Int J Adv Manuf Technol (2006) 29: 308–316
DOI 10.1007/s00170-005-2528-x

ORIGINAL ARTICLE

Jie Zhu · Joseph C. Chen

Fuzzy neural network-based in-process mixed material-caused flash prediction (FNN-IPMFP) in injection molding operations

Received: 10 May 2004 / Accepted: 20 December 2004 / Published online: 25 January 2006

Abstract This paper describes the development of a fuzzy neural network-based in-process mixed material-caused flash prediction (FNN-IPMFP) system for injection molding processes. The goal is to employ a fuzzy neural network to predict flash in injection molding operations when using recycled mixed plastics. Major processing parameters, such as injection speed, melt temperature, and holding pressure, are varied within a small range. The vibration signal data during the mold closing and injection filling stages was collected in real-time using an accelerometer sensor. The data was analyzed with neural networks and fuzzy reasoning algorithms, in conjunction with a multiple-regression model, to obtain flash prediction threshold values under different parameter settings. The FNN-IPMFP system was shown to predict flash with 96.1% accuracy during the injection molding process.

Keywords Accelerometer sensor · Flash · Fuzzy neural network · Injection molding process

1 Introduction

Injection molding is the principal process for converting raw plastics into products. Demand for injection-molded products such as TVs, VCRs, computer housing, glasses, automobile parts, office furniture, etc., has increased over the past several years [1]. In recent years, the market has expanded from low- and moderate-quality products to high-quality, precision-molded items, such as medical devices and automobile connectors [2]. To produce such precision parts, quality control is critical during the injection molding process.

Injection molding is a complex process with many factors that can affect part quality. Correct process control is critical for making identical parts within tight tolerances and for meeting quality standards. If any processing conditions are altered, defects will occur. Processing conditions, such as processing parameters and material selection, are major contributors to defective plastic parts.

One of the most common defects in injection molding is flash. Flash occurs when excess plastic material is extruded from the edges of a mold. Flash may be caused by changes in the processing conditions such as injection speed, melting temperature, clamping pressure, improper feeding materials or mold damage [3]. In many cases, the excess material must be trimmed manually, which lowers process efficiency. To reduce inefficiency caused by flash, in-process flash detection techniques are essential.

Many on-line or in-process control systems have been built to predict part defects in the injection molding process. Various sensors have been applied to such systems [4–6]; however, these sensors have various limitations in flash prediction. One such system, the accelerometer sensor, has several advantages, such as fixed voltage sensitivity, low impedance output, high resolution, easy installation, and low cost. An accelerometer can detect vibration signals during production in the mold closing, plastic injection filling, and packing phases. The accelerometer sensor has been reported to detect flash in the injection molding process in previous research [7]. In this research, the processing parameters were optimized and fixed. However, during the injection molding process, any variations in these parameters, such as injection speed, holding pressure, injection pressure, melting temperature, and clamping force, are considered to be the predominant causes of poor part quality [8]. Since injection molding is a highly complex process involving many factors, it is necessary to develop multivariate control strategies and intelligent systems that are able to relate flash to any signal changes from an accelerometer.

One multivariate control technology is artificial neural networks, which are an information processing technology inspired by the human brain and nervous system. Artificial neural networks model arbitrary input signals by adjusting internal network connections to minimize output and input signals. Artificial neural networks have been employed to develop intelligent sys-

J. Zhu · J.C. Chen (✉)
Iowa State University
221 I. ED. II, Ames, Iowa 50011-3130, USA
E-mail: cschen@iastate.ed
Tel.: +15-15-2948040
Fax: +15-15-2941123

tems for resetting injection molding parameters and detecting deficient parts [9–11].

One of the major advantages of artificial neural networks over traditional expert systems is their ability to learn automatically from examples [12]. Artificial neural networks have proven effective not only in process modelling, but also in part defect diagnosis. However, the mathematics underlying artificial neural networks are insufficient to capture uncertainty or vagueness associated with human cognitive processes, such as reasoning or decision-making [13]. Such uncertainty in the manufacturing process must be handled by another tool: fuzzy logic.

Fuzzy logic was introduced by Zadeh [14] in the 1960s as a means to model the uncertainty of natural language. It provides an inference morphology enabling approximate human reasoning capabilities to be applied to knowledge-based systems. The mathematics underlying fuzzy logic can capture the uncertainty and vagueness associated with human cognitive processes [15]. In research, fuzzy logic has been employed to search for an acceptable machine setting in an expert system for reducing defects in injection molding [16]. In industry, fuzzy logic has been widely used for controlling complex systems due to its simplicity, low cost, and easy maintenance.

However, because injection molding is a complex process, using fuzzy logic alone is not enough to obtain a critical component of intelligent control systems – a fuzzy rule bank generated dynamically from expert data. To accomplish intelligent control in complex processes, such as injection molding, researchers have been combining the learning capabilities of neural networks with the reasoning capabilities of fuzzy logic, resulting in hybrid systems called fuzzy neural networks [17]. The incorporation of the two approaches overcomes the limitations of each and leverages their advantages.

