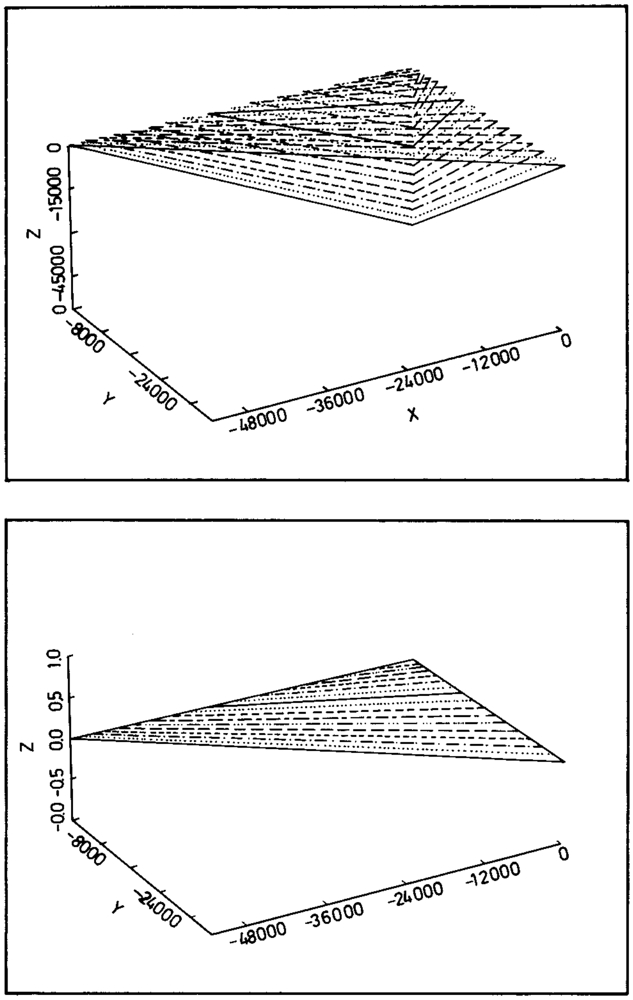

Figure 1. Fractal shapes of a rule set that can be reduced to a single rule. (a) A 3-D plot of a $n = 3$ system. (b) The projection of (a) to a 2-D plane.

Figure 2 plots the other subsets of the rules; however, the rules form two different sets of similar triangles. Figures 1 and 2 lead to the following definition and algorithm:

Definition 1: Reset Dynamics Set of a System. Consider a set containing the following reset dynamics:

$$\mathbf{R} = \{x|x = (\Delta \mathbf{y}_{k+1'}\Delta \mathbf{y}_k, \Delta \mathbf{y}_{k-1}, ..., \Delta \mathbf{y}_{k-n+1}, \Delta \mathbf{u}_k, ..., \Delta \mathbf{u}_{k-m+1})\}$$

We term **R** as the reset dynamics set of the dynamic system.

Algorithm 1: Fractal Analysis. Consider two rules in **R**, that are with same control actions $\Delta \mathbf{u}_k$, $\Delta \mathbf{u}_{k-1}$, ..., $\Delta \mathbf{u}_{k-m+1}$, if there exists a real number **M** such that for any $\Delta \mathbf{y}_{k+1}$, $\Delta \mathbf{y}_k$, $\Delta \mathbf{y}_{k-1}$, ..., $\Delta \mathbf{y}_{k-n+1}$ in rule No. 1 and $\Delta \mathbf{y}'_{k+1}$, $\Delta \mathbf{y}'_k$, $\Delta \mathbf{y}'_{k-1}$, ..., $\Delta \mathbf{y}'_{k-n+1}$ in rule No. 2 the following is true

$$\frac{\Delta \mathbf{y}_{k+1}}{\Delta \mathbf{y}'_{k+1}} = \frac{\Delta \mathbf{y}_k}{\Delta \mathbf{y}'_k} = \frac{\Delta \mathbf{y}_{k-1}}{\Delta \mathbf{y}'_{k-1}} = \cdots = \frac{\Delta \mathbf{y}_{k-n+1}}{\Delta \mathbf{y}'_{k-n+1}} = \mathbf{M} \quad (10)$$

then one of the rules should be eliminated from the rule set.

In real applications, the exact **M** in (10) cannot be found owing to consideration of model uncertainties and measuring noise. The equalities in (10) should be considered with certain range tolerances. Using the above algorithm, all of the rules in Figure 1 will obviously be eliminated except for one rule. Also, all of the rules in Figure 2 will be eliminated except for two rules. Therefore, the following condition is true:

