PERIODICA POLYTECHNICA SER. TRANSP. ENG. VOL. 27, NO. 1–2, PP. 113–119 (1999)

COMPARISON OF THREE DIFFERENT METHODS IN THE PREDICTION OF THE MATERIAL FLOW IN A MATERIAL HANDLING SYSTEM

Gábor BOHÁCS and Béla KULCSÁR

Department of Building and Materials Handling Machines
Technical University of Budapest
H–1111 Budapest, Bertalan L. u. 7-9, Hungary
Tel./Fax: 0036-1-463-3059
e-mail: bohacs-gabor@eagt.bme.hu, kulcsar-bela@eagt.bme.hu

Received: Dec. 10, 1999

Abstract

The damaged barcodes cause problems particularly in production and warehousing, because in many cases their identification with conventional barcode readers is not possible. Here we deal with only a special part of damaged barcode identification. In this paper a prediction system is presented, that analyses the regularities of the material flow. The application of such a system can be an effective help with the reconstruction of the incomplete visual information.

Keywords: barcodes, prediction, material handling systems.

1. Introduction

Nowadays the flexible manufacturing and logistic systems require real time processing of even complex information. This can be fulfilled only with the use of appropriate software and hardware tools. In our current research we are engaged in the reconstruction of incomplete information of barcodes in a complex material handling system. It seems to be necessary, because the information of the barcode comes also forward as control parameter for the material handling system, so missing information often causes functional problems (KULCSÁR, 1998). Barcode damaging happens particularly in the manufacturing and warehousing processes. The damage means that a part of the label becomes unrecognizable.

In this article we will be dealing with only a part of the results. In our current research we try to reconstruct the missing information with an image processing system based on a CCD matrix camera. Next we summarize the functioning of the system, that recognizes all the visual information. The experiments implemented in the laboratory of the Department of Building and Materials Handling Machines. The hardware was installed at a certain point of a roller conveyer system (*Fig. 1*). The unit loads with barcodes travel in the conveyer system. The requirements against the processing software are the following:

- The decoding of the information has to pass off quickly, in order that the

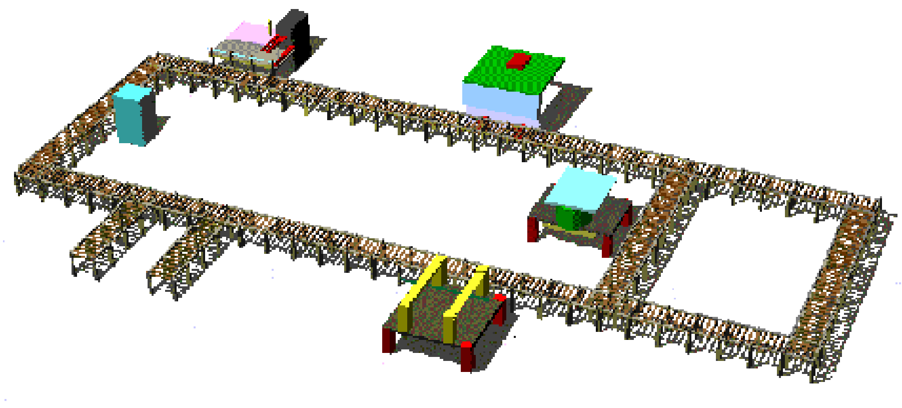

Fig. 1. The experimental system

information becomes available before the unit reaches the next branch in the conveyer system.

- If a character of the code damaged so much that more than one character suits to the damaged one, then the output must consist of all the codes that fit.

The implemented system using fuzzy inference fulfils all the requirements above.

So, in this paper we present a prediction system that helps the image processing system by the decoding of the information. Next the necessity of the prediction system will be described.

2. Necessity of the Predicting System

In this research the experimental conveyer system is regarded, as a material handling system in a manufacturing facility. The independently driven parts of the conveyer are marked with $G_1, G_2 \ldots$. The unit loads may travel in only one direction, and there can be only one unit load in each driven part. Certain points of the conveyer are regarded as workplaces (marked with $M_1, M_2 \ldots$) with deterministic process times. Let's implement various production programs in the system. Shortly after the beginning of the process, the flow of the unit loads becomes chaotic, because of the waiting times at a workplace, and the interlocking in the conveyer. For example if a product A has to be manufactured at M_2 but product B has not, then a unit B that follows unit A, will overtake A. That also means, that the CCD camera will observe a chaotic material flow. Despite it, because of the fixed process and transport times, there will be some regularity in the material flow. That makes reasonable the using of a prediction system. The output of the prediction is the most likely code for the following unit load. That output will be taken in consideration by the determination of the output of the image processing system.

Therefore the reliability of the whole system improves. Certainly the output of the image processing system is always “stronger” than the prediction.

The training data creation for the prediction system will be produced using a Siman simulation software. The codes are three character long Code 39 barcodes that change after every production step. The first character is the product identifier, the second is the destination code and the third determines how many production steps have been completed. The codes of the unit loads are the members of the time series used by the prediction models.