Lin and Lee [18] proposed a general neural-fuzzy model combining the neural network structure with learning ability and a fuzzy logic controller. This model was applied to simulate the control of a fuzzy car that automatically moves along a rectangular path. He et al. [19] developed a fuzzy-neural system for parameter resetting in injection molding. The system can predict the amount to be adjusted for each parameter toward reducing or eliminating the observed defects, drastically reducing production time and effort.

In previous research [7] of an in-process mixed material-caused flash prediction (IPMFP) system, all processing parameters were set at fixed values. In a real injection molding process, any variation of processing parameters will cause the vibration signals collected by the accelerometer to fluctuate and then change the response – flash threshold value θ. The purpose of this study is to create a fuzzy neural network-based in-process mixed material-caused flash prediction (FNN-IPMFP) system in which the change of injection speed, holding pressure, and melting temperature will be monitored and corresponding output will be calculated by an FNN-IPMFP system. Different flash threshold values θ will be compared with the real value of the average max-avg. ratio $\overline{\gamma_j}$ [7]. The FNN-IPMFP system will employ an accelerometer to collect vibration data during the mold closing and injection processes.

2 Structure of the FNN-IPMFP system

The structure of the FNN-IPMFP system is shown in Fig. 1. This system includes three major systems: a signal collecting system, a processing parameter system, and a decision-making system. For the signal collecting system, an accelerometer was

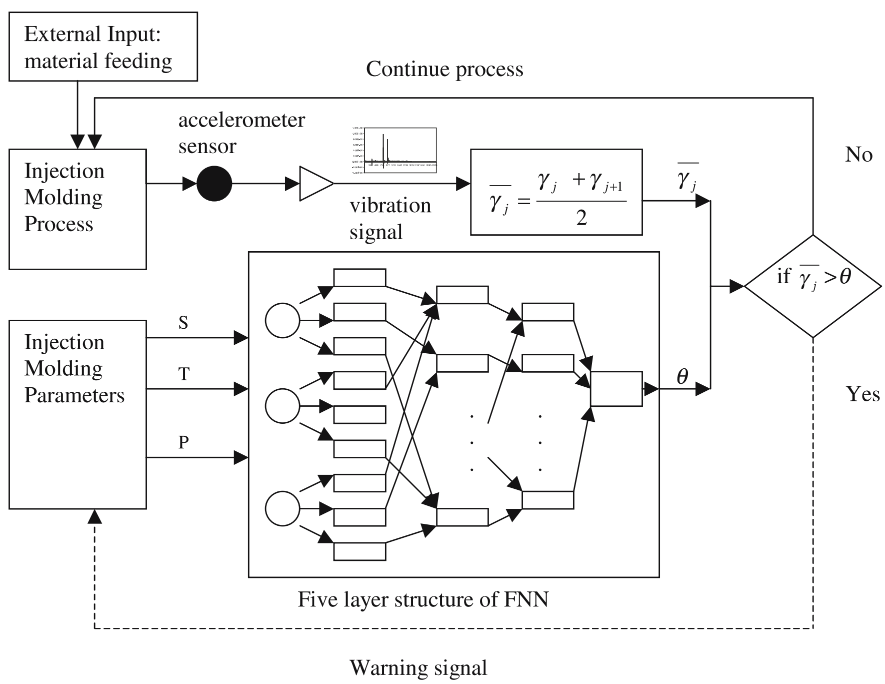


Fig. 1. Structure of the FNN-IPMFP system

chosen to collect real-time data. The accelerometer sensor measures real-time vibration and records the dynamic characters of the injection molding process. The second is the processing parameter system. Parameters that most affect the occurrence of flash, such as injection speed (S), melting temperature (T) of injection material, and holding pressure (P), are transmitted to the FNN-IPMFP system before or during machining. The third component is the decision-making system, which receives, converts, transforms, and calculates the values of the vibration signal from the accelerometer during the injection molding process. The average value of the max-avg. ratio $\overline{\gamma_j}$ is then compared with the FNN-calculated value θ. The system searches for significant differences and accurately predicts whether or not flash will occur. Warning signals are then sent to the machine operator. The experimental design is used to develop the IPFP system.

3 Experimental setup

To fully understand the IFPS system, it is necessary to design and build an experimental setup to evaluate its performance. The experimental setup consists of both hardware and software.

3.1 Hardware setup

The hardware system is shown in Fig. 2. As illustrated in the schematic, an in-process flash prediction system (IPFP) consists of a BOY 22M injection molding machine with Procan MD microprocessor control (BOY Machines Inc.); a personal computer; a 3-axis PCB accelerometer sensor, model No. 356B08, serial No. 6980 (PCB Piezotronics, 1994); a DBK11A Screw Terminal Expansion Card; and a DaqBook 100 data acquisition system (IOtech, 2000). The PCB accelerometer sensor was installed at the center of the top of the stationary mold and connected to the PC. The measured signals were connected to a battery-powered unit to be amplified and filtered. The X-axis orientation was in the extension of the mold surface, the Y-axis orientation was in the vertical direction, and the Z-axis orientation was along the moving direction of the movable mold.

