

## Cause and Effect Prediction in Manufacturing Process Using an Improved Neural Networks

Nazri Mohd Nawi<sup>#</sup>, Noorhamreeza Abdul Hamid<sup>#</sup>, Noor Azah Samsudin<sup>#</sup>, Zawati Harun<sup>\*</sup>,  
Mohd Firdaus Ab Aziz<sup>#</sup>, Azizul Azhar Ramli<sup>#</sup>

<sup>#</sup>*Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, 86400, Johor, Malaysia*  
*E-mail: nazri@uthm.edu.my*

<sup>\*</sup>*Faculty of Mechanical and Manufacturing, Universiti Tun Hussein Onn Malaysia, 86400, Johor, Malaysia*

---

**Abstract**— The limitations of the existing Knowledge Hyper-surface method in learning cause and effect relationships in the manufacturing process is explored. A new approach to enhance the performance of the current Knowledge Hyper-surface method has been proposed by constructing midpoints between each primary weight along each dimension by using a quadratic Lagrange interpolation polynomial. The new secondary-weight values, generated due to the addition of midpoints, were also represented as a linear combination of the corresponding primary/axial weight values. An improved neural networks in learning from examples have also been proposed where both of the proposed algorithms able to constrain the shape of the surface in two-dimensional and multi-dimensional cases and produced more realistic and acceptable results as compared to the previous version. The ability of the proposed approach to models the exponential increase/decrease in the belief values by using high-ordered polynomials without introducing ‘over-fitting’ effects was investigated. The performance of the proposed method in modelling the exponential increase/decrease in belief values was carried out on real cases taken from real casting data. The computed graphical results of the proposed methods were compared with the current Knowledge Hyper-surface and neural-network methods. As a result, the proposed methods correctly predict the sensitivity of process-parameter variations with the occurrence of a defect and very important area of research in a robust design methodology.

**Keywords**— backpropagation algorithm; hypersurface method; activation function; Lagrange interpolation; search direction

---

### I. INTRODUCTION

Manufacturing has evolved drastically since the introduction of the intelligent system in the machine. The new emerge of technology had shown that intelligent manufacturing had become one of the most promising and quickly developed fields of today's science and technology. The goal for intelligent manufacturing is satisfying customer needs to the highest standard, for the lowest possible cost by incorporating computer technology and introducing human-like decision-making capabilities into the manufacturing system. The manufacturing industries face difficulties in developing a new paradigm to cope with ever-changing consumer preferences and tastes, which results in shorter and shorter product life cycle. These difficulties increase the globalisation of manufacturing, as a cheap labour force is available in Eastern countries like China and India and classical manufacturing systems are not capable of satisfying all the needs of the global market [1].

Future manufacturing process needs to have an ability to automatically and continuously adapt production resources

and processes in an optimal way with respect to business and production objectives as well as market and technical conditions. These adaptive production systems should integrate innovative processes, overcome existing process limitations, handle the transfer of manufacturing know-how into totally new manufacturing-related methods and also adapt to the existing manufacturing equipment and resources in order to implement changes related to radically new technologies. This is indeed a vision shared by the European Commission and formulated in the recently announced Framework 7 program [2].

Delivering reliable, high-quality casting products and processes at low cost has become the key to the survival of foundries in the twenty-first Century. Driven by the need to compete on cost and performance, many quality conscious organisations are increasingly focusing on employing optimisation methods and numerical simulation technologies to improve their product quality with lower cost and reach the desired result as quickly as possible.

The quality, productivity and costs of components manufactured by most of the casting processes are influenced by a large number of process controls, material

and design considerations. The analysis of cause and effect relationship is complex for many manufacturing processes and in most cases 'experience' is the only factor which can help to take corrective actions. Normally in the manufacturing industry, manufactured products are usually tested for quality and sub-standard products are rejected. During that process, the fault or faults are noted, and reasons for the occurrence of the faults are established so that the corrective actions can be taken. In this way, the chances of manufacturing sub-standard products thereafter are minimised. Such a diagnosis is usually performed by experts in the field, who have acquired a fundamental understanding of the process over years of experience in analysing cause and effect relationships. Experience takes time to gather and, when an expert leaves a particular industry, his expertise is also lost to that employer. To be competent in the nowadays modern manufacturing industry, the ability to learn causal relationship from diagnosis examples is extremely useful and demanding.

In an earlier work, Ransing [3] had proposed a method known as '*Knowledge Hyper-surface method*' that provided the industry with a self-learning decision-making tool, which can store the knowledge of current/past rejection levels within the manufacturing set up. The tool automatically learns a cause and effect relationship by using the diagnosis information provided by experts. Such learning ability can help managers not only to quantify the influence of causes of defects for existing products but also to set up a new process, material and design parameters to manufacture new, high-quality products. Furthermore, the method has also potential to assist industry in retaining some of the expertise when experienced staff either retire or leave the job.