In this paper we consider three different production programs:

1. Only one product is manufactured (maximal number of unit loads in the system: 7, number of production steps: 4).
   Production sequence: $M_2 - M_4 - M_3 - M_1$
2. Two products are manufactured (maximal number of unit loads in the system: 6, number of production steps: 5).
   Production sequence for product 1: $M_1 - M_4 - M_3 - M_1 - M_3$
   Production sequence for product 2: $M_3 - M_1 - M_2 - M_4 - M_3$
3. Three products are manufactured (maximal number of unit loads in the system: 10, number of production steps: 3).
   Production sequence for product 1: $M_3 - M_4 - M_1$
   Production sequence for product 2: $M_3 - M_2 - M_4$
   Production sequence for product 3: $M_1 - M_2 - M_3$
   Production sequence for product 4: $M_2 - M_1 - M_4$

Next we survey the different models used by the prediction.

3. Predicting with AR Models

The autoregressive schemes (KENDALL – ORD, 1990) are a very important class of linear time series models. It has been very successively used in the time series forecasting.

A p^{th}-order scheme may be written as follows:

$$y = \delta + a_1 y_{t-1} + a_2 y_{t-2} + \ldots + a_p y_{t-p} + \varepsilon_t. \tag{1}$$

The model is denoted with $AR(p)$, and the $a_1 \ldots a_p$ coefficients are the autoregressive coefficients. δ is a constant term, and ε_t is called the error term. The goal is the determination of the autoregressive coefficients. For these time series we used the standard least squares estimate method (JANG, 1997). The advantage of this method is the relatively low computing time. The only question is, which order to choose for the model. It depends certainly on the process itself. There are methods based on the autocorrelation and partial autocorrelation functions to estimate the order of the model. We have tried to determine the order this way, but the obtained functions were not appropriate for this purpose. That gave an early warning that the AR models will not suit very well to the time series.

4. Predicting with ANFIS

The ANFIS is a very well known neuro-fuzzy network (JANG, 1993). The modelling of various nonlinear systems has very successively employed it. This network implements fuzzy if-then rules with adaptive parameters. The ANFIS structure is shown in *Fig. 2*. In this example network, there are three different barcodes: K_1, K_2, K_3, four arbitrary chosen fuzzy rules, and two input variables: f_{t-1}, f_{t-2} (the previous two observations). The node functions for this case are described below:

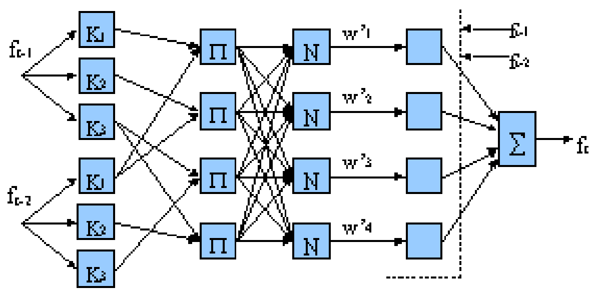

Fig. 2. ANFIS structure

Layer 1 determines the values of the membership function. In this case the membership function (μ_A) has only two values:

$$\mu_{ij} = 1, \quad \text{if} \quad f_{t-i} = K_j \quad \text{and} \quad \mu_{ij} = 0,1 \quad \text{otherwise}. \tag{2}$$

Layer 2 multiplies the incoming signals and sends the product out. These products are called 'firing strengths' of the rule. Layer 3 implements normalization: each firing strength is divided by the sum of all firing strengths. The node function of Layer 4 for the ith node:

$$O_i^4 = w_i'(p_i f_{t-1} + q_i f_{t-2} + r_i). \tag{3}$$

The parameters (p_i, q_i, r_i) are called consequent parameters. In this case the optimizing will change these parameters. Layer 5 computes the sum of the overall output. The training of the parameters is carried out as batch learning, using the sequential formulas of the least squares estimation (JANG, 1997).

5. Predicting with CAN

The CAN (Connection Analyzing Network) is an own developed one, that is very suitable for predicting time series containing complex information. The architecture is shown in *Fig. 3*. In this case the nodes correspond to the certain barcodes passing by the camera. Every node implements both input and output functions. The 'h_{xy}'

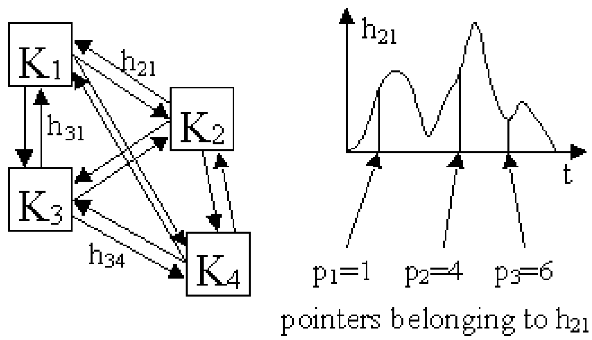

Fig. 3. CAN structure

parameters are functions (called: weight function) and not real values as usual by the neural networks. The system also uses ‘pointers’. The pointer points to a value of the ‘t’ axis of a weight function. The value of a pointer increases from step by step $(0, 1, 2 \ldots)$. The training can be carried out both off line and on line way. The i^{th} step of the training consists of the following procedures (in this step $f_i = K_x$, maximal node number: N):

1. Creation of a new pointer for all h_{xy} $(y : 1..N)$ functions.
2. Increase of the values of the h_{yx} $(y : 1..N)$ weight functions by one at the points shown by the belonging pointers.
3. Increase of the value of all pointers by one.