The materials used in this research were Polystyrene (PS) 147F KG21 and low-density polyethylene (LDPE) 2072, all purchased from Prime Alliance. PS was considered the main recycled material in this research while LDPE was considered foreign material. The product made in this study was an injection-molded tensile bar.

3.2 Software setup

3.2.1 Data collection program

The software used to collect vibration data from the accelerometer sensor is DaqView 11.8 form IOtech, Inc (Ohio, USA). DaqView is a 32-bit, Windows-based data acquisition program that can be used to operate DaqBook100 series devices, DaqBoard series boards, and other models' products from IOtech. The data file is transferred to and compatible with Microsoft Excel.

In this system, the X-, Y-, and Z- directional vibration signals were measured by the PCB accelerometer while the mold was closed and the plastic material in the reciprocating screw was being injected into the mold under high pressure. These signals were transduced, amplified, collected, and then converted into digital data by DaqBook 100 in a personal computer. The Z-direction was our main research concern, as it followed the the mold closing and hitting direction.

3.2.2 The FNN- IPMFP training and predicting program

The C-based program is the major program in this research used to develop the rule bank for flash prediction, based on inputs of injection speed, melting temperature, and holding pressure. The procedure will be discussed in the next section.

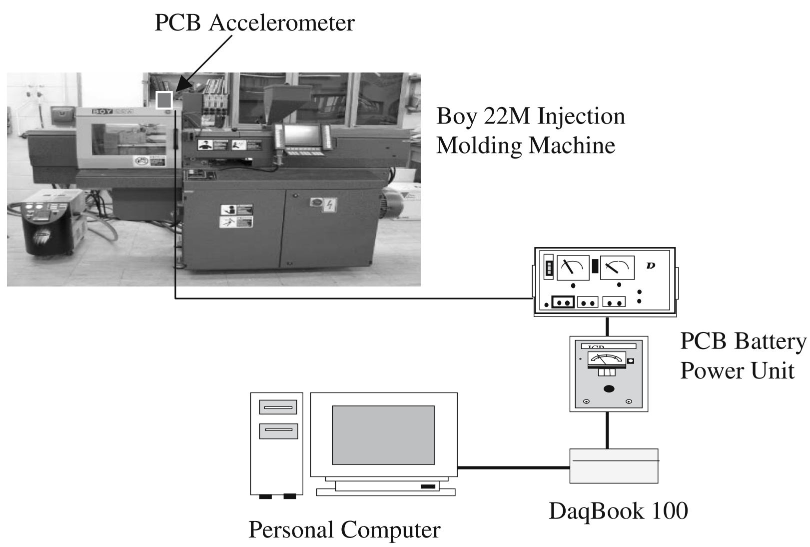


Fig. 2. Experimental setup

4 Experimental design

After the experimental setup was completed, the FNN-IPMFP system was ready to be evaluated. Using Dr. C-Mold software for injection molding parameter setting, processing parameters were set at different levels. The injection speed was set at three levels, melt temperature was set at two levels, and holding pressure was set at two levels. Pure PS was used as feeding material instead of mixed PS and HDPE. The 3x2x2 factorial experimental design is shown in Table 1. The product made in this research, shown in Fig. 3, was an injection-molded tensile bar that measured $4.95\,\text{in} \times 0.5\,\text{in}$. Fifteen injection-molded tensile bars were made for each situation.

Table 1. Training data of the FNN-IPMFP system

Run number	Injection speed (%)	Melt temperature (F)	Holding pressure (psi)	Control limit θ (based on $\overline{\gamma_j}$)
1	95	450	1100	5.69
2	95	450	900	5.81
3	95	430	1100	5.57
4	95	430	900	5.63
5	90	450	1100	5.60
6	90	450	900	5.75
7	90	430	1100	5.32
8	90	430	900	5.48
9	85	450	1100	5.50
10	85	450	900	5.56
11	85	430	1100	5.24
12	85	430	900	5.28

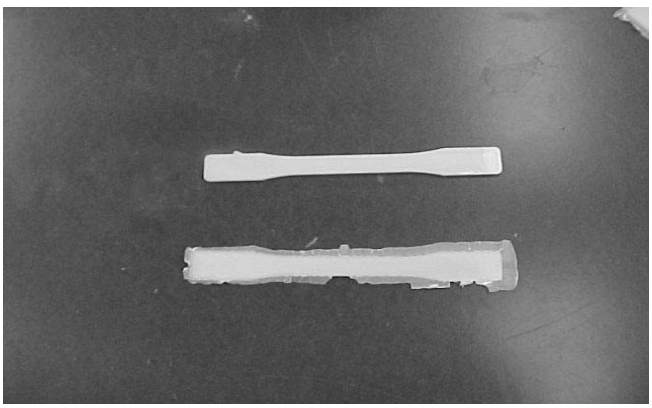


Fig. 3. Injection-molded tensile bar with and without flash

5 Development of FNN-IPMFP system

The procedure to develop the FNN-IPMFP system was based on previous research into the decision-making mechanisms of IPMFP and FNN-IPMFP systems. The procedure is as follows:

5.1 Eight steps of calculating vibration signal output: max-avg ratio $\overline{\gamma_j}$

Step 1 Starting from the point when the mold opens, collect 3000 Z-axis vibration data. This data collection covers vibration signals from the moment the mold opens until the end of the injection filling stage.