The Knowledge Hyper-surface method retained the advantage of regression analysis and neural network techniques and at the same time overcome the limitations of each other for cause and effect relationship. The method describes that the belief variation in the occurrence of a cause, with respect to a change in the belief value of the occurrence of an effect, follows a pattern. It was observed that such a variation is generally either linear, quadratic or cubic and certainly not an arbitrary higher ordered polynomial.

The knowledge hyper-surface method used lower ordered, one-dimensional Lagrange Interpolation Polynomials to construct the multi-dimensional hypersurfaces. A number of equidistant reference points were chosen in the input space created by belief values representing the strength of the effects. A Lagrange Interpolation polynomial and a weight value are associated with each of the reference points. A weight value at a reference point is considered to be representative of the belief value in the cause. The reference points have been divided into two categories, referred to as primary and secondary reference points. Weight values associated with these primary reference points were considered as independent variables (primary weight values) and other weight values associated with secondary reference points (secondary weight values), have been considered to be linearly dependent on one or more primary weight values. However, the current methodology was unable to model exponential increase/decrease in belief values particularly in cause and effect relationships. Therefore, a strategy that is

computationally efficient and able to model the exponential increase/decrease in belief values in cause and effects relationships without introducing the side-effects of 'overfitting' is essential by introducing the capabilities of neural networks.

Over the past several years, many works have been done and proposed by previous researchers based on a few areas such as operations research, statistics and computer simulation. Moreover, control theory has been developed and applied to solve a wide spectrum of problems in casting. Nowadays the casting environment is characterised by its complexity and ever-growing demand for new tools and techniques to solve difficult problems. Therefore, neural network had been known for offering a new and intelligent alternative to investigate and analyze challenging issues related to manufacturing.

Part of this interest is due to some features of the Multi-Layer Perceptrons not found altogether in the techniques traditionally used for causal relationship analysis. Neural network is used to capture the general relationship between variables of a system that is difficult to relate analytically. Neural network has been described as 'brain metaphor of information processing' or as 'a biologically inspired statistical tool' [4]. It has the capability to learn or to be trained about a particular task, its computational capabilities and the ability to formulate abstractions and generalisations.

Neural network is used to learn patterns and relationship in data. Having to know the relationship in the data means that two or more factors work together to predict the model outcome. Neural networks are universal function approximators [5], a non-parametric system capable of mapping complex non-linear relations among explanatory factors (e.g., defects or input data) and the outcome (causes or output data) and achieve excellent generalisation capacity. Neural networks discover this non-linear relationship during training phase when the input and output data are repeatedly presented to the network. The output data are compared with the results calculated by the neural network and the difference, or the error is calculated via mathematical procedure, which adjusts the value of network parameters (such as weights, bias, etc.) in order to minimise the error.

In a neural network, weights are generally modified on the basis of the errors between desired and actual outputs in an iterative fashion, and one of the commonly used training algorithms is the '*Delta Rule*' [6]. Basically, the neural network learns the desired outputs by adjusting its internal connection weights by minimising the discrepancy between the actual outputs of the system and the desired outputs [7], [8], [9].

There are many alternative training methods and variants for neural networks. In the case of feedforward multilayer networks, the most successful algorithm was the classical backpropagation [10]. Although this approach is very useful for the training process of this kind of neural networks, it has its own drawbacks. One of the main drawbacks is that the training becomes inefficient and training takes too long when compared to the training algorithms in use today.

In order to solve these problems, several variations of the commonly used neural network algorithm and also new methods have been proposed. Focusing the attention on the problem of the slow learning speed, some algorithms have

been developed to accelerate it. In this research, a novel and efficient method for speeding up and improving the training efficiency of a backpropagation algorithm has been developed.

In order to verify the efficacy of the proposed method, some simulation experiments were performed on four selected benchmark problems. The remaining of the paper is organised as follows: In Section II, some discussion on the implementation of Lagrange Interpolation Polynomials into the current Knowledge Hyper-surface method and highlights the limitation of the current method in learning from examples. The enhancements to the current method by incorporating midpoints in the existing shape formulation were discussed in Section III. The experiments and simulation results are presented in Section IV. The final section contains concluding remarks and short discussion for further research.

## II. MATERIAL AND METHOD

The method proposed by Ransing [3] proposed a method that retains advantages of regression analysis and neural-network techniques and at the same time overcomes the limitations of both techniques. The Knowledge Hyper-surface method described that the belief variation in the occurrence of a cause, with respect to a change in the belief value of the occurrence of an effect, follows a pattern. Such a variation is generally linear, quadratic or cubic and certainly not an arbitrary higher-ordered polynomial.

The method described that to model an  $n^{\text{th}}$  order relationship along a dimension,  $(n+1)$  equidistant reference points between -1 and +1 are chosen. For each reference point ' $i$ ' ( $i=1$  to  $n+1$ ), a one-dimensional Lagrange Interpolation Polynomial is used based on the following formula:

$$l_i(\xi) = l_k^n(\xi) = \frac{\xi - \xi_0 * \xi - \xi_1 * \xi - \xi_2 * \dots * \xi - \xi_{k-1} * \xi - \xi_{k+1} * \dots * \xi - \xi_n}{\xi_k - \xi_0 * \xi_k - \xi_1 * \xi_k - \xi_2 * \dots * \xi_k - \xi_{k-1} * \xi_k - \xi_{k+1} * \dots * \xi_k - \xi_n} \quad (1)$$

where:

$n$  : Order of the Lagrange Interpolation Polynomial (e.g., one for linear; two for quadratic; three for cubic; etc.)

