ELSEVIER

Fuzzy Sets and Systems 78 (1996) 107–119

FUZZY sets and systems

The integration and application of fuzzy and grey modeling methods

Yo-Ping Huang*, Chi-Chang Huang

Department of Computer Science and Engineering, Tatung Institute of Technology, 40 Chungshan N Rd. 3rd Sec., Taipei, Taiwan 104, ROC

Received May 1994; revised November 1994

Abstract

An integrated fuzzy and grey model and its applications to the prediction control problems are presented. The basic grey model GM(1, 1) is accompanied with the adaptive fuzzy method to improve its prediction capability. The gradient descent scheme is applied to the fuzzy rules to determine whether the predicted results from the grey model should be adjusted. The quantity of adjustment is judged from the degrees of the correlation between the past data and the current input. We select a few most correlated patterns to decide the direction of adjustment. Due to the simplicity of the structure and its fast learning characteristics, this model is good as a real-time controller. Under the proposed methodology, the simulation results are shown to be superior to those systems which exploit complicated control variables and rules. A well-known difference equation and the weather forecast prediction problems are depicted to verify the superiority of the proposed method.

Keywords: Fuzzy modeling; Grey system; Grey relational space; Gradient descent method; Prediction control

1. Introduction

In the past decade, the integrated neural network and fuzzy logic technique has been successfully applied to solving many complicated control problems such as the truck backer-upper and lock avoidance systems [6–8, 11]. The advantage of the neuro-fuzzy approach is that we can design a simple model to replace the conventional mathematically based methodology. The performance of the constructed model, however, depends heavily on the acquired knowledge base. For example, how to define the fuzzy rules and how to tune the corresponding membership functions are two requisites. A well-defined expert knowledge or the sample data would have a positive contribution to the system performance. In the situation that the model lacks the assistance of an expert or the collected data are not representative, this ill-defined knowledge base cannot be expected to produce a fascinating outcome.

* Corresponding author.

SSDI 0165-0114(95)00083-6

A systematic approach is therefore necessary for us to design a good fuzzy controller [4, 9]. The steepest descent method is normally exploited to tune the membership functions. The tuning process is gone through the hope that the performance can be improved. For a large inference rule-based system, the process is time-consuming because of its repeated tuning nature. Therefore, in our proposed method we use the a priori linguistic variables which are chosen to be necessary for forming the fuzzy rules. As a result, the converging speed is further improved with the system performance maintained.

The grey model was first proposed by Deng in 1982 [1–2]. A grey system is a partially known and partially unknown system. This system provides a method which uses only a few data just used to predict the next output from the plant. Without resorting to forming a knowledge base, the grey modeling scheme constructs a differential equation to characterize the controlled system. Therefore, the next output from the system can be obtained by solving the differential equation. Since only a few data are adopted in forming the GM(1, 1) model, there is a chance that the constructed differential equation cannot completely represent the system dynamics. To better improve the prediction capability, the fuzzy method is incorporated into the grey controller.

To supplement the grey prediction model, we use the grey correlation model to select some past patterns which possess similar characteristics with the current one to modify the outcome from the original grey model. The quantity of modification is determined by the fuzzy model. In short, the proposed model incorporates the fuzzy model and grey correlation into the grey model to improve the grey prediction accuracy. In Section 2, an on-line GM(1, 1) prediction model is presented. The grey correlation scheme is stated in Section 3. After learning how to capture the most correlated patterns, a hybrid gradient descent learning algorithm is discussed in Section 4. Section 5 illustrates how to modify the GM(1, 1) prediction model. Examples and their simulation results are given in Section 6. Conclusions are made in the final section.

2. The on-line GM(1, 1) prediction model

Traditionally, to mathematically model an unknown system was a cumbersome work. Instead of analyzing the characteristics of the unknown system mathematically, the grey system exploits the accumulated generating technique to approach the plant's behavior. The raw data output from the plant may not possess any regularity. However, the original data may become more regular after a repeatedly accumulated generating operation [1]. Therefore, we can use the differential equation to approximate such a regularity and hopefully to predict the next output from the plant.

A grey system is a system with insufficient information contained. The advantages of the grey model are: (1) it can use only a few data to estimate an unknown system, and (2) it can use a first order differential equation to characterize the unknown system behavior [5]. That is, we can use a few discrete data to form a first order differential equation to characterize such an unknown system. This is why it is applicable to the time-varying nonlinear system prediction problem.

The GM(1, 1) grey model, i.e., a single variable first order grey model, is summarized as follows [1]:

Step 1: Given the initial data $X(0) = [x(1), x(2), \ldots, x(n-1), x(n)]$, where $x(i)$ corresponds to the system output at time i. We try to predict the next $x(n+k)$, $k \geqslant 1$.