Step 2 Find the second peak of the Z-axis vibration signal, which represents the beginning of the injection filling stage.

Step 3 Starting from the second peak point, collect 850 Z-axis vibration points. (Z_{ij}, $i = 1, 2, \ldots, 850$, $j = 1, 2, \ldots,$ 20, where i denotes the data point and j denotes the product number used in this research.)

Step 4 Find the maximum absolute peak value $Z_{j\,\max}$ within the last 200 data points.

$$Z_{j\,\max} = Max\left|Z_{ij}\right| = Max\left\{\left|Z_{651j}\right|, \left|Z_{652j}\right|, \ldots, \left|Z_{850j}\right|\right\} \tag{1}$$

Step 5 Calculate the average absolute peak value of the 200 points.

$$\overline{Z_j} = \frac{\sum_{i=651}^{850}\left|Z_{ij}\right|}{200} \tag{2}$$

Step 6 Calculate the ratio of the maximum peak value over the average peak value. This value is called the max-avg. ratio γ, which has the following formula:

$$\gamma_j = \frac{Z_{j\,\max}}{\overline{Z_j}}, \quad \text{where } j \text{ is number of experiment} \tag{3}$$

Step 7 Save the max-avg. ratio as γ_j in the memory.

Step 8 Calculate the average of two consecutive max-avg. ratio data using the formula:

$$\overline{\gamma_j} = \frac{\gamma_j + \gamma_{j+1}}{2}\,. \tag{4}$$

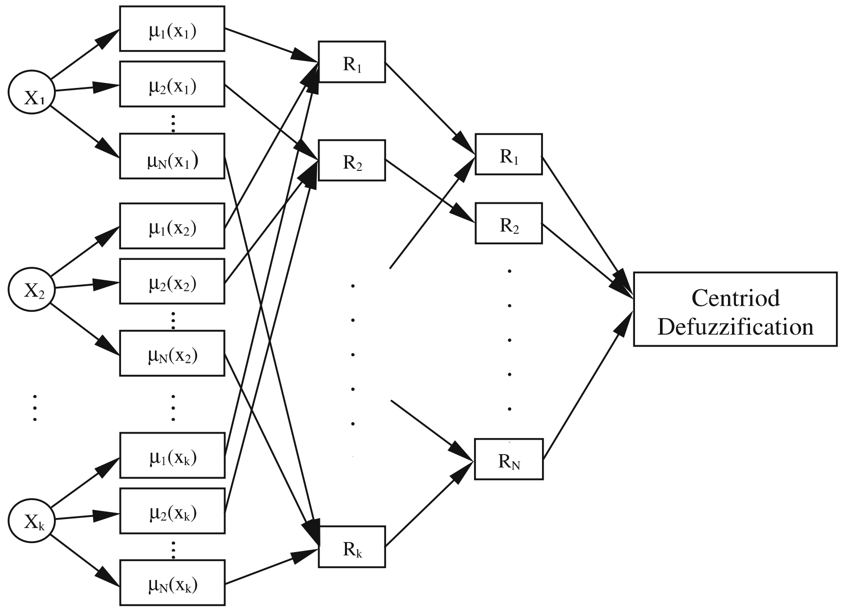
5.2 A five-step training scheme for the FNN-IPMFP system

The procedure for training the FNN-IPMFP system was a modified version of the fuzzy-nets five-step training procedure proposed by Chen [20] to define the fuzzy rule bank and the membership functions. Chen proposed a five-step training scheme in a five-layer FNN structure as shown in Fig. 4. The detailed procedures are summarized as follows:

Step 1: Divide both the input and output domains into fuzzy regions and create membership functions

A systematic methodology has been developed to divide all input and output variables into fuzzy regions, with each region represented by a membership function. Before constructing the FNN-IPMFP system, training data should be collected. Twelve experimental data sets (see Table 1), with the input of three levels

Fig. 4. The five layer structure of the fuzzy nets algorithm



of injection speed, two levels of melting temperature, two levels of holding pressure, and the output of control limit θ based on the average max-avg. ratio $\overline{\gamma_j}$, were collected (see Table 1).

The fuzzy domain for a certain input or output was defined as the space between the maximum value and the minimum value of the experiment training data for that input (output) plus a small "allowance" at each end (95% minimum and 105% maximum value were used in the fuzzy domain). For instance, the injection speed could be defined as $[S^-, S^+] = [79.8, 99.8]$, the melt temperature could be $[T^-, T^+] = [424.5, 472.5]$ Fahrenheit, and the holding pressure could be $[P^-, P^+] = [855, 1155]$ pounds per square inch. The output flash threshold value could be defined as $[\theta^-, \theta^+]$. The "+" means maximum value, while the "−" means minimum value of the applied domain intervals.