$k$  : A reference point at which the one-dimensional Lagrange Interpolation Polynomial  $l_k^n(\xi)$  is constructed ( $k$  ranges from 0 to  $n$ ).

$i$  : Ranges from one to the total number of reference points, i.e.  $(n+1)$ .

The variable  $\xi$  is used to store the belief value representing the strength of the corresponding effects, ranges from -1 to +1. For one-dimensional Lagrange Polynomial Interpolation, the reference points are drawn along this dimension. Whereas for a given cause connected to ' $p$ ' effects, the Lagrange Interpolation Polynomial at a reference point ' $i$ ' is

defined as ' $p$ ' dimensional and is given by the following equation:

$$l_i(\xi^1, \xi^2, \xi^3, \dots, \xi^j, \dots, \xi^p) = l_{k_1}^{n_1}(\xi^1) * l_{k_2}^{n_2}(\xi^2) * \dots * l_{k_j}^{n_j}(\xi^j) * \dots * l_{k_p}^{n_p}(\xi^p) \quad (2)$$

where:

$$l_{k_j}^{n_j}(\xi^j) = \frac{\xi^j - \xi_0^j * \xi^j - \xi_1^j * \xi^j - \xi_2^j * \dots * \xi^j - \xi_{k_j-1}^j * \xi^j - \xi_{k_j+1}^j * \dots * \xi^j - \xi_{n_j}^j}{\xi_{k_j}^j - \xi_0^j * \xi_{k_j}^j - \xi_1^j * \xi_{k_j}^j - \xi_2^j * \dots * \xi_{k_j}^j - \xi_{k_j-1}^j * \xi_{k_j}^j - \xi_{k_j+1}^j * \dots * \xi_{k_j}^j - \xi_{n_j}^j} \quad (3)$$

$n_j$  : The order of one dimensional Lagrange Interpolation Polynomial ( $l_{k_j}^{n_j}(\xi^j)$ ) corresponding to

$j^{\text{th}}$  dimension that represents the relationship between  $j^{\text{th}}$  effect and the cause under consideration.

$k_j$  : Reference point along  $j^{\text{th}}$  dimension, at which the one-dimensional Lagrange Interpolation Polynomial  $l_{k_j}^{n_j}(\xi^j)$  is evaluated.  $k_j$  Independently

ranges from 0 to  $n_j$  for each Lagrange Polynomial Interpolation).

$\xi_0^j, \xi_1^j, \xi_2^j, \dots, \xi_{n_j}^j$  are  $(n_j+1)$  reference points along the  $j^{\text{th}}$  dimension.

$i$  : for a ' $p$ ' dimensional case, ' $i$ ' ranges from one to the total number of reference points ' $q$ ' as given below:

$$q = (n_1+1) * (n_2+1) * (n_3+1) * \dots * (n_j+1) * \dots * (n_p+1) \quad (4)$$

The method also prescribed that a Lagrange Interpolation polynomial and a weight value can be associated with each of the said reference points as shown by the equation below:

The belief value in the cause

$$= \sum_{i=1}^q w_i l_i(\xi^1, \xi^2, \dots, \xi^p) \quad (5)$$

where:

$q$  : Total number of reference points.

$l_i(\xi^1, \xi^2, \dots, \xi^p)$  is given by Equation (1)

$w_i$  : Weight variable associated with the  $i^{\text{th}}$  reference point.

By considering a weight value at a reference point to be representative of the belief value in the cause, the total number of weights is, therefore, the same as the total number of reference points. However, this formulation had its own limitation. As the number of dimensions increased, the total number of weights in a network also increased exponentially. This rapidly increased the number of unknown variables within the network, and it was not a practical implementation, as it would not only slow down the system but also requires an excessively large training dataset.

In order to overcome that limitation, Ransing [3] divided the reference points into two categories, referred to as primary and secondary reference points. Weight values associated with these primary reference points have been considered as independent variables (primary weight values) and other weight values associated with secondary reference points (secondary weight values), have been considered to be linearly dependent on one or more primary weight values (see Fig. 1).