Step 2: From the initial $X(0)$ a new sequence $X(1)$ is generated by the accumulated generating operation (AGO), where $X(1) = [x^1(1), x^1(2), \ldots, x^1(n)]$ and is derived as follows:

$$x^1(k) = \sum_{m=1}^{k} x(m). \tag{1}$$

Step 3: From $X(1)$ we can form the following first order differential equation:

$$\frac{\mathrm{d}x^1}{\mathrm{d}t} + ax^1 = u. \tag{2}$$

Step 4: From Step 3 we have

$$\hat{x}^1(k+1) = \left(x(1) - \frac{u}{a}\right)\mathrm{e}^{-ak} + \frac{u}{a}, \tag{3}$$

$$\hat{x}(k+1) = \hat{x}^1(k+1) - \hat{x}^1(k), \tag{4}$$

where

$$\hat{a} = \begin{bmatrix} a \\ u \end{bmatrix} = (B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}y_N, \tag{5}$$

$$B = \begin{bmatrix} -0.5(x^1(1) + x^1(2)) & 1 \\ -0.5(x^1(2) + x^1(3)) & 1 \\ \cdots & \cdots \\ -0.5(x^1(n-1) + x^1(n)) & 1 \end{bmatrix},$$

$y_N = [x(2), x(3), \ldots, x(n)]^{\mathrm{T}}$, and $\hat{x}(k+1)$ is the predicted value of $x(k+1)$ at time $k+1$.

Fig. 1 depicts the structure of the GM(1, 1) prediction model. The GM(1, 1) block is dedicated to predict the system next output based on the recently past outputs. The control strategy in Fig. 1 is adopted to modify the predicted result or used to adjust the system input based on the predicted outcome such that the system can reach the desired outcome. Thus, the performance of the system heavily depends on the GM(1, 1) prediction accuracy. Empirically, if the ratio between two consecutive data is close to one, then the grey model shows a satisfactory result. Consider the following function [12]:

$$y(k) = 0.6\sin(\pi u(k)) + 0.3\sin(3\pi u(k)) + 0.1\sin(5\pi u(k)) + 0.8, \tag{6}$$

where $u(k) = \sin(2\pi k/250)$, k is an integer. We use only four past data to construct the GM(1, 1) model. The actual and predicted results are plotted in Fig. 2 for comparison. As can be seen, it is hard to distinguish between these two curves. Actually, the predicted error from the grey model is only 2.22%. This result demonstrates that the grey modeling technique is simple but powerful in the prediction control.

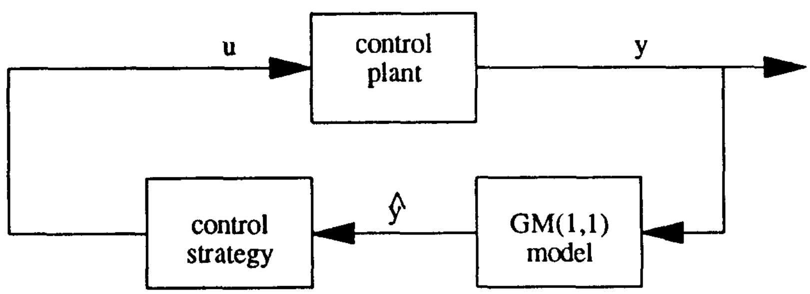

Fig. 1. The grey prediction control system. Note that y and $\hat{y}$ are the actual output from the plant and the predicted outcome from the grey model, respectively.

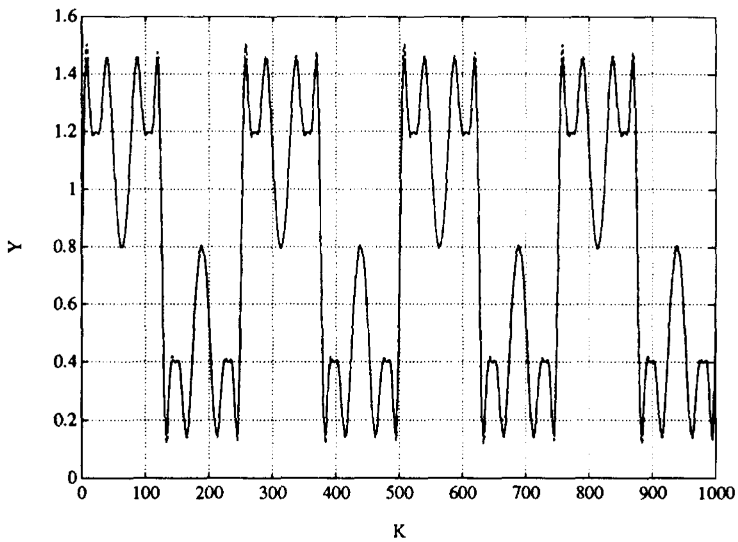

Fig. 2. The actual and predicted results from the function formulated in Eq. (6).