Each domain is divided into $2k+1$ overlapped regions, which are denoted by linguistic variables Sk, $S(k-1)$, ... ,M, ... ,$L(k-1)$, and Lk. The k value for all the domains was set to 1 at the beginning of the training process. Thus, all domains can be first divided into three regions: $S1$, $M1$, and $L1$, and then expanded to five regions: $S2$, $S1$, M, $L1$, $L2$, if necessary (See Fig. 5.) Each linguistic variable A (A is linguistic variable of $S1$, $M1$, or $L1$) is associated with a fuzzy set, each of which has a defined membership function μ_A. The membership function $\mu_A(x)$ gives the degree of membership of x in the region A.

Triangular functions can be used in this research to build fuzzy membership functions because it is one of the easiest shapes to use. The membership function $\mu_A(x)$ for region A can be expressed as:

$$\mu_A(x) = \begin{cases} 1 - \frac{x - C_A}{W}, & x \in [C_A, C_A + W] \\ 1 - \frac{C_A - x}{W}, & x \in [C_A - W, C_A] \\ 1, & x \in \left[-\infty, X^-\right] \text{ or } x \in \left[X^+, \infty\right] \\ 0, & \text{elsewhere} \end{cases} \tag{5}$$

where W is the spread width (defined as half of the base of the triangle), and $C_A$is the center point value of the membership function A.

As a result, the fuzzy degree of an input injection speed value of S_i for any fuzzy region A in the injection speed domain was given using Eq. 5.

$$\mu_A^S(S_i^{)} = \begin{cases} 1 - \frac{S_i - C_A^S}{W^S}, & S_i \in \left[C_A^S, C_A^S + W_S\right] \\ 1 - \frac{C_A^S - S_i}{W^F}, & S_i \in \left[C_A^S - W^S, C_A^S\right] \\ 1, & S_i \in \left[-\infty, X^-\right] \text{ or } x \in \left[X^+, \infty\right] \\ 0, & \text{elsewhere} \end{cases} \tag{6}$$

Using the same principle, the fuzzy degree of all membership functions in all input and output domains could be decided.

Step 2. Generate fuzzy rules for the given data

Since membership functions in a domain overlap one another, a specific value in a domain will usually produce two fuzzy degrees, one from each membership function. To reduce complexity, only one membership function will be used. Thus, this step contains three procedures:

1. Acquire the training data sets including the input variables and output response via experiments.
2. Retrieve the linguistic variables via the membership functions from step 1.
3. Specify the maximum strength of each input variable and output response, choosing the membership function with the larger fuzzy degree value.

Using the same example from Fig. 5a, if the injection speed is 88%, then two μ^S values will be produced: 0.4 from $S1$, and 0.6 from M. To reduce complexity, only one membership function was used. In this study, the membership function with the

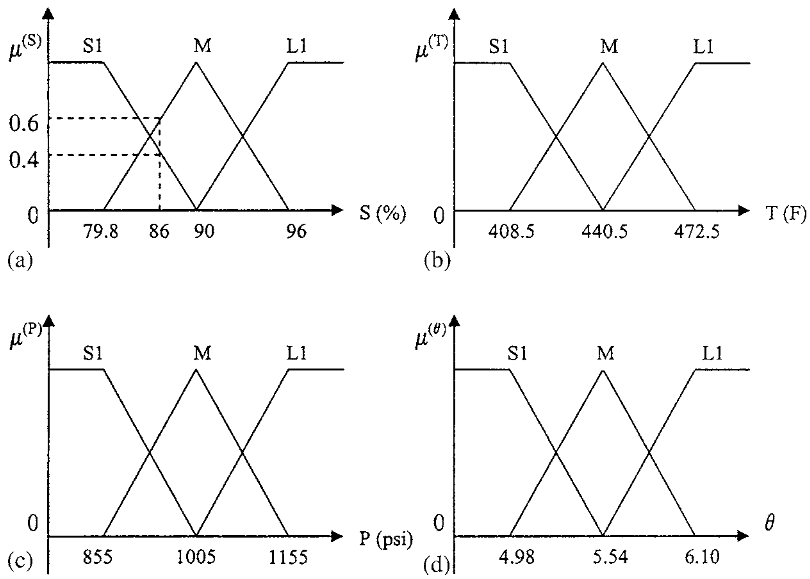


Fig. 5a–d. Membership functions of inputs and output (**a**) membership function of injection speed S; (**b**) membership function of melt temperature T; (**c**) membership function of holding pressure P; (**d**) membership function of output θ

larger fuzzy degree was chosen. In Fig. 7a, when $S = 88$, the M membership function was chosen. Therefore, every value from the experimental training data for every input or output issues a corresponding membership function.

Step 3. Solve conflicting rules

Since a large number of training data sets was used to produce fuzzy rules, it is possible that there are many conflicting rules (rules with the same "IF" condition but different "THEN" actions). To solve these conflicts, two general approaches are used: top-down and bottom-up. Initially, the top-down methodology is used because it's generally faster. Sometimes, however, the top-down methodology can't resolve the conflict. In this case, the bottom-up method is employed to resolve the process.