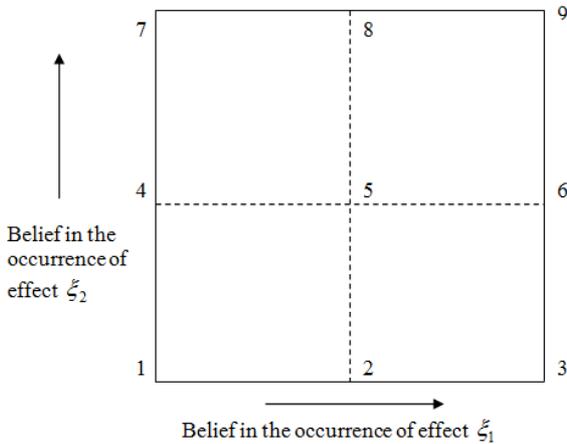


Fig. 1 Dependent (1,2,3,4,7) and Independent Weight Values (5,6,8,9) associated with Reference Points 1 to 9

For a ‘ $p$ ’ dimensional problem, the total number of primary weights is calculated as:

$$\text{Primary weights} = \left[ \left( \sum_{j=1}^p n_j + 1 \right) - (p-1) \right] \quad (6)$$

As we can see from Fig. 1, weights associated with primary reference points 1, 2, 3, 4 and 7 are primary weights. The secondary weight values at locations 5, 6, 8 and 9 are expressed as a linear combination of the primary weights and in particular:

$$w_5 = \frac{c(w_2 + w_4)}{2} \quad (7)$$

$$w_6 = \frac{c(w_3 + w_4)}{2} \quad (8)$$

$$w_8 = \frac{c(w_2 + w_7)}{2} \quad (9)$$

$$w_9 = \frac{c(w_3 + w_7)}{2} \quad (10)$$

The current method was capable *a priori* of storing any known information about the cause-effect relationship within the network and at the same time was able to learn from examples. For some selected datasets the proposed algorithm has shown superior extrapolation abilities as compared to the multi-layer neural network. The extrapolation ability was enhanced by the network’s ability to constrain the shape of the resulting multi-dimensional hyper-surface to the known variation in the belief values in causes and effects. The

dependence of the secondary weight values on the primary weight values had reduced the number of unknowns to an acceptable number.

Despite the superior extrapolation abilities of the current knowledge Hyper-surface method, two major limitations have been identified. First (1) the use of higher ordered polynomials can lead to the ‘over-fitting’ effect as observed in other interpolation techniques including neural networks. Second (2), an exponential rise in the belief value (as shown in Fig. 2) cannot be modelled by lower-ordered polynomials such as quadratic and cubic Lagrange interpolation polynomials.

To demonstrate the over-fitting effect, the following dataset is created by choosing a few data points, and then a maximum of twenty percent noise with a normal distribution with mean zero and unit standard deviation value is added randomly. The variations are plotted using linear-, quadratic- and quartic-shape functions to observe the performance of the current method as shown in Fig. 2.

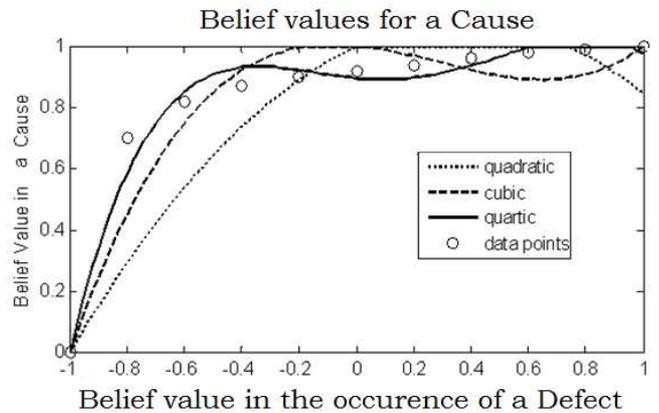


Fig. 2 Data points plotted with linear-, quadratic- and quartic-shape functions to demonstrate the over-fitting effect caused by the current Knowledge Hyper-surface method

Fig. 2 clearly shows that the use of quartic-shape functions in the current Knowledge Hyper-surface method had fitted all the data points perfectly as compared to the others, but the resulting shape of the decision hyper-surface is unrealistic and is a clear case of ‘over-fitting’ to the data points.

In order to overcome the current method from ‘over-fitting’ problem, Meghana [3] introduced an improvement by adding reference points between the end-and mid-reference points. The primary weights determined previously at end-and mid-reference points are kept constant and optimal values for the two new reference points ( $X_1$  and

$X_2$ ) are determined by a second-stage optimisation process using the current knowledge hyper method using fourth-ordered (quartic) Lagrange interpolation polynomials.

Furthermore, in solving the same problem from ‘over-fitting’, Nazri [11] also introduced a new method in neural networks by improving gain parameters in activation. The proposed method had significantly improved the back-propagation training algorithm. The detail of the proposed algorithm by Nazri can be referred to some papers [12], [13], [14].

### III. RESULTS AND DISCUSSION

The abilities of the proposed method by Meghana [3] and method by Nazri [11] in capturing the exponential change in the belief variation of the cause when the belief in the effect is at its minimum is compared with the outputs from both the current Knowledge Hyper-surface method on a real dataset. This dataset was also used by Ransing [3]. The data was collected from 'Kaye Prestigne'— a pressure die-casting foundry. A total of fourteen defects were identified and associated with forty-three process, material or design parameters. The data was collected for similar components over a period of one year. A total of sixty representative examples were finalised. For this case study as shown in Table 5.4, sixteen process parameters, three defects, and eleven examples were chosen. The same information was also used by Ransing [3].