The 1st and 3rd step of the ith step of testing is identical with the training phase. In the second step, the output of each ‘x’ node is computed by summing the values of the h_{yx} $(y : 1..N)$ weight functions at the points shown by the belonging pointers. The output of the whole network corresponds to the identifier with the biggest output value.

6. Experimental Results

We have analyzed each production program with all three methods. In order to compare the results of the various methods we use the ‘number of correct answers’ (NOCA). The NOCA increases step by step if: round(output$_{\text{method}}$) $= f_{\text{actual}}$, otherwise remains unchanged. In words the NOCA counts, how many times is the difference between the output and the actual code smaller than for any of the other codes.

The results are summarized in *Table 1*.

The AR models give good results only for the simplest case (1st program), therefore it doesn’t seem to be suitable for predicting so complex systems. Otherwise the result of the AR scheme depends very much on the order of the model.

Table 1. Experimental results

Production program number	Number of obser-vations	NOCA AR	NOCA ANFIS	NOCA CAN
1	2508	2231	2114	**2246**
2	577	135	**559**	521
3	685	123	140	**316**

Fig. 4 shows the NOCA for the 1. production program. It can be seen, that a higher order AR model may give worse results than a lower one. We could not find any connection between the art of the process and the order of the model.

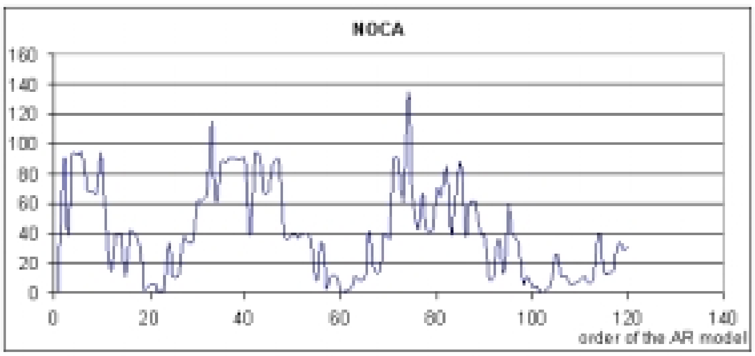

Fig. 4. NOCA for different order AR models

The ANFIS models give very good results for the first two cases. That makes this method suitable for this research. The only disadvantage of the implementation of this system is that the automatic construction of such a network is not possible. Let's suppose that the number of occurring barcodes is 7, and we would like to regard the previous 20 observations. That would make 7^{20} rules by automatic rule creation. This great number of inputs is necessary, because a single change in the input has got a long-term influence for the following codes. Therefore before training only the most significant rules must be chosen. The number of these rules is certainly not obvious from the system structure. To determine these rules we need knowledge of the whole teaching data and a separate program. So the parameters of the ANFIS structures are as follows:

- Case 1: for 20 previous data: 21 rules
- Case 2: for 20 previous data: 18 rules
- Case 3: for 3 previous data: 52 rules

The third case is so chaotic, that even for a low number of inputs, the implementation of at least 52 rules is reasonable. So the subjective factor is unfortunately not avoidable.

In general we can say, that the best results are given by the CAN network. The good results are due to the new architecture (every node can be input and output as well), and the application of weight functions instead of single real numbers. It has also a big advantage: by the determination of the structure no subjective information is needed, the construction happens automatically. The CAN has also a further advantage, the computing time. The determination of weight function in CAN took only some minutes (for ANFIS the determination of parameters took 1.5-6 hours). Therefore the CAN network seems to be the most suitable for our further research, but its application for other purposes is also thinkable.

References

[1] JANG, J. (1993): ANFIS: Adaptive-Network-Based Fuzzy Inference system. *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. 23 No. 3 pp. 665–685.
[2] JANG, J. – SUN, C. – MIZUTANI, E. (1997): Neuro-fuzzy and Soft Computing, Prentice-Hall, ISBN 0-13-261066-3, Upper Saddle River.
[3] KENDALL, M. – ORD, K. (1990): Time Series, Edward Arnold, ISBN 0-85264-295-4, London.
[4] KULCSÁR, B. (1998): Ipari logisztika, *LSI Oktatóközpont A Mikroelektronika Alkalmazásának Kultúrájáért Alapítvány*, ISBN: 9635772424, Budapest.