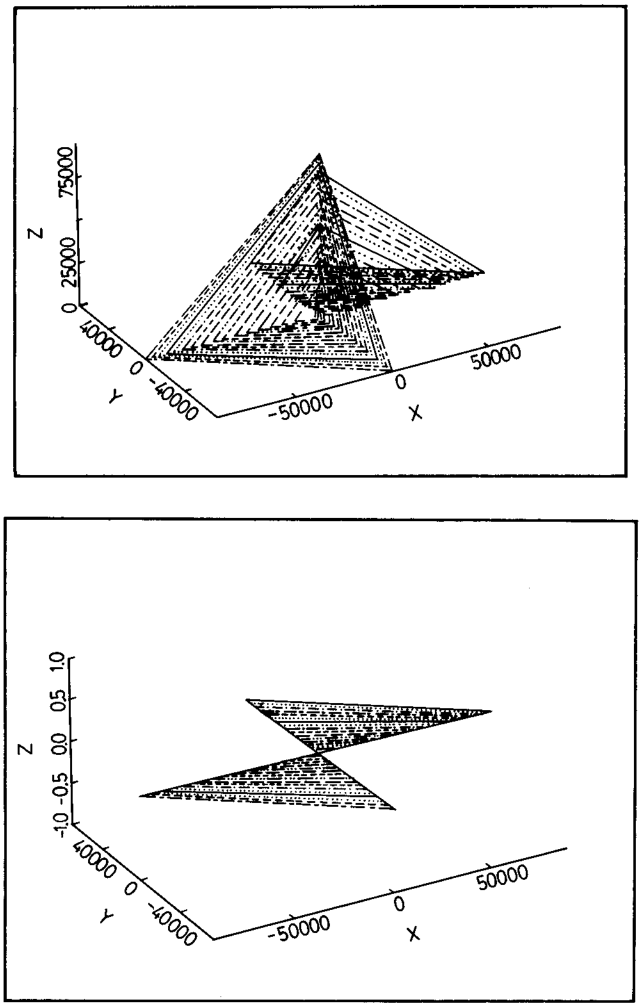

Figure 2. Fractal shape of a rule set that can be reduced to a two rules. (a) A 3-D plot of a $n = 3$ system. (b) The projection of (a) to a 2-D plane.

Property 1. Consider a linear first-order system in the following form:

$$\mathbf{y}_{k+1} = \alpha \mathbf{y}_k \quad (11)$$

and if a rule set exists in the form of (R2) that is obtained from the above linear system, then all of the rules will be eliminated except for only one rule by performing the magnification proposed by Algorithm 1.

Proof: *Omitted.*

However, not all rule sets in any of the systems can be reduced to one rule. The remaining rules in the transformed rule set of **R** may not be so significantly reduced from the original **R**. Hence, the following scaling factor and experimental fractal dimension are defined as:

Definition 2: Scaling Factor and Fractal Dimension. Given a subset **S** (fractal set) of **R** that the rules in the subset are all with the same control actions $\Delta \mathbf{u}_k$, $\Delta \mathbf{u}_{k-1}$, ..., $\Delta \mathbf{u}_{k-m+1}$, then the magnification factor **M** in the above Algorithm 1 is denoted as the scaling factor of the fractal set, and the experimental fractal dimension **D** is denoted by the following:

$$\mathbf{D} = \text{slop}\left(\frac{\ln(\text{no. of rules removed from } \mathbf{S})}{\ln(\mathbf{M})}\right) \quad (12)$$

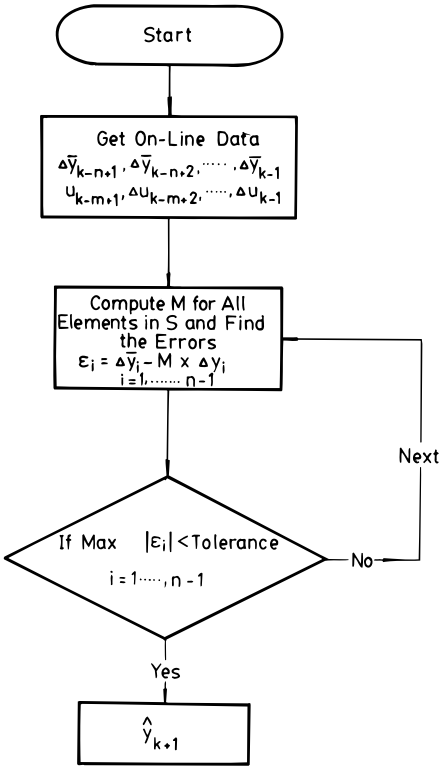
Start
Get On-Line Data
$\Delta\bar{y}_{k-n+1}, \Delta\bar{y}_{k-n+2}, \dots, \Delta\bar{y}_{k-1}$
$u_{k-m+1}, \Delta u_{k-m+2}, \dots, \Delta u_{k-1}$
Compute M for All Elements in S and Find the Errors
$\varepsilon_i = \Delta\bar{y}_i - M \times \Delta y_i$
$i=1, \dots, n-1$
Next
If Max $|\varepsilon_i|$ < Tolerance
$i=1, \dots, n-1$
No
Yes
$\hat{y}_{k+1}$

Figure 3. Flow chart of the reconstruction of a rule by searching for a fractal rule-based model and predictions into the future.

In most systems, systematic approach is unavailable to find the number of rules remaining after the analysis in Algorithm 1 has been performed for **R**. In fractal analysis, the above fractal dimension reflects the "density" of the systems. The following fractal-transformed rule set is defined as follows.

