Interscience Management Review

Volume 1 | Issue 1 Article 5

January 2008

Studies of Manufacturing Process Control: A Technological Transfer of Soft Computing

J. P. Panda
SIET, Dhenkanal, jppanda123@yahoo.co.in

R. N. Satpathy
Trident Academy of Technology , Near Infocity, Bhubaneswar-24, rabinarayan.satpathy@gmail.com

Recommended Citation
Panda, J. P. and Satpathy, R. N. (2008) "Studies of Manufacturing Process Control: A Technological Transfer of Soft Computing," *Interscience Management Review*: Vol. 1 : Iss. 1 , Article 5.
DOI: 10.47893/IMR.2008.1004
Available at: https://www.interscience.in/imr/vol1/iss1/5

Studies of Manufacturing Process Control: A Technological Transfer of Soft Computing*

J. P. Panda[1]
R. N. Satpathy[2]

Abstract:

The field of soft computing embraces several techniques that have been inspired by nature but are mathematical. These techniques are artificial neural networks, fuzzy logic and evolutionary algorithms. Often these techniques are considered part of artificial intelligence, however the name artificial intelligence is more properly given to techniques which try to capture and emulate biological intelligence, such as expert systems and thinking computers. This paper focuses on the technology transfer issues and solutions when using soft computing for off line control of manufacturing processes. This paper will discuss each of these three techniques – neural networks, fuzzy logic and evolutionary algorithms - in turn and how they might be used in manufacturing. The kind of problems these techniques are best suited for will be defined, and competing techniques will be compared and contrasted.

Keywords: Soft Computing , EA , Fuzzy Logic , AGP . Neural Network , Genetic Algorithms.

1. Introduction

New techniques from soft computing have attracted the interest of manufacturing researchers in academia, government and industry. These techniques include artificial neural networks, fuzzy logic and evolutionary computing. While these computational techniques inspired by nature have shown promise in many manufacturing applications such as robotics, machine vision, process control, process planning and scheduling, there is little in the literature on their practical use. Moving these techniques from simulated data sets, toy problems or laboratory settings to real industrial applications is a large and uncertain step.

However, there are a wealth of difficult and real problems in manufacturing that could benefit from soft computing techniques. These problems often involve modelling and optimization of complex systems, and the computational intelligence techniques cover the middle ground of the modelling continuum. That is, they range from structured and articulated knowledge to continuous empirical models. These techniques are often an improvement over unarticulated wisdom, which is found in all manufacturing environments, but do contain the certainty or elegance of analytic models derived from first principles.

2. Artificial Neural Networks

Artificial neural networks are computing mechanisms roughly modelled after the biological brain. Neural networks depend on an organized group of simple elements, called neurons. Neurons are uni-directional computing elements that receive multiple inputs, sum them, then produce a single output through a nonlinear transfer function f (see figure 1). Neurons exist in parallel (a layer) and in series. Weighted connections exist between neurons to move the output of a neuron to other neurons. A neural network can be quite small and simple, but is more likely to be large (many neurons in multiple layers) and complex. The biological human brain has about 1014 weighted connections (synapses), so even a very large artificial neural network is a poor substitute for any living brain.

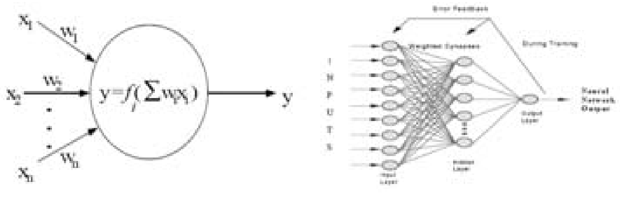


Figure 1. Typical neuron and structure of a neural network.

[1] SIET , Dhenkanal, E-Mail : jppanda123@yahoo.co.in
[2] Trident Academy of Technology ,
Near Infocity, Bhubaneswar-24,
E-Mail : rabinarayan.satpathy@gmail.com

*Citation for this article: J. P. Panda, R. N. Satpathy: Studies Of Manufacturing Process Control : A Technological Transfer Of Soft Computing, Interscience MR, Vol.1 Issue 1, pp 31 to 37.