The top-down methodology works by assigning a degree (d) to each rule. The degree of the rule, "If S is M and T is M and P is L, then θ is S," is defined as:

$$d(rule) = \mu_M(S)\mu_M(T)\mu_L(P)\mu_S(\theta) \quad \mu_D \tag{7}$$

where μ_D is the condition degree, from 0 to 1, determined by a human expert based on the injection condition at the time the rule is produced. If nothing unusual happens during injection molding, then μ_D would usually be 1. An example of two conflicting rules (j and k) is:

Rule j: "If S is M and T is M and P is L, then θ is S." (8)

Rule k: "If S is M and T is M and P is L, then θ is M." (9)

To solve this conflict, if the magnitude of the deviation $|d(\text{rule k}) - d(\text{rule j})| > \delta$, where $0 < \delta < 0.5$, and δ is a user-defined value, then the rule with the maximum active value is chosen. Otherwise (i.e. $|d(\text{rule k}) - d(\text{rule j})| \leq \delta$), and the training is suspended. A bottom-up procedure is employed to resolve this problem.

The bottom-up methodology expands the number of fuzzy regions to decrease the fuzziness and increase the degree of discrimination in two conflicting rules. The rule of expansion is to add two more regions to one feature of the input domain. For example, S is set up initially for five regions. If the differential degree of rule j and rule k is less than δ, then S is extended to seven regions. Thus, all the previously trained input-output data-pairs must be retained. If any other rules conflict, two more regions must be added to the output feature. If the conflicts still exist, the number of regions of the next input feature and output feature are extended sequentially until all the conflicting situations are resolved.

In this research, all the fuzzy condition degrees (μ_D) of the data decided by the researchers of the study were 1. The value $\delta$was set to 0.3. With a C-based FNN-IPMFP training and predicting program, all the fuzzy rules resulting from the experiment could be created with all conflicts resolved. The final input domains of the injection speed and melt temperature were expanded to five regions. The output domain of the flash prediction threshold value θ was expanded to five regions. Only the holding pressure was set at three regions (see Fig. 6).

Step 4. Develop fuzzy rule bank

After all fuzzy rules had been generated and each of the conflicting rules resolved, a fuzzy rule bank was built by filling the rules into the cells. One rule will fill a cell, owing to the "AND" logic that has been applied. The rule bank structure was constructed with the antecedents of the rules. Since there were three antecedents in each rule (injection speed S, melt temperature T, holding pressure P), and if each input has 3 regions, then the rule bank is three-dimensional, measuring 5x5x3 in structure as shown in Fig. 7.

After the fuzzy rule bank was filled with the fuzzy rules from the training data sets, there were still many empty cells in the

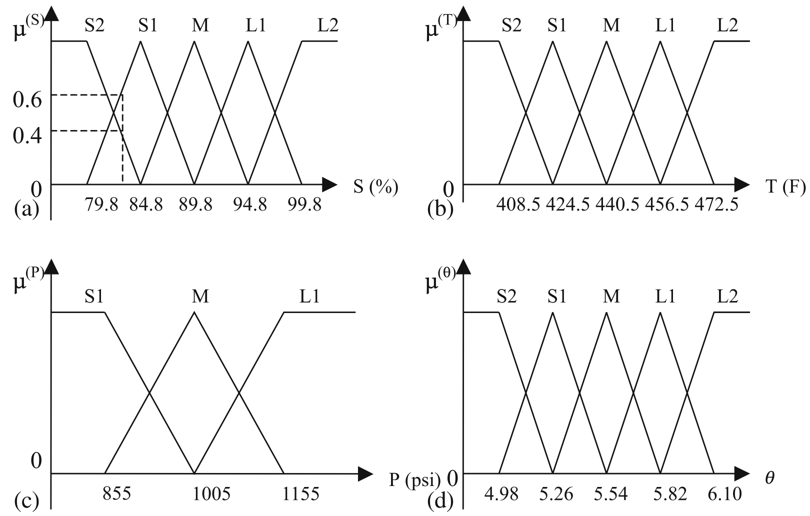


Fig. 6a–d. Final membership functions of inputs and output (**a**) membership function of injection speed S; (**b**) membership function of melt temperature T; (**c**) membership function of holding pressure P; (**d**) membership function of output θ

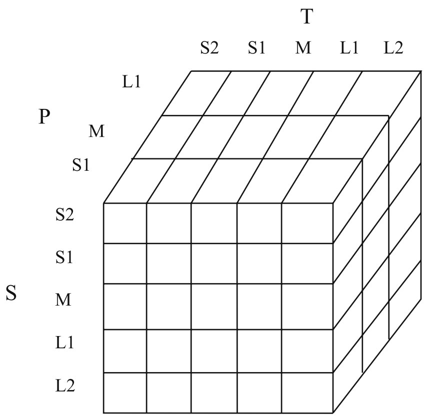


Fig. 7. The three-dimensional fuzzy rule bank

bank because of the limited number of experimental data sets. These empty cells were filled using a multiple-regression model to estimate the possible rules to complete the fuzzy rule bank structure. Using the training data from Table 1, the multiple-regression model was generated as:

$$\begin{aligned}\theta &= 9.884 + (-0.128) * (S) + (-0.0186) * (T) + (-0.0357) * (P)\\ &\quad + (0.0004) * (S * T) + (0.00042) * (S * P)\\ &\quad + (0.000084) * (T * P) + (-0.000001) * (S * T * P) \qquad (10)\end{aligned}$$