Definition 3: Fractal Rule Set and Prediction Operator. Consider the rule set **R**. If the methodology in Algorithm 1 is performed to the rule set, and the new rule set **S** can be obtained from this operation, then we denote the operator by

$$\mathbf{S} = \tau\{\mathbf{R}\} \tag{13}$$

The future state can be predicted through the reconstructed algorithm in Figure 3, let's also denote the following operator:

$$\hat{\mathbf{y}}_{k+1} = \chi\{\mathbf{S}\} \tag{14}$$

Given a set of data ($\mathbf{y}_k$, $\mathbf{y}_{k-1}$, ..., $\mathbf{y}_{k-n}$, $\mathbf{u}_k$, ..., $\mathbf{u}_{k-m}$), for the current state, Figure 3 illustrates the flowchart that predicts the future states $\hat{\mathbf{y}}_{k+1}$, $\hat{\mathbf{y}}_{k+2}$, ... using the fractal-transformed rule set **S**. The reconstructed algorithm in this figure can also be expanded to a wider situation that requires on-line identification of the generalized NMPC (4). The following can be easily attained by:

Property 2: Fractal Equivalence. Consider the following dynamic system:

$$\mathbf{y}_{k+1} = \mathbf{F}(\cdot, \mathbf{u}_k, \mathbf{d}) \tag{15}$$

and n sets of rules $\mathbf{R}_1(\mathbf{d}_1)$, $\mathbf{R}_2(\mathbf{d}_2)$, ..., $\mathbf{R}_n(\mathbf{d}_n)$, directly obtained from (15). In case that

$$\mathbf{S} = \tau(\mathbf{R}_1) = \tau(\mathbf{R}_2) = \cdots = \tau(\mathbf{R}_n) \tag{16}$$

then

$$\mathbf{S} = \tau(\mathbf{R}_1 \cup \mathbf{R}_2 \cup \cdots \cup \mathbf{R}_n) \tag{17}$$

also

$$\mathbf{y}_{k+1}(\mathbf{d}_i) = \chi(\mathbf{S}) \tag{18}$$

where $i = 1, ..., n$

Proof: *See Appendix 1.*

The validity of the above fractal transform can be summarized in the following: (i) The rule set **S** is usually much smaller than **R**. (ii) The reconstruction of the rule set by Figure 3 using **S** may yield a better estimate of the future states than the original search of **R** because finding the rule to exactly match the current states in **R** is not possible. However, in **S**, current state can more likely be reconstructed exactly if no measuring noise exists by selecting an appropriate magnification factor **M** in (10). This indicates that **S** is actually more "dense" than **R**. (iii) If the condition in Property 2 is true, on-line identification is not required to solve the NMPC problem (4).

3. Numerical Examples

Three realistic examples are simulated in this section to demonstrate the validity of a fractal-transformed rule-based model. The first one is the temperature control of a single-input-single-output (SISO) CSTR control problem, and the second example is the temperature and compostion control of a multi-input-multi-output (MIMO) jacketed CSTR. The above two examples are studied in our previous work (Peng and Jang, 1994). In the SISO CSTR, the output composition of the product ($\mathbf{R}_o$) is manipulated by inlet temperature (T_i) and the inlet composition (A_i) is an unmeasurable disturbance. In the MIMO CSTR example, inlet flow rate (W) and coolant flow rate (W_c) are used to manipulate the reactant composition (C_A) tank temperature (T). Figure 4 shows the schematic plot of a high-purity column that separates methanol from water. The design parameters are displayed in Table 1, while the steady state data are displayed in Table 2. This latter table reveals that the distillation column is actually a high-purity column with top product $x_D = 0.998$ and bottom $x_B = 0.001$. The dynamic model of the column is given in Appendix 2. The simulation is based on a rigorous material and energy balances equation of the tray-by-tray system. A comparison of two conventional controllers were made in Figure 4. The manipulated variables are (i) the reflux flow rate that manipulates the top composition and (ii) the heat duty that manipulates the bottom composition. For simplification, both compositions are assumed to be measured on-line without any time delay. Moreover, the two levels in the sump and the accumulator are assumed to be perfectly controlled. On-line testing indicated that a $n = 3$, $m = 3$ rule-based model is sufficient for NMPC, although the system orders are actually very high. In this case, a comparison is made of the case in which only the top product is controlled (SISO) with that in which both compositions of top and bottom are controlled (MIMO).