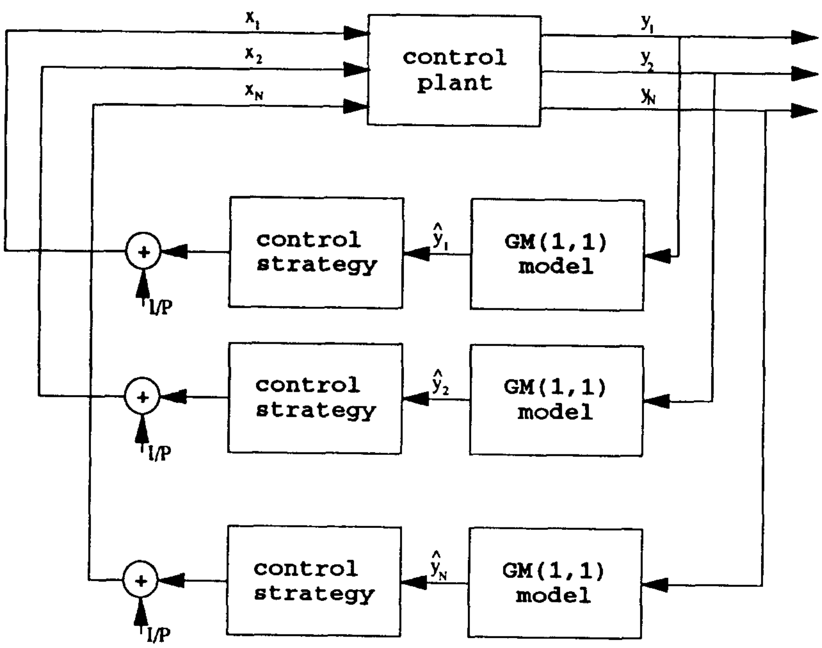

Fig. 3. The multiple inputs multiple outputs grey prediction control system.

The GM(1, 1) model can be easily extended to become a GM(1, N) model [2]. Note that the second index in the GM(1, N) stands for N variables considered in the first order differential equation. Fig. 3 shows such a multiple input multiple output system.

3. The grey relational model

The grey relational space (GRS) is originally believed to have captured the relationship between the main factor and the other reference factors in a given system [1]. We applied this technique here to relate the current modeling pattern to the previous patterns which show some degrees of similarity to the current one such that the quantity of adjustment to the predicted outcome can be determined. The grey relational model can be summarized as follows:

Step 1: Let the reference sequence be $x_0 = (x_0(1), x_0(2), \ldots, x_0(n))$.

Step 2: Denote the m sequences to be compared by $x_i = (x_i(1), x_i(2), \ldots, x_i(n))$, $i = 1, 2, \ldots, m$.

Step 3:

$$\gamma(x_0(k), x_i(k)) = \frac{\min_j \min_k |x_0(k) - x_j(k)| + \zeta \max_j \max_k |x_0(k) - x_j(k)|}{|x_0(k) - x_i(k)| + \zeta \max_j \max_k |x_0(k) - x_j(k)|},$$

where $\zeta \in (0, 1]$ is the distinguishing coefficient. $j = 1, 2, \ldots, m$. $k = 1, 2, \ldots, n$. $\gamma(x_0(k), x_i(k))$ is called the grey relational coefficient at point k. From this grey relational coefficient we can obtain the grey relational grade.

Step 4: The grey relational grade is derived as follows:

$$\gamma(x_0, x_i) = \frac{1}{n} \sum_{k=1}^{n} \gamma(x_0(k), x_i(k)).$$

Here, $\gamma(x_0, x_i)$ represents to what degree of influence the sequence x_i can exert on the reference sequence x_0. In other words, the reference sequence can grasp some useful information about the variation of data points from other similar sequences. From the analysis of the grey relational grade we can understand which factors will crucially affect the reference factor. This is especially important in the decision making process. Here, we use the relation to decide the quantity of adjustment.

4. The hybrid gradient descent learning algorithm

After selecting the related patterns, we have to calculate the degree of compensation for the predicted output. The question is how much we should compensate for the predicted one. In this section, we discuss how to exploit the fuzzy reasoning and the gradient descent method to facilitate the learning process.

4.1. The symmetric and asymmetric adjusting methods

When a set of input–output data pairs is given, the inference rules can be easily generated as follows [10]:

Rule i: IF x_1 is A_{i1} and x_2 is A_{i2} and … and x_m is A_{im}, THEN y is w_i.

Where x_j's are the system input data, y is the output, A_{ij}'s are the membership functions in the antecedent, and w_i is a real number in the consequent part. For convenience, the triangular membership functions are assumed. The parameters associated with the membership functions A_{ij} are the center a_{ij} and the spreads b_{ij} and c_{ij} as shown in Fig. 4. If the left spread b_{ij} is equal to the right spread c_{ij}, then this membership function is called symmetric; otherwise, it is asymmetric. To properly compensate for the predicted outcome, the selected grey related patterns must be trained first. It means we have to tune the roughly determined membership functions to satisfy those selected patterns. This invokes the employment of the gradient descent method [3, 9]. The tuning process is stated as follows:

Step 1: Calculate the firing grades for each input from Fig. 4.