Neural networks usually begin with their weights in a random state, which is "untrained." The network weights are iteratively evolved to a fixed (trained) state by repeated data input and error calculation. In many real manufacturing problems, theory and analytic relationships drawn from first principles are not adequate to explain the many complexities, interactions and imperfections. Similarly, many problems do not adhere to well known statistical techniques such as linear regression or clustering algorithms. For these problems, a neural network may provide the only reasonable mathematical solution technique. In fact, a neural network should be considered as a technique of last resort where other techniques have failed. This is because neural networks have some serious drawbacks. They are empirical models and depend heavily on data.

However, a neural network can be a successful technique for manufacturing applications. These include real-time control systems, diagnostic systems, machine vision systems, robotic and AGV control systems, and others. In this paper the author has used neural network models for manufacturing process control and optimization. These are usually static models which use the raw material characteristics, the product design specifications, the ambient conditions, operator information and actions, the machine settings and so on to predict outcomes of the process. Model dynamics can be added by using feedback during the process. The outcomes may be quality indicators, such as blemishes or incomplete joinings, or they may adherence to specifications and tolerances, or they may be some surrogate correlated with an outcome measure of interest. If such models mimic the manufacturing process with fidelity, they can be used for a variety of purposes. They can be used to interactively select manufacturing conditions and machine settings which produce the best outcome. They can be used to select design features which improve manufacturability. They can be used to estimate proper settings for new products without trial-and-error testing. They can be used to identify the most important variables of a process. They can be used with an optimization algorithm to directly identify optimal settings or conditions.

Here author has used neural networks to model the following processes when working with industrial partners: injection mouldings, plastic pipe extrusion, ceramic casting, metal furniture assembly, wave soldering and abrasive flow machining. These are very diverse processes but they share important common elements. First, theoretic or analytic models were not adequate for the processes. Second, they exhibited non-linear behaviour with variable interactions. Third, observational data was available. Fourth, the companies desired to improve control of the processes by systematic selection of the controllable variable settings. However, each process required a somewhat different approach depending on the company's objectives, the available data, the kind and number of process variables, and the repeatability of the process.

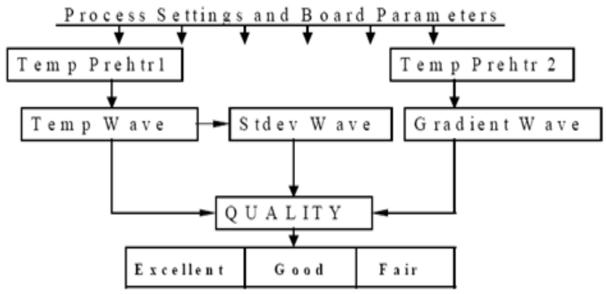


Figure 2. Hierarchical system of six neural networks to model a wave solder process.

Figure 2 shows a hierarchy of six neural networks that were built to model a wave solder process for Lockheed Martin. Two initial neural networks use the circuit board design parameters and the process settings (line speed and pre-heater temperature) to predict the board surface temperature at each bank of pre-heaters. These predictions are added to the design and process settings and are used to predict the mean surface temperature of the board at the solder wave and the rate of change of temperature at the solder wave. The variance of temperatures over different places on the board surface is also predicted using the prediction of mean temperature at the wave. All of the thermal predictions at the wave are combined with the initial variable set (board design and process settings) to predict the solder quality of the board using a categorical metric of excellent, good and fair. The ultimate goal was to predict solder quality so that it could be optimized, and the thermal condition of the board at the solder wave was highly correlated with solder quality. However, the thermal condition could not be regularly observed during production and could only be observed during special experimentation used to gather data for the project.

Therefore, during production, the thermal condition had to be estimated by the neural networks, motivating the need for the hierarchical model. Another use of a neural network model is shown in figure 3. This neural network modelled a ceramic casting process for a large sanitary ware manufacturer. After the neural network was finalized, it could be used to analyze which process variables were most important and their general effects on the process. Holding all other process variables constant (at their minimum, mean and maximum values, respectively), plant temperature was varied from its minimum to its maximum. It is easy to see from figure 3 that there is a non-linear effect that is more pronounced when the other variables are near their minimum values than when they are near their maximum values, indicating significant interactions.