3.1. Fractal Analysis of the Rule-Based Model. One thousand rules based on the same control actions $\Delta\mathbf{u}_1$, $\Delta\mathbf{u}_2$, and $\Delta\mathbf{u}_3$, but different initial conditions, are obtained from the SISO CSTR example. Figure 5a shows the experimental fractal dimensions plot for this case. Figure 5b also shows the same case; however, the number of the initial rules are 10 000. These figures reveal that for both cases the fractal dimensions are

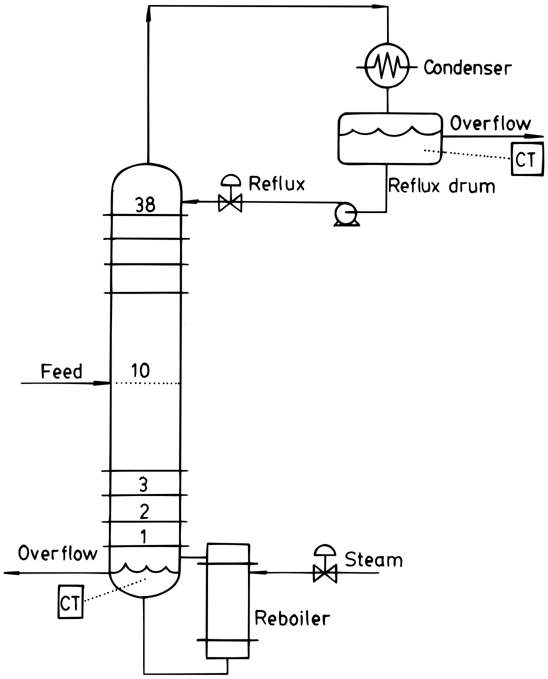

Figure 4. Schematic plot of the high-purity distillation column.

Table 1. Design Parameters of the High-Purity Column

feed composition X_F (mol fraction)	0.3
top product composition X_D (mol fraction)	0.999
bottom product composition X_B (mol fraction)	0.001
feed flow rate (gmol/min)	45 000
feed temperature (K)	330
top product flow rate (gmol/min)	13 495
reflux flow rate (gmol/min)	15 128
bottom product flow rate (gmol/min)	31 505
heat duty (gmol/min)	13 496
pressure (kPa)	101.3
relative volatility	2.45−7.58
tray no.	38
feed position	10
tray efficiency	0.75
tray diameter (m)	3.2
reflux ratio	1.19
reboiler heat load (4.18 × 10^3 kJ/min)	1152
reboiler temperature (K)	372.5
reflux drum temperature (K)	337.7

roughly the same (**D** = 0.0102344 for Figure 5a and **D** = 0.0101234 for Figure 5b). Figure 6 shows a rule reduction example in this SISO CSTR. It gives the rules remained, number of elements in **S**, after fractal reduction with different numbers of initial rule set **R** that is with the same control action. It shows that the number of remaining rules after the fractal reduction is about the same if the numbers of initial rules reaches to 15 000. This reveals the fact that, under this set of control action, this 52 elements in **S** may represent the dynamics of the system much more “dense” than its initial rule set **R**. Furthermore, we found that with a different A_i, the remaining rules after fractal reduction are also the same. According to Property 2, on-line identification is not necessary for a general NMPC. Table 3 shows that the computer memories needed to save the original rule set and the fractal transformed rule set for the above examples. The computer used is an IBM PC-486. It can be found from Table 3 that the memories implemented to store the rule set is drastically reduced by the fractal analysis. It also can be found in Table 3 that the CPU time needed using the fractal condensed rule set is about the same with the CPU time needed using the original rule set.

Table 2. Light Key Component Steady State Composition, Temperature, Liquid Flow Rate, and Vapor Flow Rate Profiles

tray no.	liquid component	vapor component	temp, °C	vapor flow rate	liquid flow rate
1	0.0041	0.0249	99.107	27 495	59 028
2	0.0121	0.0700	98.020	27 439	59 000
3	0.0331	0.1672	95.362	27 408	58 943
4	0.0783	0.3146	90.444	27 577	58 912
5	0.1474	0.4597	84.750	27 967	59 081
6	0.2167	0.5615	80.755	28 345	59 471
7	0.2664	0.6210	78.672	28 594	59 850
8	0.2960	0.6529	77.671	28 734	60 099
9	0.3120	0.6693	77.189	28 807	60 238
10	0.3202	0.6777	76.954	26 886	60 311
11	0.3549	0.6963	76.062	26 982	13 391
12	0.3944	0.7181	75.187	27 094	13 487
13	0.4404	0.7420	74.298	27 217	13 599
14	0.4901	0.7665	73.418	27 344	13 722
15	0.5408	0.7905	72.543	27 407	13 849
16	0.5902	0.8135	71.664	27 591	13 974
17	0.6369	0.8351	70.786	27 705	14 095
18	0.6803	0.8550	69.923	27 813	14 210
19	0.7202	0.8732	69.100	27 912	14 317
20	0.7564	0.8897	68.336	28 004	14 417
21	0.7890	0.9046	67.648	28 088	14 508
22	0.8183	0.9180	67.048	28 163	14 592
23	0.8444	0.9299	66.538	28 231	14 668
24	0.8674	0.9403	66.115	28 291	14 736
25	0.8877	0.9495	65.774	28 344	14 796
26	0.9054	0.9575	65.504	28 391	14 849
27	0.9208	0.9644	65.294	28 431	14 895
28	0.9341	0.9704	65.136	28 465	14 935
29	0.9455	0.9755	65.017	28 495	14 970
30	0.9553	0.9799	64.930	28 520	14 999
31	0.9636	0.9836	64.867	28 541	15 024
32	0.9707	0.9868	64.822	28 559	15 046
33	0.9767	0.9894	64.790	28 574	15 064
34	0.9817	0.9917	64.768	28 587	15 079
35	0.9860	0.9936	64.753	28 598	15 092
36	0.9896	0.9952	64.743	28 607	15 102
37	0.9926	0.9965	64.737	28 614	15 111
38	0.9952	0.9980	64.733	28 623	15 119