An example of filling an empty cell using the multiple-regression model is shown below. The cell of $\{S2, M, S1\}$ was empty. Replacing these linguistic variables with the center values $\{79.8, 440.5, 855\}$, the estimated flash prediction threshold value was shown as:

$$\begin{aligned}\theta &= 9.884 - 0.128 * 79.8 - 0.0186 * 440.5 - 0.0357 * 855\\ &\quad + 0.0004 * 79.8 * 440.5 + 0.00042 * 79.8 * 855\\ &\quad + 0.000084 * 440.5 * 855 - 0.000001 * 79.8 * 440.5 * 855\\ &= 5.25 \qquad (11)\end{aligned}$$

This crisp value was "fuzzified" by replacing it with a linguistic variable that represents the best-fit membership function of the threshold value for predicting flash. This value had membership functions in both $S2$ and $S1$ fuzzy regions of the flash prediction threshold with fuzzy degrees of 0.036 and 0.964, respectively. Since it has a higher membership in $S1$ than in $S2$, the linguistic variable $S1$ was chosen to fill the empty cell. Therefore, the original empty rule was filled with the rule "IF S is $S2$, T is M, and P is $S1$, THEN θ is $S1$". After all empty rule bank cells had been filled using the multiple-regression process the final rule bank was completely constructed. The entire rule bank, consisting 75 cells, is shown in Table 2.

Step 5. Defuzzification

The output θ from the fuzzy rule bank is a linguistic variable that is still fuzzy. To make it useful, the linguistic variables must be transferred into numerical values. This process is called defuzzification. Of the many different defuzzification methods, the centroid method was chosen in this study. Two steps are to be considered in this method.

First, for given inputs (S, T, P), the antecedents of the fuzzy rule select the minimum value from the fuzzy degrees of the input values. This value becomes the fuzzy degree of the output value θ, expressed as:

$$\mu^{j}_{output^{j}} = Min\left\{\mu^{j}_{input^{j}_{S}}, \mu^{j}_{input^{j}_{T}}, \mu^{j}_{input^{j}_{P}}\right\} \qquad (12)$$

where $output^{j}$ denotes the output regions of rule j, and $input^{j}$ denotes the input region of rule j of the input vector. Second, the predicted output value from defuzzification is calculated based on the following equation:

$$y = \frac{\sum_{j}^{k} \mu_{output}(\theta_j) C(\theta_j)}{\sum_{j}^{k} \mu_{output}(\theta_j)} \qquad (13)$$

Table 2. The complete fuzzy rule bank for the FNN-IPMFP system

Injection Speed	Hold Pressure	Temperature S2	S1	M	L1	L2
		Flash threshold value θ (75 cells)				
S2	S1	S2	S2	S1	M	L1
S2	M	S2	S2	S1	M	L1
S2	L1	S2	S2	S1	M	L1
S1	S1	S2	S1	M	L1	L1
S1	M	S2	S1	S1	M	L1
S1	L1	S2	S1	M	M	L1
M	S1	S1	M	L1	L1	L2
M	M	S2	S1	M	L1	L1
M	L1	S2	S1	M	M	L1
L1	S1	S1	M	L1	L1	L2
L1	M	S1	M	M	L1	L1
L1	L1	S1	M	L1	M	L1
L2	S1	M	L1	L1	L2	L2
L2	M	M	M	L1	L1	L1
L2	L1	M	M	M	L1	L1

where $C(\theta_j)$ denotes the center of the output region, $ou\,put^j$, and k is the number of adjacent fuzzy rules in the combined fuzzy rule-base.

5.3 Evaluation of the FNN-IPFP system

Following the above five steps, the C-based FNN-IPMFP system was then developed. To evaluate the system's performance, testing experiments were conducted. Testing was completed using pure PS material and mixed PS with 5% LDPE. The testing design is shown in Table 3. In each test, 15 products were collected and corresponding flash signal data recorded. After completing the testing, the flash prediction threshold values from the FNN-IPMFP system were compared with the calculated values to determine the existence of flash.

From Table 3, under a pure PS situation, no flash was found among all 60 products. Only four specimens were found having higher $\overline{\gamma_j}$ than the flash threshold value θ, which indicates flash. The testing accuracy was 95.6% in this case. Under a mixed-material feeding situation, all products had flash. However, three products had lower $\overline{\gamma_j}$ than the flash threshold value θ. The testing results showed 96.7% accuracy in this case. In summary, the FNN-IPMFP system could efficiently predict flash with 96.1% accuracy.