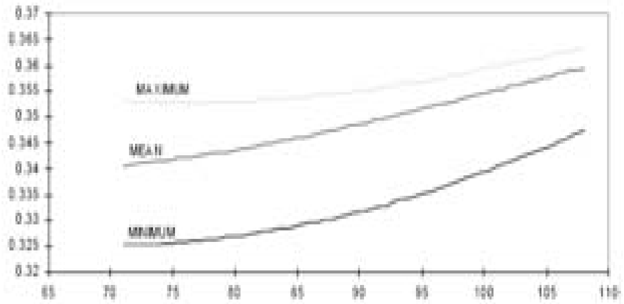


Figure 3. Identifying the effects of plant temperature on a ceramic casting process.

3. The fuzzy concept

3.1 Fuzzy Logic

In a seminar paper written in 1965 Lotfi A. Zadeh[5:338-353] described the properties of fuzzy sets, a class of objects with a continuum of grades of membership in the interval (0,1). This idea stands in stark contrast to conventional set theory in which objects have only membership (characteristic function) values taken from the doubleton set {0,1}. Each object x in a fuzzy set X is assigned a grade of membership by a membership function usually denoted by .00 (x) whose values range between zero and one. Many people tend to confuse the idea of a membership function .00 (x) with that of a probability density function f(x), however, this is correct since the integral of f(x) must sum to 1. There is no such restriction of .00 (x). The foundation of fuzzy logic is fuzzy set theory, first proposed by Bellman and Zadeh [1970], Wang and Wang [1985a, b] Soh and Yang [1996], Yang and Soh [2000] and Rao [1987], applied fuzzy optimization techniques.

3.2. Introduction to Fuzzy Sets

Zadeh makes a case that humans reason not in terms of discrete symbols and numbers, but in terms of fuzzy sets . These fuzzy terms define general categories, but not rigid, fixed collections. The transition from one category-concept, idea, or problem state-to the next is gradual with some states having greater or less membership in the one set and then another. From this idea of elastic sets, Zadeh proposed the concept of a fuzzy set. Fuzzy sets are functions that map a value that might be a member of the set to a number between zero and one indicating its actual degree of membership. A degree of zero means that the value is not in the set, and a degree of one means that the value is completely representative of the set. This produces a curve across the members of the set. There are many books that have been written on the subject of fuzzy sets since Zadeh introduced the fuzzy set concept in [1965, 1-19].

3.3. Membership Functions

Let X be a set of objects, called the universe, whose elements are denoted x. Membership in a subset A of X is the membership function, μ_A from X to the real interval [0,1]. The universe is all the possible elements of concern in the particular context. A is called a fuzzy set and is a subset of X that has no sharp boundary. is the grade of membership x in A. The closer the value of is to 1, the more x belongs to A. The total allowable universe of values is called the domain of the fuzzy set. The domain is a set of real numbers, increasing monotonically from left to right where the values can be both positive and negative. A is completely characterized by the set of pair

$$A = \{(x, \mu_A(x)), x \in X\} \qquad (1)$$

3.4 Operation on Membership Functions

($\mu_A, \mu_B, \mu_C, \ldots$ are membership values)

- Fuzzy AND : $\mu_{AND} = MIN(\mu_A, \mu_B, \mu_C, \ldots)$
- Fuzzy OR : $\mu_{OR} = MAX(\mu_A, \mu_B, \mu_C, \ldots)$
- Fuzzy Algebraic Product : $\mu_{Product} = \prod_{i=1}^{n} \mu_i$
- Fuzzy Algebraic Sum: $\mu_{Sum} = 1 - (\prod_{i=1}^{n} (1 - \mu_i))$
- Fuzzy Gamma: $\mu_\gamma = (Fuzzy\ Algabric\ Sum)^\gamma * (Fuzzy\ Algabric\ Product)^{1-\gamma}$

3.5 Fuzzy Systems

Fuzzy logic is an extension of Boolean logic, where an item can have partial membership in a set. Membership degree ranges from 0 (definitely not in the set) to 1 (definitely in the set). A simple example is shown in figure 4, where plant temperature is shown as a Boolean variable (not hot or hot) versus a fuzzy variable (from not hot to hot). It is readily seen that a fuzzy variable has more information than a Boolean variable and more properly expresses transitions. Fuzzy logic should not be confused with probability; fuzzy logic connotes imprecision rather than uncertainty. Fuzzy logic is useful as a rigorous and numeric way to handle qualitative variables. It has been most notably used in control systems, but there are many other possibilities in manufacturing, especially in the development of expert systems and the analysis of imprecise data.