3.2. Nonlinear Model Predictive Control Using Fractal-Transformed Rule-Based Model. Figure 7 shows that the fractal-transformed rule-based NMPC performs much better than a well-tuned PID controller for the case of SISO CSTR, if the inlet composition is changed from 1 to 1.05. It should be noted that, in Figure 7, the fractal rule-based model predictive control performs a little better than our original method (Peng and Jang, 1994). This is due to the fact that the fractal rule model can be more “dense” than the original rule set from which the fractal rule model is obtained. Interestingly, the on-line identification for inlet composition is unnecessary as mentioned previously. Figure 8 summarizes the case of regulation control that 8% Gaussian noise is applied to the measurement. Once again, the fractal rule-based model predictive control is superior to PID and our original method. Figure 9a,b also compares the performances of the rule-based NMPC and with a well-tuned PID controller for the case of the MIMO CSTR example.

Figure 10 shows the on-line tracking of the system dynamics for the top product under the case of $x_F = 0.2$ using both the fractal transformed model and the original rule-based model that are obtained from the case of $x_F = 0.3$. It can be observed from Figure 10 that the fractal-transformed rule-based model can track the model well, while the original model does not work. This is due to the fact that Property 2 can be applied to the change of inlet composition changes.

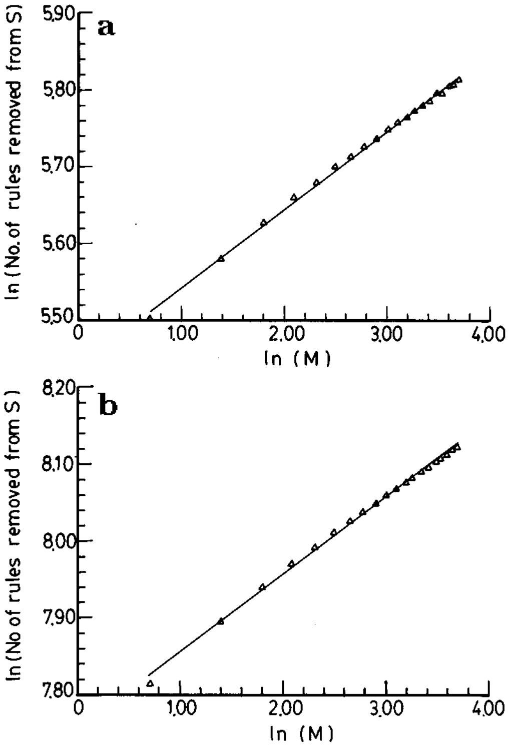

Figure 5. (a) Fractal dimension analysis of the SISO CSTR based on 1000 rules. (b) Fractal dimension analysis of the SISO CSTR example based on 10 000 rules.

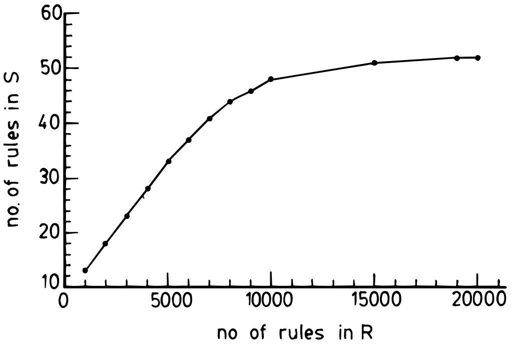

Figure 6. Numbers of rules remained in the fractal-transformed set **S** from the original rule sets **R** with different initial number.

Table 3. Comparison of Computer Memories for Storing the Original Set (R) and Fractal Transform Set (S) for Different Examples

	computer memory for set **R** (MB)	CPU time using set **R** (s)	computer memory for set **S** (MB)	CPU time using set **S** (s)
SISO CSTR	88	1.1	1.3	0.21
MIMO CSTR	465	0.65	0.7	0.01
high-purity column SISO case	21	0.08	3	0.1
high-purity column MIMO case	80	0.2	3.1	0.08

Figure 11 compares the regulation behavior using a well-tuned PI controller and NMPC for the SISO case of a high-purity column. The fractal-transformed rule-based control is markedly superior to PI controller in regulation control. In other simulation studies, the rule-based model is also markedly superior to PI controllers in servo control. Figure 12 also shows that the fractal-transformed model is superior to a nontransformed case for the case that the inlet composition is changed (x_F suddenly changed to 0.2 from 0.3). Since the prediction of the rule-based model is not correct as shown in Figure 10 and an identification phase is not devised, our original approach failed. However, the fractal-transformed model works fine as shown in Figure 12. Figure 13 compares the performances of the fractal-transformed rule-based control and the NMPC proposed by Economou et al. (1986). We assume that both controllers are all designed based on the wrong case (x_F = 0.3). Since the fractal-transformed rule-based model can track the column dynamics well for x_F = 0.2 the rule-based NMPC works fine. The NMPC using a physical model with the wrong parameter (x_F = 0.3) may work for regulation case because the system does not deviate from its steady state too much during the