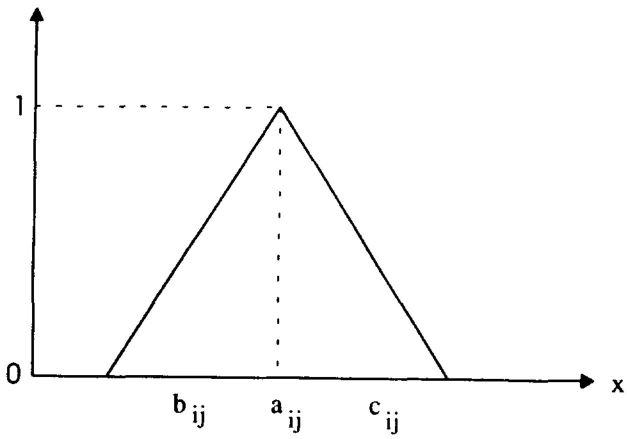

Fig. 4. The triangular membership function and the related parameters. a_{ij}, b_{ij}, and c_{ij} stand for the center, the left spread, and right spread of the triangular membership function, respectively.

Step 2: Take the AND (product) operation in the antecedent, i.e.,

$$\mu_i = \prod_{j=1}^{m} A_{ij}(x_j),$$

where m represents the input dimensions.

Step 3: Calculate the output y_c by the center of gravity method, i.e.,

$$y_c = \left(\sum_{i=1}^{n} \mu_i w_i \right) \Big/ \sum_{i=1}^{n} \mu_i,$$

where n is the number of rules considered.

Step 4: Calculate the mean square error E, i.e., $E = \frac{1}{2}(y - y_c)^2$, where y is the desired output.

Step 5: Update the parameters by the gradient method, $z_i(t+1) = z_i(t) - k \cdot \partial E/\partial z_i$, where z_i represents either a_{ij}, or b_{ij}, or c_{ij}, or w_i, and k stands for k_w, or k_a, or k_b, or k_c, and is called the learning rate. Note that the subscript of k denotes the parameter under consideration. The choice of the learning rates, however, is subjectively made or problem depended.

The purpose in employing the gradient descent method is to automatically adjust the roughly determined membership functions such that the behavior of those selected patterns can be grasped. Once the rule base has been established, the compensation for the current input pattern can be easily done.

4.2. *The hybrid adjusting algorithm*

As we mentioned in Section 4.1 that the selection of the learning rates is very important in the gradient descent method. Without resorting to a good selection of initial parameters, the underlying techniques are proposed to adjust the parameters. The proposed hybrid adjusting algorithm is given as follows:

Step 1: Roughly determine the required number of labels in each variable. Uniformly distribute the membership functions in the universe of discourse. In our simulation, we use six labels for each variable.

Step 2: Fix the antecedent membership functions. Adjust the consequence by the gradient descent method. Note that the consequent parts in the rule base are assumed to be crisp numbers.

Step 3: Adjust the antecedent membership functions by the symmetric method if no local minimum occurs. Note that local minimum is detected if no apparent change of error (a preset value) is observed during two consecutive runs. Switch to the asymmetric method when a local minimum appears. After skipping out of the local minimum, the system returns to the symmetric adjusting process.

Step 4: Stop the pruning when the total error does not cross the tolerance threshold; otherwise, go to Step 3 and continue the adjusting process.

In order to facilitate the implementation of this approach by the neural network and to alleviate the dependence on the expert knowledge in constructing a good rule base, only crisp numbers are adopted in the consequent parts of the rule base here. Besides, in this adjusting algorithm, the symmetric approach is performed first. This is done in the hope that we can locate the domain of each label quickly. However, if a local minimum occurs, we have to resort to the asymmetric technique. Unlike the symmetric adjustment, the asymmetric method tunes only one side of the related membership function. This asymmetric approach, therefore, can prevent the newly adjusted membership functions from affecting most of the fuzzy rules.

When the system returns from the asymmetric to the symmetric procedure, the symmetric property of the membership functions has been altered. Therefore, if one side of the membership function is adjusted, the other side should also be tuned to maintain the symmetric property. The degree of adjustment of the unfired part depends on its own spread and the grade taken from the fired side. As a result, the functions remain asymmetric after the adjustment.

5. The modified GM(1, 1) prediction model

After figuring out how to find the grey relational grades and how to use the hybrid learning algorithm, we integrate these techniques into a new prediction model as depicted in Fig. 5. In this model, we use the grey relational grades to find the plant's past output patterns with characteristics similar to the current pattern. In selecting the related patterns, we can also place the priority of the past data patterns. For example, in weather forecast we can settle on the pattern derived from last year, considering it to have priority over those from a decade ago. Of course, this priority concept is case-dependent. For a fair study, we can eliminate it.