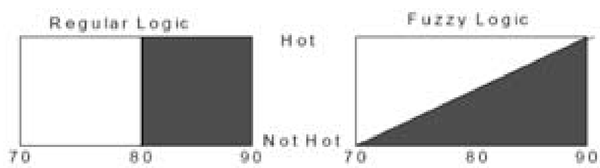
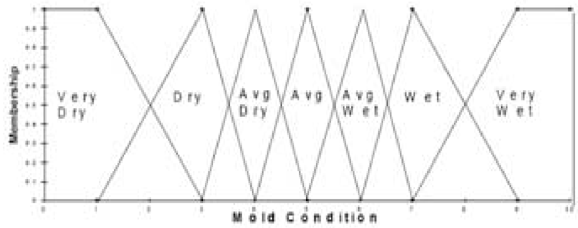


Figure 4. Example of difference between regular and fuzzy logic for plant temperature.

Development of a fuzzy logic system can be time consuming and tedious. All relevant variables must be identified, then descriptors determined. The range and shape of the membership functions for each descriptor must be specified. For example, in figure 5, mold condition is an ordinal measure ranging from 0 to 10 in ceramic casting, where 0 indicates an extremely dry mold and 10 indicates an extremely wet mold. For this variable, seven descriptors were chosen, ranging from very dry to very wet. The membership functions are the traditional triangular or trapezoidal shapes. Note the regions of descriptor overlap - it is these overlaps that allow fuzzy logic to make smooth transitions. After the variables, their descriptors and the membership functions are defined, a set of rules must be developed to invoke fuzzy reasoning. These are generally of the IF/THEN or modus ponens type, where the IF section contains the premise and the THEN section contains the consequents (conclusions or actions). The bottom of figure 5 shows a fuzzy associative memory (FAM) which is a compact table of rules. For example, if temperature (Temp) is low and the humidity is medium and the mold age is old, the mold condition is very wet. Rules are processed using sequential reasoning (forward or backward chaining) and final results are usually defuzzified to a non-fuzzy (crisp) answer.



Temp	Humidity / Age								
	Low			Medium			High		
	New	Mid	Old	New	Mid	Old	New	Mid	Old
Low	Dry	Dry	AvgWet	Avg	AvgWet	VeryWet	AvgWet	Wet	VeryWet
Medium	VeryDry	Dry	AvgWet	Dry	Avg	Wet	AvgDry	AvgWet	VeryWet
High	VeryDry	VeryDry	Avg	VeryDry	Dry	AvgWet	AvgDry	AvgWet	Wet

Figure 5. Membership functions and fuzzy associate memory for mold condition.

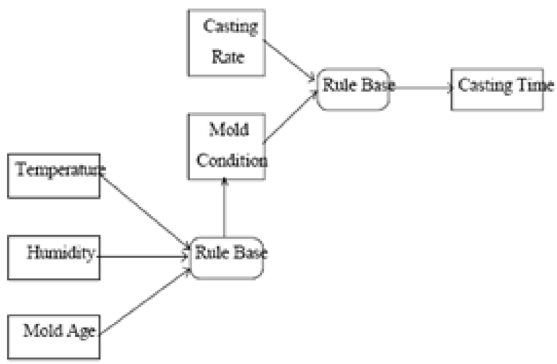


Figure 6. Schematic of hierarchy of two fuzzy rule bases for ceramic casting.

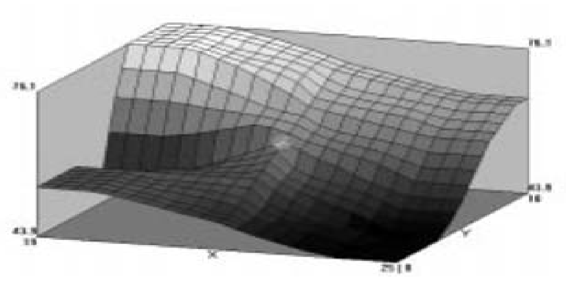


Figure 7. Prediction surface of cast time (z) vs. cast rate (x) and mold condition (y).