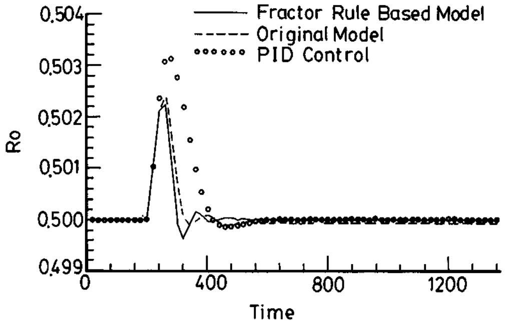

Figure 7. Comparisons of regulation (A_i = 1.05) behaviors of the SISO CSTR example among (1) the fractal rule-based model control, (2) original rule-based model control, and (3) PID control.

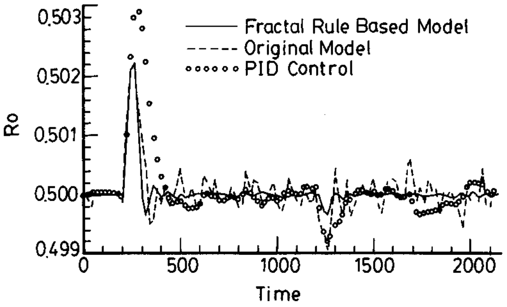

Figure 8. Comparisons of regulation behaviors of the SISO CSTR example with 8% noise among the (1) fractal rule-based model control, (2) original rule-based model control, and (3) PID control.

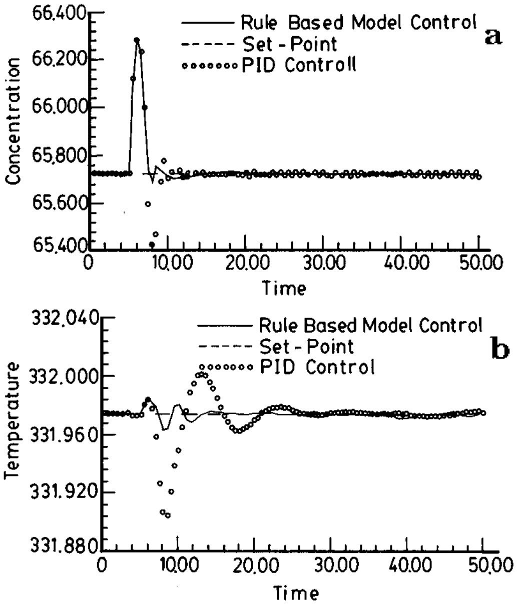

Figure 9. Comparisons of regulation behaviors with 8% noise between a fractal-transformed rule-based NMPC and well-tune PID controllers. (a) CA vs time; (b) temperature vs time.

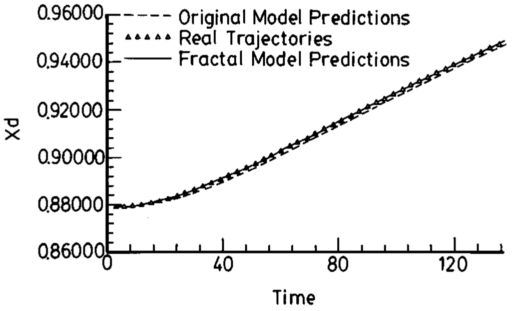

Figure 10. On-line tracking of the top product dynamics if the feed composition is $x_f = 0.2$ based on (i) a rule set obtained from $x_f = 0.3$ and (ii) a fractal-transformed rule set obtained for (i).

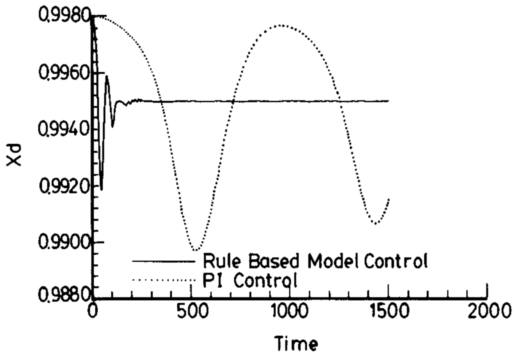

Figure 11. Regularity behavior (x_F change from 0.3 to 0.2) of fractal-transformed NMPC and PI controller for SISO high-purity column.

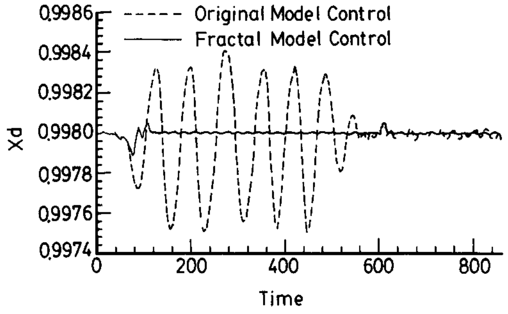

Figure 12. Comparison of regular behavior (x_F change from 0.3 to 0.25) between (1) fractal-transformed NMPC and (2) original rule-based NMPC (Peng and Jang, 1994) but without an identification phase.