A belief value in the occurrence of defects was calculated as corresponding to the belief values representing the occurrence and non-occurrence of associated process, design and material parameters as given by the experts in the foundry. Three defects known as 'Porosity', 'Mismakes' and 'Dimensional' are identified, and all defects chosen are represented as defects A, B, and C. For the purpose of comparison, the graphical variation of belief surfaces learnt by the neural network, the current method, and the proposed method are shown only on two defects which are 'Porosity' and 'Mismakes'. Sixteen associated process, material, and design parameters were identified to create a neural network with two input nodes corresponding to defects 'A' and 'B', and sixteen output nodes corresponding to the sixteen process, material, and design parameters. The belief values which were used in a training dataset are shown in Fig. 3.

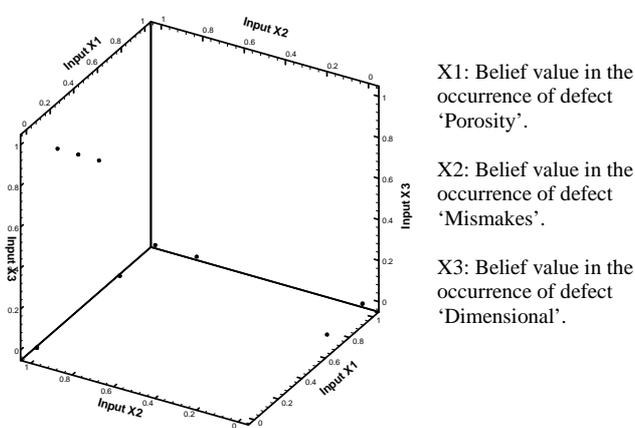


Fig. 3 Data points used in the training dataset

The proposed conjugate gradient neural-network method (CGPR/AG) [11] with five hidden nodes is constructed and trained on the training dataset with a learning rate equal to 0.4 and with a target error of 0.001. Since a neural network uses sigmoid activation function, the input data for the neural network was scaled between [0, 1]. A quadratic variation between input and output relationships was assumed in both the current method and the proposed method. Codes for all methods have been written in MATLAB.

All networks achieved the target error of 0.001 and seemed to have learnt the training dataset. The speed of all networks in learning the training dataset is not the main concern in this test, as the resulting shape of the hypersurface is of importance. The belief surface has been plotted for cause 'The position of gate' (cause number 8) which influences the occurrence of 'Porosity' (defect A) and 'Mismakes' (defect B) as this data requires to model the exponential rise in the belief values variation.

The variation in the belief value in the occurrence of 'The position of gate' for defect A, i.e. 'Porosity' using the current method and the proposed method is plotted when only defect A is connected to the cause (one-dimensional case) and when both defects (i.e., defects A and B) are connected to the cause (two-dimensional case).

The results are shown in Figs. 4 and 5. Since the proposed method is able to model an exponential increase in belief values, it was shown to be a better fit to data points using the quadratic polynomials as compared to the current method. This is because of the introduction of midpoints which gives an additional degree of freedom to control the resulting curve. Furthermore, Fig. 6 also demonstrates that the proposed neural networks showed a reasonable fit to these data points. However, as demonstrated by Ransing [3], the proposed neural networks do not guarantee a better shape for hyper surfaces. The proposed neural networks tend to interpolate better point and exhibit all the limitations as identified by Ransing[3].

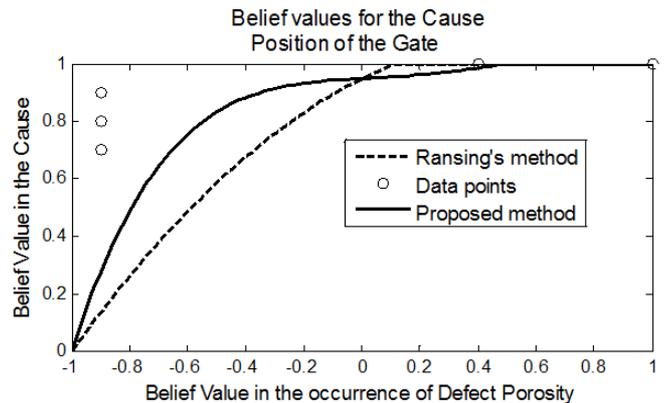


Fig. 4 The performance of Ransing's method and the proposed method for one- dimensional belief-value variation modelled by quadratic polynomials for defect Porosity.