6 Conclusions

A fuzzy neural network-based in-process, mixed material-caused flash prediction (FNN-IPMFP) system for injection molding processes was developed and examined in this study. The system was shown to predict flash during the injection molding operation. The main conclusions drawn from this research are summarized as follows:

1. The FNN-IPMFP system predicted flash caused by mixed material with 96.1% accuracy.
2. Use of neural networks and fuzzy reasoning algorithms, in conjunction with a multiple-regression model, made the FNN-IPMFP system easier to use.
3. The FNN-IPMFP system generated accurate flash threshold values and efficiently predicted flash when major processing parameters, such as injection speed, melt temperature, and holding pressure varied within a small range.

This research is limited to only two types of plastic materials and one type of injection mold. Enlarging this system to include more materials and various molds for workpieces could provide greater applicability to future automated machining processes and implementation in industry.

References

1. Richardson TL, Lokensgard E (2003) Industrial plastics, theory and applications, 4th edn. Thomson Delmar, Clifton Park, New York
2. Thyregod P (2001) Modelling and monitoring in injection molding. Dissertation, Technical University of Denmark

Table 3. The testing of the FNN-IPMFP system (Nos. 1-6: pure PS feeding; nos. 7-12: mixed PS and 5% LDPE feeding)

Testing No.	S (%)	T (F)	P (psi)	FNN-based θ	System prediction result # of flash	# of non-flash	Actual result # of flash	# of non-flash
1	94	455	1120	5.64	0	15	0	15
2	94	455	920	5.55	1	14	0	15
3	94	435	1120	5.76	0	15	0	15
4	94	435	920	5.58	2	13	0	15
5	89	455	1120	5.56	0	15	0	15
6	89	455	920	5.37	1	14	0	15
7	94	455	1120	5.64	15	0	15	0
8	94	455	920	5.55	14	1	15	0
9	89	455	1120	5.56	15	0	15	0
10	89	455	920	5.37	14	1	15	0
11	84	435	1120	5.52	14	1	15	0
12	84	435	920	5.27	15	0	15	0

Total number of testing $= 180$,
Total number of testing error $= 7$,
FNN-IPMFP system accuracy $= (180-7)/180 = 96.1\%$

3. Osswald TA, Turning LS, Gramann PJ (2001) Injection molding handbook. Hanser Gardner, Cincinnati
4. Bur AJ, Thomas CL (1997) Fluorescence monitoring of polymer injection molding: mold development. Polym Eng Sci 37:1430–1441
5. Edwards R, Thomas C (2001) On-line measurement of polymer orientation using ultrasonic technology. Polym Eng Sci 41:1644–1653
6. Gonnet JM, Guillet J, Raveyere C, Assezat G, Fulchrion R, Seytre G (2002) In line monitoring of the injection molding process by dielectric spectroscopy. Poly Eng Sci 42:1171–1180
7. Zhu J, Chen JC, Zhang Z (in press) In-process mixed materials caused flash prediction (IPMFP) system in injection molding process. Int J Manuf Sci Technol
8. Woll LB, Cooper DJ (1997) Pattern-based closed-loop quality control for the injection molding process. Poly Eng Sci 37:801–821
9. Rewal N, Tonich D, Friedl C (1998) Prediction part quality in injection molding using artificial neurals. J Inject Mold Technol 2(3):109–119
10. Cho GH, Lee KD, Chang N, Kim SG (1994) Optimization of process parameters of injection molding with neural network application in a process simulation environment. Ann CIRP 43:449–452
11. Demirci HH, Coulter JP (1994) Neural network based control of molding processes. J Mater Process Manuf Sci 2(3):335–354
12. Chen JC, Lin NH (1997) Optimizing the fuzzy-nets training scheme using the Taguchi parameter design. Int J Adv Manuf Technol 13: 587–599
13. Zadeh LA, Tsoi AC (1976) Learning fuzzy inference systems using an adaptive member-ship scheme. IEEE Trans Syst Man Cybern 26(2): 326–331
14. Zadeh LA (1968) Fuzzy sets. Inf Control 8:338–353
15. Lotfi A, Tsoi AC (1996) Learning fuzzy inference systems using an adaptive membership scheme. IEEE Trans Syst Man Cybern 26(2): 326–331
16. Tan KH, Yuen MF (1995) An expert system for injection molding defect correction. In: Proceedings of the 3rd international conference: computer integrated manufacturing, volume 2, 1995
17. Gupta MM, Rao DH (1994) On the principles of fuzzy neural networks. Fuzzy Sets Syst 61:1–18
18. Lin CT, Lee CSG (1991) Neural-network-based fuzzy logic control and decision system. IEEE Trans Comput 40(12):1320–1336
19. He H, Zhang YF, Lee KS (2001) Development of a fuzzy-neural system for parameter resetting of injection molding. J Manuf Sci Eng 123:110–118
20. Chen JC (2000) An effective fuzzy-nets training scheme for monitoring tool breakage. J Int Manuf 11(1):85–101