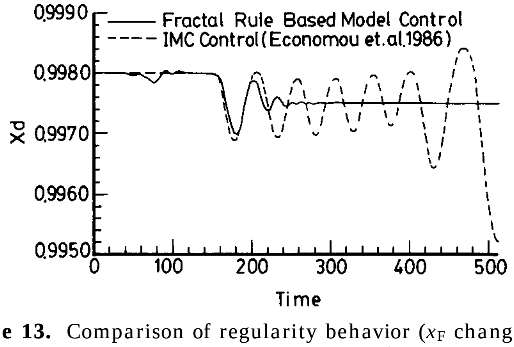

Figure 13. Comparison of regularity behavior (x_F change from 0.3 to 0.2) and servo behavior (setpoint change from 0.998 to 0.9975 at $t = 150$ min) between (1) fractal-transformed NMPC and (2) physical-model-based NMPC (Economou et al., 1986).

transient. However, if a set point change is implemented, the system becomes unstable as shown in Figure 13.

Figure 14a,b compares the MIMO distillation control using the NMPC with that using the PI controller with 2% measuring noise. Once again, the NMPC performs much better than PI controllers.

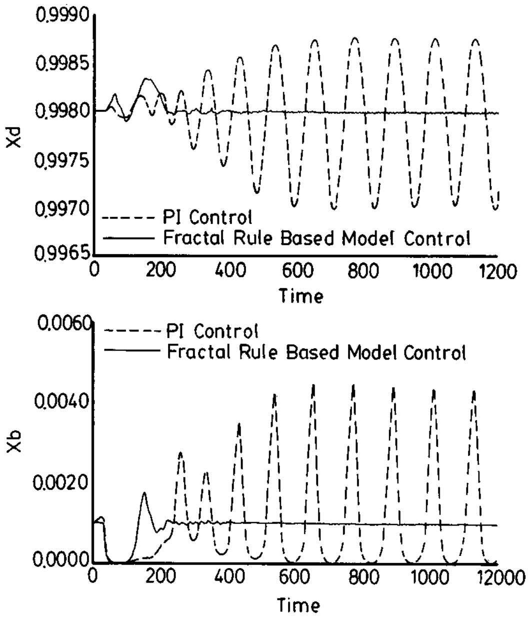

Figure 14. Comparisons of regulation behavior (x_F change from 0.3 to 0.2) between the fractal transformed NMPC and PI controllers for MIMO high-purity column.

4. Conclusion

A fractal-transformed rule-based model was proposed to perform NMPC in this work. The transformed model could actually represent the system dynamics in a wider range than the original rule-based model in the sense that a different number of the original rules could be transformed into the same rule set. Fractal analysis results indicated that if the rule set with different values of disturbances can be transformed to the same rule set, then on-line identification would be unnecessary.

Furthermore, realistic examples including high-purity distillation column were also simulated. The simulation examples demonstrated that the fractal-transformed rule-based model is markedly superior to the conventional controllers.

Acknowledgment

The authors thank the National Science Council, Republic of China, through Grant NSC 81-0402-E007-16, for financial support of this work.

Nomenclature

D = fractal dimension
d = disturbances
m = input order
n = output order
N_I = identification horizon
N_O = optimization horizon
R = rule set
S = reconstructed rule set
t = time
U = discretized manipulated variable in the rule set
u = manipulated input
x = vector of state variables
Y = discretized output in the rule set
y = output variable

Greek Letters

μ = membership function

ϵ = tolerance
τ = fractal analysis operator
χ = prediction operator using a fractal transformed set

Superscripts

$\hat{}$ = predicted value
$-$ = measured value

Subscripts

k = current time

Appendix 1. Proof of Property 2

Consider $\mathbf{S} = \{\mathbf{R}_i\}$, then for any $\mathbf{y}\mathbf{R}_i$, there exists a $x\mathbf{S}$ and scaling factor $\mathbf{M}$ such that (10) holds, where i = 1, ..., n. Since $\mathbf{S} = \{\mathbf{R}_i\}$ for all i, so for all

$$\mathbf{y} \in \mathbf{R}_1 \cup \mathbf{R}_2 \cup ... \cup \mathbf{R}_n$$

there exists a $x \in \mathbf{S}$ and scaling factor $\mathbf{M}$ such that (10) holds. This concludes the proof.