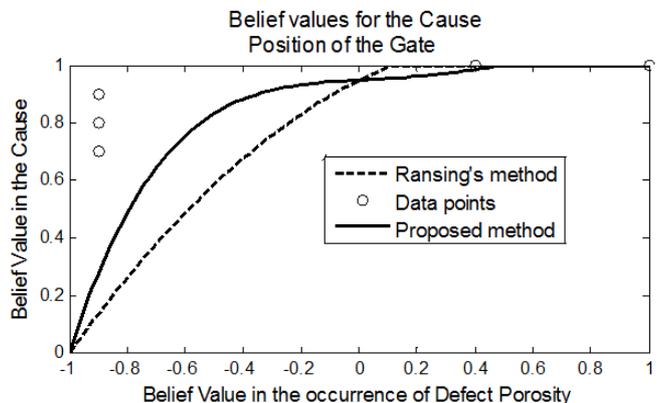


Fig. 5 The performance of Ransing's method and the proposed method for 2D belief-value variation modelled by quadratic polynomials for defect Porosity

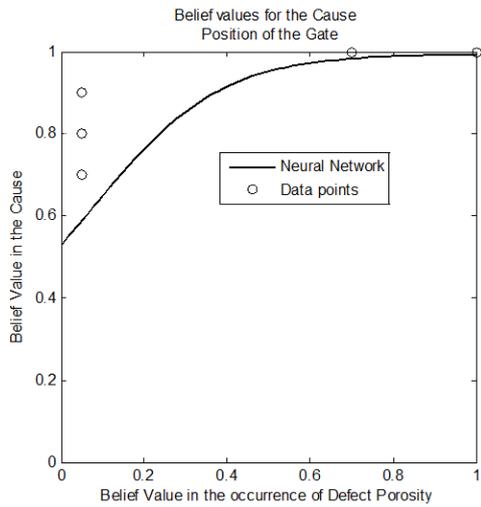


Fig. 6 The performance of the proposed neural-network method for a 2D belief-value variation for defect Porosity

Figs. 7, 8 and 9 show the variation in the belief value in the occurrence of the ‘The position of gate’ for defect B, i.e. ‘Mismakes’ using the proposed method, the method proposed by Ransing [3] and the proposed neural-network method plotted for both one-dimensional and two-dimensional cases. The results demonstrate that the proposed method has modelled the exponential rise in the data points better than both Ransing’s and the neural-network methods.

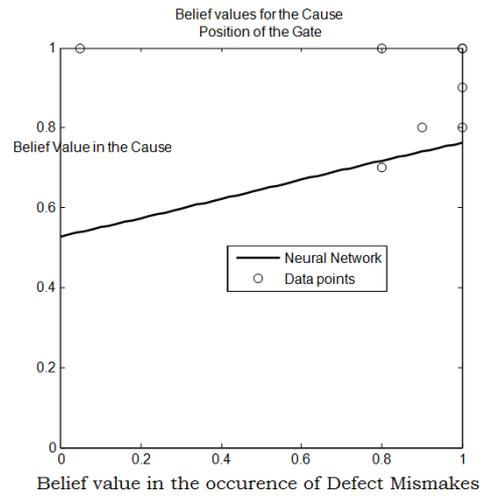


Fig. 9 The performance of the proposed neural network method for 2D belief-value variation for defect Mismakes

Figs. 10, 11 and 12 show the variation in the belief values in the occurrence of ‘The position of gate’ for belief values for defects ‘Porosity’ and ‘Mismakes’ using the proposed method, Ransing’s method and the proposed neural-network method. It can easily be observed that the proposed method has an ability to accurately model the exponential rise in the belief values rather than the other two techniques.

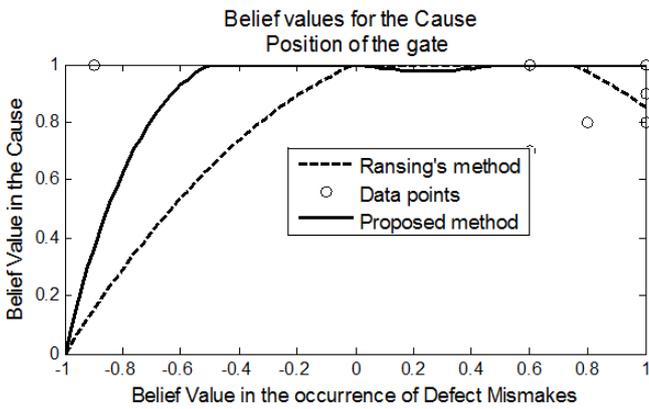


Fig. 7 The performance of Ransing’s method and the proposed method for 1D belief variation modelled by quadratic polynomials for defect Mismakes.

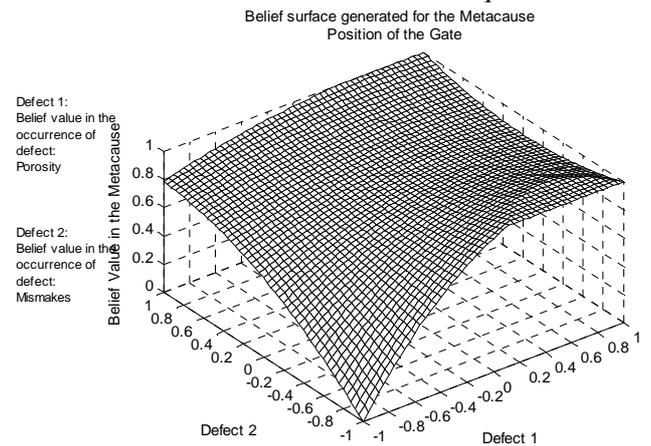


Fig. 10 2D quadratic output surface for defects Porosity and Mismakes generated by Ransing’s method