Figure 6 shows the schematic of a hierarchical fuzzy rule base developed in support of the modeling of the ceramic casting process mentioned earlier . These rule bases were necessary because not all the important variables were quantitative or had exact measurements, so they could not be modeled with a neural network. The first rule base takes the plant ambient temperature and humidity combined with the age of the mold in weeks and predicts the mold condition on an ordinal scale of 0 to 10. This prediction is paired with the continuous variable of casting rate (predicted by a neural network) for the second rule base. This rule base predicts the casting time in minutes (it is defuzzified to a crisp variable using the centroid method). The two rule bases were developed by eliciting the expertise of the plant engineers and foremen, and the membership functions were developed by analyzing past history of the values of the variables (temperature, humidity, mold age, etc.). The rules, membership functions and descriptors were refined until the hierarchical rule bases produced an suitable prediction surface (figure 7). Note that the surface is highly non-linear but smooth, properties that a properly crafted fuzzy rule base will possess. The knowledge elicitation and refining steps of the development process are time consuming, and can be frustrating and tedious for the experts involved. However, the development is a one-time activity while the fuzzy system may be used daily in decision making for many years to come. Like neural networks, development of a fuzzy rule base will require someone with expertise in the area and specialized software,

4. Evolutionary Computing

The field of evolutionary computing (EC) includes the following subjects: genetic algorithms, evolutionary strategies, genetic programming and classifier systems. All are meta-heuristics inspired by the process of biological evolution where an initially random population evolves iteratively to a superior, or optimized, state. Solutions are selected for recombination based on their objective value function, where a better value yields a higher fitness. The recombination, called crossover, usually involves two selected solutions (parents) that are combined to form one or more new solutions (children). The children solutions are randomly perturbed slightly (mutation) to move the search to new regions. Evolutionary computing is generally used for optimization, whether it be of a continuous function, a combinatorial problem, or finding optimal rules to explain data. The advantages of evolutionary computing over more traditional approaches such as mathematical programming are that a population (small group) of superior solutions are obtained, no assumptions about form or derivatives are made, the iterative nature is usually diminishing in improvements so the computational time needed is flexible, EC is very flexible and can accommodate almost any problem, and EC is easy to code and to understand. Moreover, EC is a global technique, that is, it is resistant to becoming trapped in local optima. Disadvantages of evolutionary computing are that it is stochastic and may return different solutions depending on random number seed, it cannot guarantee convergence or optimality except under very restrictive conditions, and it may not be computationally efficient compared to other problem-specific methods. The advantages of not specifying a functional form that is differentiable, continuous and so on is a tremendous advantage in real world problems. It is also advantageous to use a global technique since it is usually unknown whether a surface is convex, a condition required for gradient methods to converge to the global optimum. The ease of coding and the flexible computing time are added inducements for the use of EC on real manufacturing problems in optimization.

To use EC, the solution space must be encoded as a series of bits, floating point numbers, or as a permutation. The traditional genetic algorithms uses a bit encoding, however this is not required, or even desirable in many instances. Solutions must be able to be compared on a numeric basis using an objective function, which translates to fitness in EC. The better the objective function, the more likely the solution will survive in later iterations and also produce children solutions. Poorly evaluated solutions tend to die off immediately. To summarize, the important steps to using EC are an encoding, crossover and mutation algorithms and a method of calculating the objective function value of a solution. There are problem-specific parameters that must be set, such as population size, probability of mutation and termination criteria, but EC is robust to a wide variety of these settings.

Consider the following example for the problem of optimizing the design of a local area network (LAN) within a manufacturing plant. The number of computer sites are fixed and their locations are known. The concern is to specify the communications cables between the links so that the LAN meets a certain network reliability objective but the cost of the network is minimized. The problem can be encoded for EC as follows in figure 9. The 0/1 encoding shows the cables present for parent 1. Two parents are combined using single point crossover to form a child, then the child is slightly perturbed (bits changed with a small probability) to become the final, mutated child.