Suppose the to-be-trained patterns have been picked according to the grey relational grades discussed in Section 3, we then resort to the algorithm discussed in Section 4 to adjust the membership functions. The

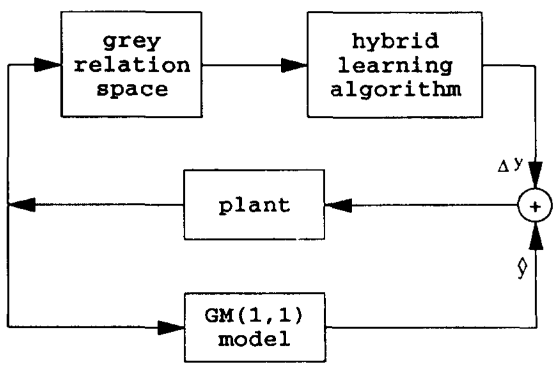

Fig. 5. The modified GM(1, 1) prediction model. The block of grey relation space is used to select some of the correlated patterns from the previous data. The hybrid learning algorithm is adopted to automatically adjust the rule base.

number of variables considered in the antecedent part of the fuzzy rule is decided by the number of data points used in constructing the grey model. The consequent part in each fuzzy rule is the difference between the actual output and the predicted one. It means that a simple GM(1, 1) model is taken for each to-be-trained pattern, a sequence of data points, to predict the next output. Since the pattern is picked from the past data sequences, the actual value next to the picked pattern is also known to us. There is no doubt that the difference between the actual and predicted values can be easily obtained.

After adjusting the membership functions to fit the selected patterns, the current input pattern is then fed to the well-tuned system to determine the quantity of compensation. A positive compensation value means an underpredicted situation occurring in the basic grey model; therefore, the predicted output from the grey model should be promoted. On the contrary, a negative value suggests that we lower the predicted output. The compensation is later added to the GM(1, 1) predicted output such that the system can have a modified output. In case a wrong direction is compensated, we can conclude that the reference patterns failed to provide positive information to the current pattern. Thus, a threshold must be set in advance to pick the number of reference patterns in Section 3. A lower threshold implies that more patterns can be selected. This may in turn complicate the tuning process. A higher threshold means that fewer patterns can be involved. This may result in a lack of useful information for the current pattern. Normally, the number of reference patterns is subjectively selected.

6. Simulation results

To verify the effectiveness of the proposed model, two different examples are given here. The first one is to predict the output of a difference equation when different initial data are known. The other example relates to the monthly average weather forecast for the city of Taipei.

Example 1. The difference equation is given as follows:

$$y(k+1) = 0.3y(k) + 0.6y(k-1) + g(u(k)) + 5.5, \tag{7}$$

where

$$g(u(k)) = 0.6\sin(\pi u(k)) + 0.3\sin(3\pi u(k)) + 0.1\sin 5\pi u(k)), \tag{8}$$

where

$$u(k) = \sin(2\pi k/250), \quad k = \text{integer}. \tag{9}$$

Suppose a number of data points have been known and we try to predict the next 250 points. In performing the simulations, we picked five most correlated patterns to learn the direction of compensation. Since we used four data points to construct the grey model, to simplify the learning, we took the difference between two consecutive data as a variable in the fuzzy rule base. Thus, only three variables, each with six labels uniformly distributed in the universe of discourse initially in this case, are needed in the antecedent part. The simulation results are shown in Figs. 6–11 for the known data equal to 150, 200, and 250. Note that in Figs. 6, 8 and 10 each comprises three curves, i.e., from the actual, the GM(1, 1), and our models, respectively. Due to the scale effect, it is hard to distinguish between them. Table 1 compares the differences betweeh our's and the original GM(1, 1) models. Note that the error percentage is defined as follows:

$$\text{error\%} = \frac{|\,\text{actual} - \text{predicted}\,|}{\text{actual}} \times 100\%. \tag{10}$$

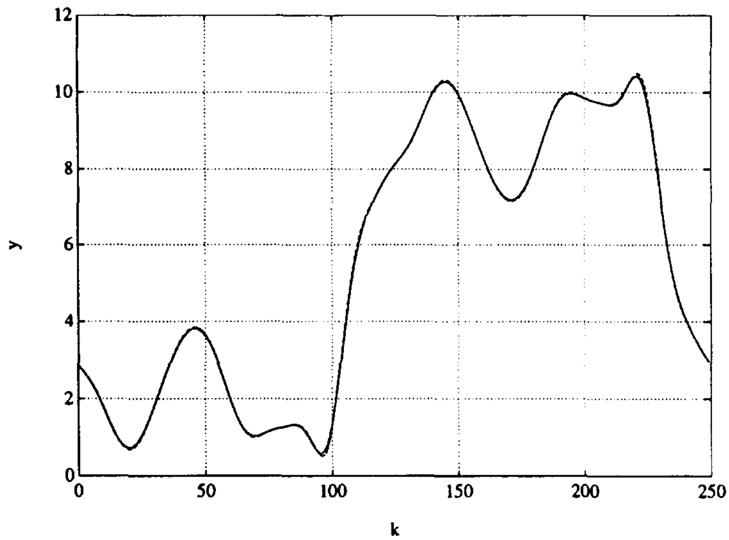

Fig. 6. The outputs from the actual, the GM(1, 1), and our models for Example 1 when only the first 150 data points are known. Note that there are three curves in this figure.