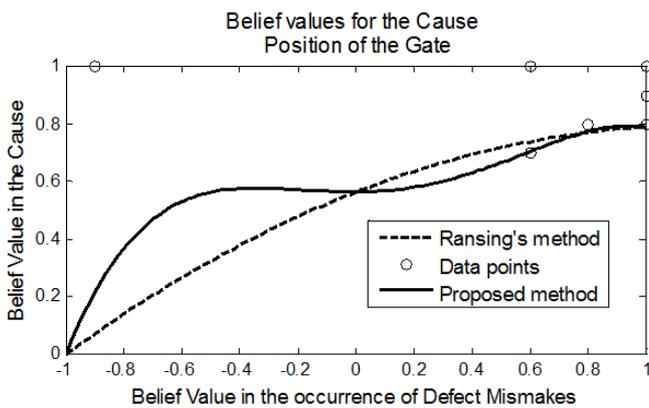


Fig. 8 The performance of Ransing’s method and the proposed method for 2D belief variation modelled by quadratic polynomials for defect Mismakes

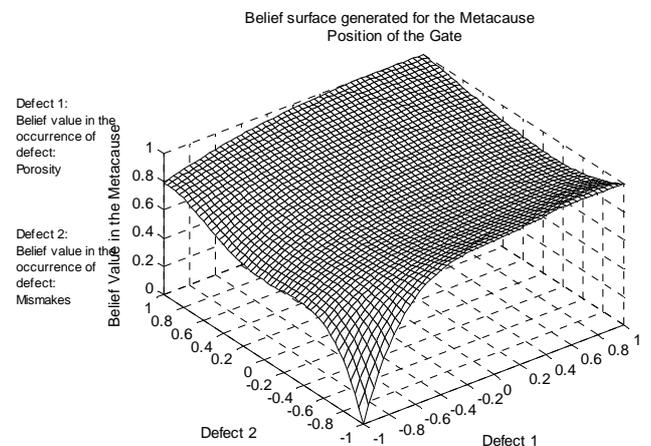


Fig. 11 2D quadratic output surface for defects Porosity and Mismakes generated by the proposed method

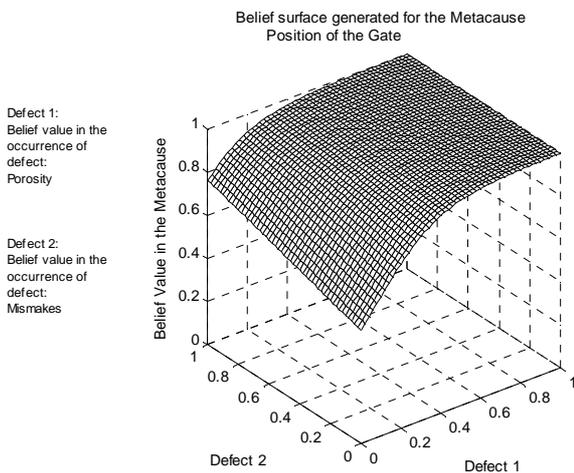


Fig. 12 2D output surface for defects Porosity and Mismakes generated by the proposed neural-network method

The major objective of a robust parameter design methodology is to make the system insensitive or ‘robust’ to a process variation. In a robust parameter-design method, the output variation can be lowered by reducing either the sensitivities to the variation in the design factor or sensitivities to noise factors. Fig. 13 shows how a factor setting may influence the variation of the output depending on the occurrence of the belief variation. When design factor setting one is chosen, more variation is transmitted from a small change design factor value to its output due to the exponential rise in the slope of the belief curve. This makes the corresponding output more sensitive to the variation of design factor setting one. Whereas for factor setting two even a larger change in values will not influence the output value. Design factor setting two thus offers a robust design setting as the process is insensitive to its variation. The proposed method has an ability to accurately model the exponential rise in the data values. This has significantly improved the applicability of the Knowledge Hyper-surface method in addressing robust design problems.

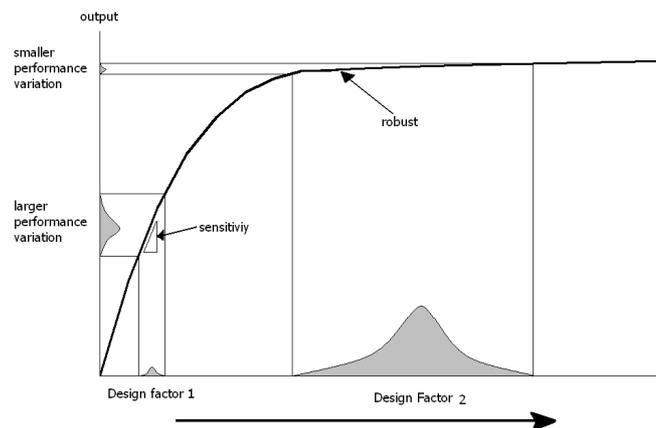


Fig. 13 Robust design principles: assessing the sensitivity of output variation to the change in design-factor settings