This LAN design problem is well handled by EC, with work by the author showing vast improvement over branch-and-bound methods or greedy search methods . Other optimization problems, such as production scheduling, product design , process planning and plant layout are suited to EC.

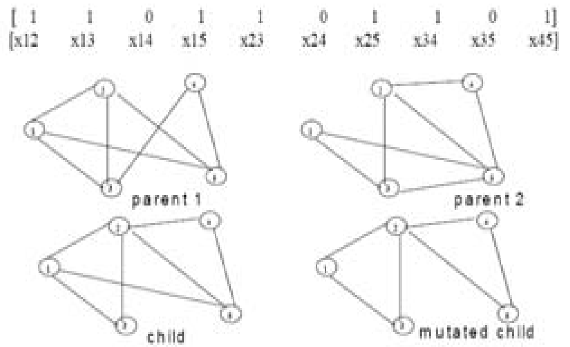


Figure 8. Encoding, crossover and mutation for LAN design optimization.

5. Conclusion

Many manufacturing environments can benefit from the judicious use of these techniques in prediction, classification, decision making and optimization tasks. While the development effort will require people with knowledge and experience in these methods and may require specialized software and hardware, the system itself should be able to be operated by almost anyone. Caution should be exercised however about the widespread use of these techniques. They should be regarded as a final alternative after more straightforward and simpler methods have been exhausted. If a linear regression model is adequate, then a neural network should not be used. If there is an analytic description of the process that works well, then an empirical or knowledge based approach should not be used. When considering manufacturing uses of computational intelligence, the timeline is even more recent. Neural networks began in the field of cognitive neuroscience, and has been dominated by that field along with computer science and electrical engineering. Fuzzy logic began in electrical engineering, and it is control applications for which it is still most known. Evolutionary algorithms were founded by computer scientists, mathematicians and electrical engineers, and these fields still produce most of the research in the area.However, for many real manufacturing problems, only a technique that is flexible and is based on data and knowledge is appropriate. In those cases, using soft computing should be regarded as a viable alternative that can work even in the most traditional and low technology circumstances.

References

1. TilShell User's Manual and software, Togai Infralogic, Inc., Irvine, CA, various years.

2. White, Halbert, "Learning in artificial neural networks: a statistical perspective," Neural Computation, vol. 1, 1989, 425-464.

3. Wilhelm, Mark, Alice E. Smith and Bopaya Bidanda, "Integrating an expert system and a neural network for process planning," Engineering Design and Automation, vol. 1, no. 4, Winter 1995, 259- 269.

4. Wilhelm, Mark, Alice E. Smith and Bopaya Bidanda, "Process Planning Using an Integrated Neural Network and Expert System Approach," in Hybrid Intelligent System Applications (J. Liebowitz, editor), Cognizant Communications/ ISIS, Elmsford, NY, 1996, pp. 3-23.

5. Zadeh, Lotfi A., "Fuzzy sets and systems," Information and Control, vol. 8, no. 3, 1965, 338-353.

6. Petri, Kimberly and Alice E. Smith, "A hierarchical fuzzy model for predicting casting time in a slipcasting process," Proceedings of the Fifth International Industrial Engineering Research Conference, Minneapolis, MN, May 1996, 217-222.

7. Smith, Alice E., "Predicting product quality with backpropagation: A thermoplastic injection moulding case study," International Journal of Advanced Manufacturing Technology, vol. 8, no. 4, 1993, 252-257.

8. Smith, Alice E. and Hulya Yazici, "An intelligent composite system for plastic extrusion process control," Journal of Engineering Applications of Artificial Intelligence, vol. 5, no. 6, 1992, 519-526.

9. Tate, David M. and Alice E. Smith, "Unequal area facility layout using genetic search," IIE

Transactions, vol. 27, 1995, 465-472.

10. NeuralWorks Reference Guide and Explorer software, NeuralWare, Pittsburgh, PA, various years.

11. Norman, Bryan A. and Alice E. Smith, "Random keys genetic algorithm with adaptive penalty function for optimization of constrained facility layout problems," Proceedings of the IEEE Conference on Evolutionary Computing, Indianapolis IN, April 1997, 407-411.