Appendix 2. Physical Model of the High-Purity Distillation Column

Tray Model

Material Balance

$$\frac{\mathrm{d}M_n}{\mathrm{d}t} = F + L_{n+1} + V_{n-1} - L_n - V_n$$

Component Balance

$$\frac{\mathrm{d}(x_i M_n)}{\mathrm{d}t} = Z_{i,f}F + x_{i,n+1} + y_{i,n-1}V_{n-1} - x_{i,n}L_n - y_{i,n}V_n$$

Energy Balance

$$\frac{\mathrm{d}(h_n^L M_n)}{\mathrm{d}t} = h_F F + h_{n+1}^L L_{n+1} + h_{n-1}^V V_{n-1} - h_n^L L_n - h_n^V V_n$$

$$\frac{\mathrm{d}h_n^L}{\mathrm{d}t} = 0$$

Liquid Hydraulics

$$L_n = C\rho_n^L w_{len} H_{ow}^{1.5}$$

(C: unit conversion constant)

(w_{len}: Francis weir)

Vapor–Liquid Equilibrium

$$y_{i,n} = E_n K_{i,n} x_{i,n} \Leftrightarrow K_{i,n} = \frac{y_{i,n} p_{i,n}^s}{P_n}$$

Reboiler Model

Material Balance

$$\frac{\mathrm{d}M}{\mathrm{d}t} = L_1 - V_0 - B$$

Component Balance

$$\frac{\mathrm{d}(x_{b,i} M)}{\mathrm{d}t} = x_{1,i}L_1 - y_{0,i}V_0 - x_{b,i}B$$

Energy Balance

$$\frac{\mathrm{d}(h_b^L M)}{\mathrm{d}t} = h_1^L L_1 - h_0^V V_0 - h_b^L B + Q_{\mathrm{R}}$$

$$V_0 = \left(h_b^L B + h_b^L \frac{\mathrm{d}M}{\mathrm{d}t} + M\frac{\mathrm{d}h_b^L}{\mathrm{d}t} - h_1^L L_1 + Q_R\right)/h_0^V$$

Vapor–Liquid Equilibrium

$$y_{0,i} = \frac{\gamma_i P_i^{\mathrm{s}}}{P_0} x_{b,i}$$

Condenser Model

Material Balance

$$\frac{\mathrm{d}M_n}{\mathrm{d}t} = V_{n-1} + F_n - L_n$$

Energy Balance

$$\frac{\mathrm{d}(M_n h_n)}{\mathrm{d}t} = F_n h_{f,n} + V_{n-1}H_{n-1} - L_n h_n + Q_n$$

$$Q_n - (h_n - H_{n-1})V_{n-1} = F_n(h_n - h_{f,n})$$

Literature Cited

Culter, C. R.; Ramaker, B. L. Dynamic Matrix Control: A Computer Control Algorithm, 86th National Meeting AIChE, Houston, TX, April, 1979.

Economou, G. E.; Morari, M.; Palsson, B. O. Internal Model Control. 5. Extension to Nonlinear Systems. *Ind. Eng. Chem. Process Des. Dev.* **1986**, *25*, 403–411.

Fan, L. T.; Kang, Y.; Neogi, D.; Yadhima, M. Fractal Analysis of Fluidized Particle Behavior in Liquid-solid Fluidized Beds. *AIChE J.* **1993**, *39*, 513–517.

Fidkowski, Z. T.; Malone, M. F.; Doherty, M. F. Nonideal Multicomponent Distillation: Use of Bifurcation Theory for Design. *AIChE J.* **1992**, *37*, 1761–1779.

Garcia, C. E.; Morari, M. Internal Model Control 1. A Unifying Review and Some New Result. *Ind. Eng. Chem., Process Des. Dev.* **1982**, *21* (2), 308–323.

Henson, M. A.; Seborg, D. E. An Internal Model Model Control Strategy for Nonlinear Systems. *AIChE J.* **1991**, *37*, 1065–1081.

Jang, S. S.; Joseph, B.; Mukai, H. On-Line Optimization of Constrained Multivariable Chemical Processes. *AIChE J.* **1987**, *33*, 26.

Lu, Z. H.; Holt, B. R. Nonlinear Robust Control: Table Look-Up Controller Design. *Proc. Am. Control Conf.* **1990**, 2758–2763.

Lucia, A.; Guo, X.; Richey, P. J.; Derebail, R. Simple Process Equations, Fixed-Point Methods, and Chaos. *AIChE J.* **1990**, *36*, 641–654.

Peng, C. Y.; Jang, S. S. Nonlinear Rule Based Model Predictive Control of Chemical Processes. *Ind. Eng. Chem. Res.* **1994**, *33*, 2140.

Richalet, J.; Rault, A.; Testud, L.; Papon, J. Model predictive Heuristic Control: Application to Industrial Processes. *Automatika* **1978**, *14*, 413.

Saucier, M. F.; Chang, H. C.; Seborg, D. E. Bifurcation-Analysis of Multivariable Feedback-Control Systems. *Chem. Eng. Commun.* **1987**, *57*, 215–232.

Ydsite, B. E.; Golden, M. P. Small Amplitude Chaos and Ergodicity in Adaptive Control. *Automatika* **1992**, *28*, 11–25.

Received for review August 25, 1995
Revised manuscript received April 24, 1996
Accepted April 25, 1996[⊗]

IE9505322

[⊗] Abstract published in *Advance ACS Abstracts*, June 15, 1996.