#### IV. CONCLUSION

This study shows that a significant effect in improving the search direction, not the learning rate. A novel method to improve the training efficiency of BP algorithms with respect to the adaptive-gain variation of activation function

has been successfully developed. The proposed method not only coupled the gain update expressions for output, as well as the hidden nodes as derived by Ransing [3], but also coupled with the adaptive-learning rate. Furthermore, the generic nature of the proposed method has been demonstrated by successfully implementing its formulation into other well-known optimisation methods to yield significant improvements in the computational speed. An enhancement to the current Knowledge Hyper-surface method has been proposed in this chapter. The method introduces midpoints in the existing shape-function formulation so that an exponential rise in the belief-value variation can be modelled without introducing the effects of ‘overfitting’. The performance of the proposed method was compared with the method proposed by Ransing [3] and the proposed neural-network method on the same casting data used by Ransing. The proposed method does not have limitations of neural-network techniques as identified by Ransing [3].

Furthermore, the ability of the proposed approach to model the exponential increase/decrease in the belief values by using high-ordered polynomials without introducing ‘over-fitting’ effects was investigated. The performance of the proposed method in modelling the exponential increase/decrease in belief values was carried out on real cases taken from real casting data used by Ransing [3]. The computed graphical result of the proposed method was compared with the current Knowledge Hyper-surface and neural-network methods. As a result of this research achievement, it will now be possible to correctly predict the sensitivity of process-parameter variations with the occurrence of defects. This is an important area of research in a robust design methodology.

#### ACKNOWLEDGMENT

The authors would like to thank Universiti Tun Hussein Onn Malaysia (UTHM) Ministry of Higher Education (MOHE) Malaysia for financially supporting this Research under Trans-disciplinary Research Grant Scheme (TRGS) vote no. T003. This research also supported by GATES IT Solution Sdn. Bhd under its publication scheme.

#### REFERENCES

- [1] Edmond Hajrizi, Felix Breitenacker, and Shabnam Tauböck, Intsched-A dynamic Optimisation Module for Modelling and Simulation in Flexible Manufacturing systems. *International. J. of SIMULATION*, 2001. 2(1): p. 50-54.
- [2] Nanosciences, Nanotechnologies, Materials And New Production Technologies - NMP, in *Seventh Framework Programme (FP7) Cooperation: THEME 4, E.s.S.F.P. (FP7)*, Editor. 2006.
- [3] Meghana R. Ransing, *Issues in Learning Cause and Effect Relationships from Examples: With particularly emphasis on casting process*, in *School of Engineering*. 2002, University of Wales Swansea: Swansea.
- [4] Hornik K., Stinchcombe M., and White H., Multilayer feedforward networks are universal approximators. *Neural Network*, 1989. 2: p. 359-399.
- [5] James W. Denton a and Ming S. Hung, A comparison of nonlinear optimization methods for supervised learning in multilayer feedforward neural networks. *European Journal of Operational Research* 93, 1996. 93: p. 358-368.
- [6] Daniel Svozil, Vladimir Kvasnicka, and Jiri Pospichal, Introduction to multi-layer feed-forward neural networks. *Chemometrics and Intelligent Laboratory Systems* 39, 1997. 39: p. 43-62.

- [7] Rumelhart D.E., Hinton G.E., and Williams R.J., Learning internal representations by error propagation. *Parallel Dist. Process*, 1986. 1.
- [8] Choon Sen Seah, Shahreen Kasim, Mohd Saberi Mohamad: Specific Tuning Parameter for Directed Random Walk Algorithm Cancer Classification. *International Journal on Advanced Science, Engineering and Information Technology*, Vol. 7 (2017) No. 1, pages: 176-182
- [9] Moaath Shatnawi, Mohammad Faizul Nasrudin, Shahnorbanun Sahran: A new initialization technique in polar coordinates for Particle Swarm Optimization and Polar PSO. *International Journal on Advanced Science, Engineering and Information Technology*, Vol. 7 (2017) No. 1, pages: 242-249
- [10] Wilson R. L. and Sharda R., Bankruptcy prediction using neural networks. *Decision Support Systems*, 1994. 11(5): p. 545-557.
- [11] Rehman, M.Z., Nawi, N.M.: The effect of adaptive momentum in improving the accuracy of gradient descent back propagation algorithm on classification problems. *Communications in Computer and Information Science* 179 Part 1, 454-459, (2011)
- [12] Nawi, N.M., Atomi, W.H., Rehman, M.Z.: The Effect of Data Pre-processing on Optimized Training of Artificial Neural Networks. *Procedia Technol.* 11, 32–39 (2013)
- [13] Nawi, N.M., Khan, A., Rehman, M.Z.: A New Levenberg Marquardt Based Back Propagation Algorithm Trained with Cuckoo Search. *Procedia Technol.* 11, 18–23 (2013)
- [14] Hamid, N.A., Nawi, N.M., Ghazali, R., Salleh, M.N.M.: Improvements of Back Propagation Algorithm Performance by Adaptively Changing Gain, Momentum and Learning Rate. *Int. J. New Comput. Archit. their Appl.* 1, 866–878 (2011)