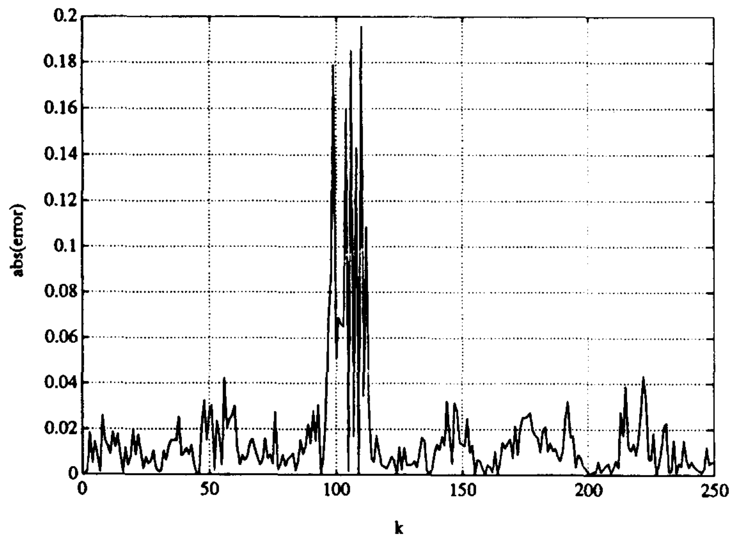

Fig. 7. The predicted error from our model for Example 1 when only the first 150 data points are known.

Figs. 7, 9 and 11 demonstrate the predicted errors from our model. As can be seen from these results, there exist some peak predicted errors. These all happen at the interval where it cannot find some related patterns from the previous intervals. However, if more data are known in advance, those peak predicted errors will disappear (this can be observed from Figs. 7, 9 and 11) and the predicted outcome from our model would become far more desirable.

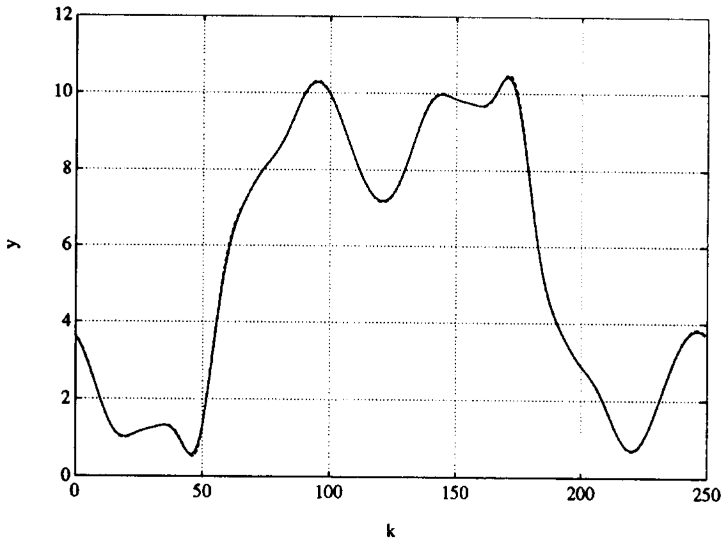

Fig. 8. The outputs from the actual, the GM(1, 1), and our models for Example 1 when only the first 200 data points are known. Note that there are three curves in this figure.

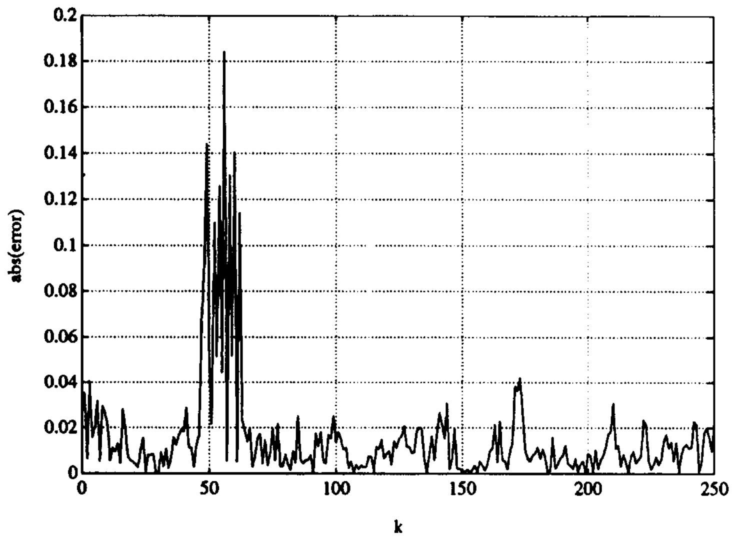

Fig. 9. The predicted error from our model for Example 1 when only the first 200 data points are known.

Example 2. The monthly average temperatures between January 1974 and December 1993 for the city of Taipei, Taiwan, are given in Fig. 12. We try to use the proposed model to predict the recently past two years temperatures, i.e., from Jan. 1992 to Dec. 1993, if only the past 18 years' data are known. Many factors may affect the temperature, but in this simulation we only use the past four months' temperatures to predict the coming one's. The predicted results are plotted in Fig. 13.

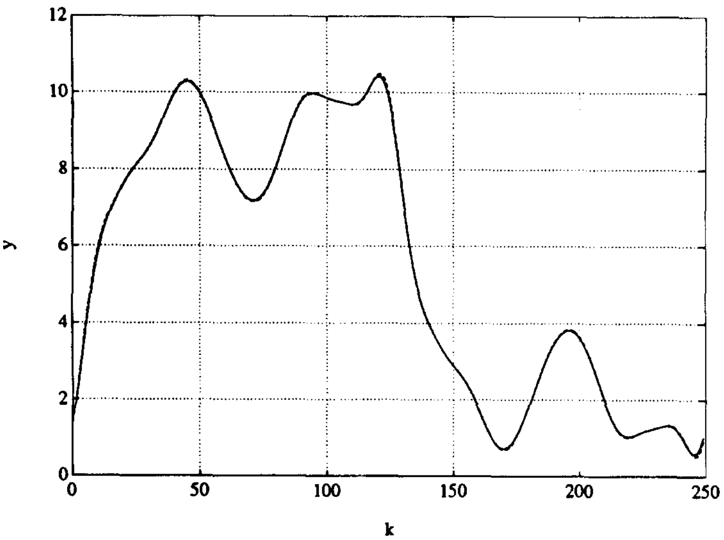

Fig. 10. The outputs from the actual, the GM(1, 1), and our models for Example 1 when only the first 250 data points are known. Note that there are three curves in this figure.

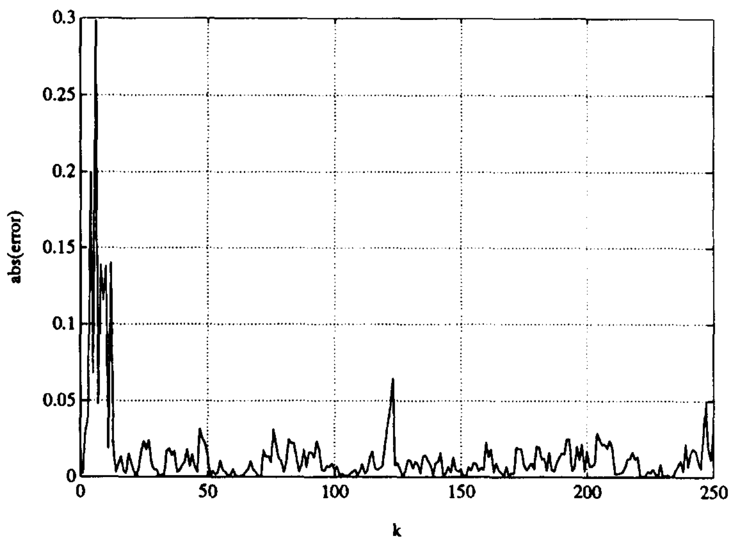

Fig. 11. The predicted error from our model for Example 1 when only the first 250 data points are known.

The original GM(1, 1) model shows an error of 10.157%, while our's has only 4.185%. This demonstrates that the proposed model shows a significant improvement over the conventional one. A detailed investigation found that most of the errors were caused by the month of Feb. 1993, when the weather was much warmer than January's (a 2.1 °C jump), adversely influencing the degree of accuracy in prediction. For the rest of the months, our model always follows the variation of the actual temperature.

Table 1
The predicted error percentages for Example 1 when the known data points are up to 150, 200, and 250, respectively

Known data	GM(1, 1) model (%)	Modified model (%)
150	1.115433	0.720618
200	1.115455	0.668849
250	1.115453	0.502268

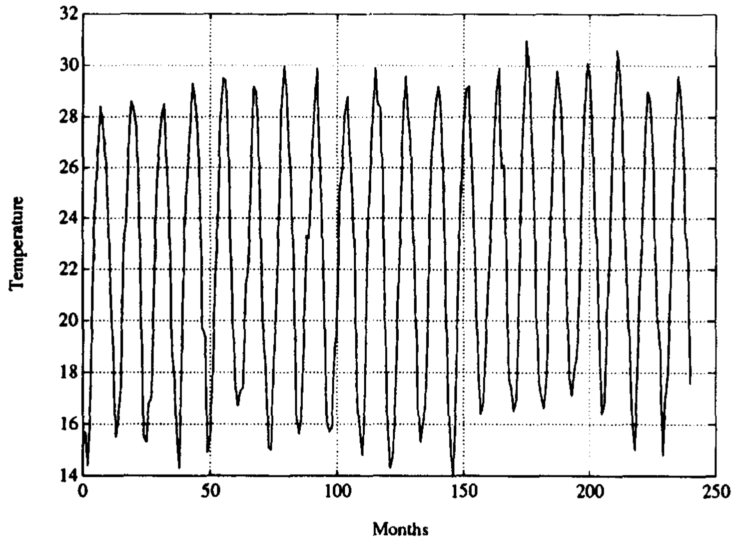

Fig. 12. The monthly average temperatures for the city of Taipei, Taiwan from Jan. 1974 to Dec. 1993. There are 240 data points.

7. Conclusions

The integrated technique for the fuzzy and the grey modeling methods is proposed. To further improve the GM(1, 1) prediction capability, we present a modified model. This model selects some most related patterns to the current one to decide how to compensate for the predicted result. Based on the selected patterns, a fuzzy model is formed through the proposed hybrid learning algorithm. The quantity of compensation is derived from the well-established fuzzy model. As a result, we can supplement the output predicted by the traditional GM(1, 1) model. The proposed methodology not only preserves the fast learning characteristic but also has a fascinating prediction capability. Simulation results from both the difference equation and the weather forecast demonstrate the applicability and effectiveness of the proposed model. Further works can focus on how to improve the peak prediction errors and how to incorporate other factors from the previous data into the current pattern to further improve the accuracy.

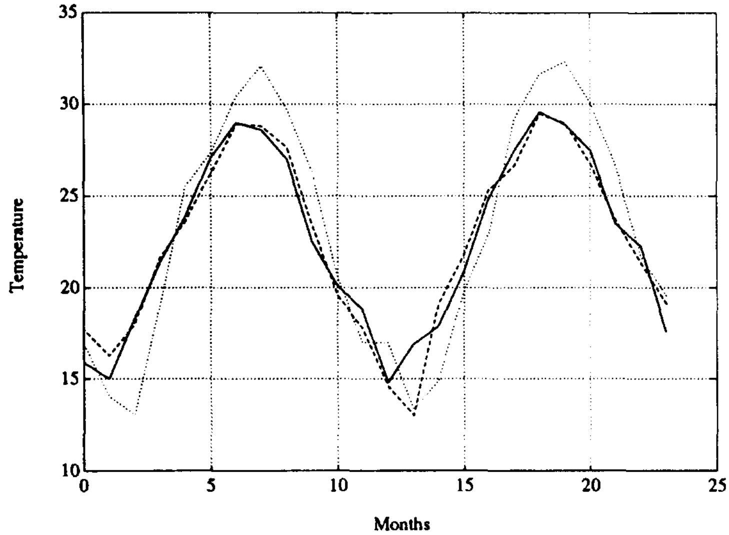

Fig. 13. The forecast outputs between Jan. 1992 and Dec. 1993 for Taipei. In this figure, the first eighteen years data are known. Solid, dotted, and dashed curves represent the outputs from the actual, the GM(1, 1), and our models, respectively.

References

[1] J.L. Deng, Introduction to grey system theory, *J. Grey System* **1**(1) (1989) 1–24.

[2] J.L. Deng, Properties of multivariable grey model GM(1, *N*), *J. Grey System* **1**(1) (1989) 25–41.

[3] F. Guely and P. Siarry, Gradient descent method for optimizing various fuzzy rule bases, *Proc. FUZZ-IEEE* (1993) 1241–1246.

[4] Y.-P. Huang, C.-C. Huang and C.-C. Chang, A systematic approach to adjust the fuzzy set parameters, in *Proc. First Asian Fuzzy Systems Symp.*, Singapore (Nov. 1993) 1–6.

[5] Y.-P. Huang and C.-C. Huang, The applications of fuzzy and gradient descent methods to the grey prediction control, *J. Grey System* **7** (1995) 9–22.

[6] B. Kosko, *Neural Networks and Fuzzy Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1992).

[7] Y.F. Li and C.C. Lan, Development of fuzzy algorithms for servo systems, *IEEE Control System Mag.* **9**(3) (1989) 65–72.

[8] D. Nguyen and B. Widrow, The truck backer-upper: an example of self-learning in neural network, *IEEE Control System Mag.* **10**(3) (1990) 18–23.

[9] H. Nomura, I. Hayashi and N. Wakami, A learning method of fuzzy inference rules by descent method, *Proc. FUZZ-IEEE* (1992) 203–210.

[10] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Trans. Systems Man Cybernet.* **15**(1) (1985) 116–132.

[11] L.-X. Wang and J.M. Mendel, Generating fuzzy rules by learning from examples, *IEEE Trans. Systems Man Cybernet.* **22**(6) (1992) 1414–1427.

[12] L.-X. Wang and J.M. Mendel, Back-propagation fuzzy system as nonlinear dynamic system identifiers, *Proc IEEE Int. Joint Conf. Neural Networks* (1992) 1409–1418.