A contemporary study into the application of neural network techniques employed to automate CAD/CAM integration for die manufacture

Lian Ding, Jason Matthews *

Innovative Manufacturing Research Centre, Department of Mechanical Engineering, University of Bath, BA2 7AY, UK.

*Corresponding author. Fax: +44 (0) 1225 385937
E-mail address: j.matthews2@bath.ac.uk

ABSTRACT

In recent years, collaborative research between academia and industry has intensified in finding a successful approach to take the information from a computer generated drawings of products such as casting dies, and produce optimal manufacturing process plans. Core to this process is feature recognition. Artificial neural networks have a proven track record in pattern recognition and there ability to learn seems to offer an approach to aid both feature recognition and process planning tasks. This paper presents an up-to-date critical study of the implementation of artificial neural networks (ANN) applied to feature recognition and computer aided process planning. In providing this comprehensive survey, the authors consider the factors which define the function of a neural network specifically: the net topology, the input node characteristic, the learning rules and the output node characteristics. In additions the authors have considered ANN hybrid approaches to computer aided process planning, where the specific capabilities of ANN's have been used to enhance the employed approaches.

1. BACKGROUND

Over the last twenty five years, computers have been employed to greater extents, to aid the design and manufacturing processes. All parts are designed and detailed using computer aided design (CAD) packages such as Pro/Engineer and the computer numerically controlled machines (CNC) are common place even in jobbing shops. Computer aided process planning (CAPP) is the core element for genuine integration of CAD and CNC. Die casting is a method to produce finished metal components by forcing molten metal into a suitable metal die under pressure. The dies investigated in this research are for cold chamber die casting and are normally manufactured from heat treated chromium alloy steels such as H13. Figure 1 shows a typical casting die, with its variety of machined features. The usual configuration of a die is a cavity which is filled by the molten metal to form the external profile of the component. In addition a die will include features such as feed tracks, overflow feeders and holes for ejection pins. Quality and cost are the key factors for die casting companies, which are dependant on rapid interpretation of design and optimal CAPP.

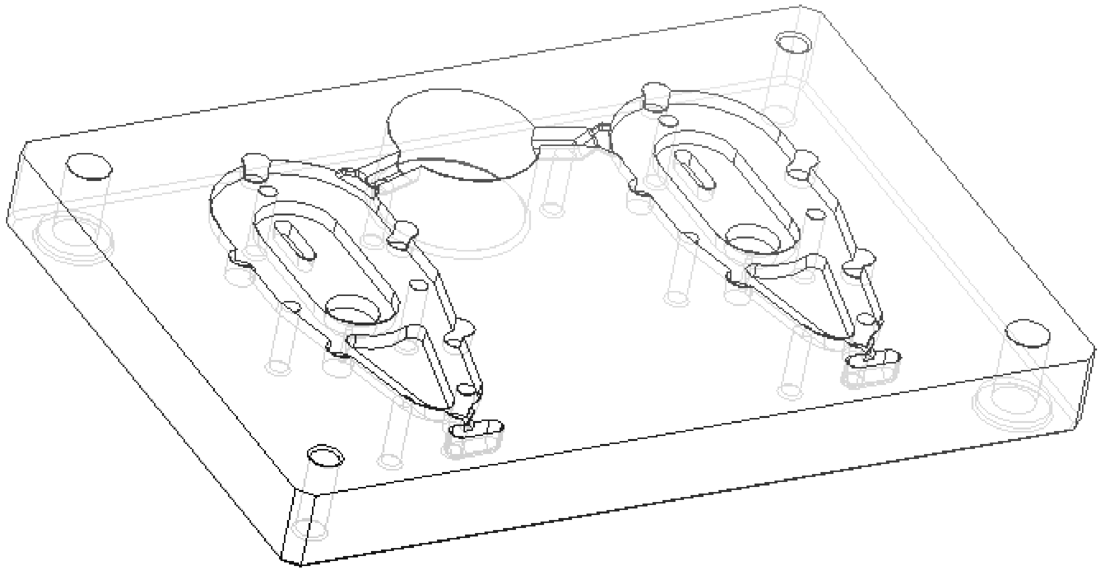

Figure 1 Twin cavity casting die for mounting plate

Although computer aided manufacturing packages such as DellCam Powermill™ are employed to find the optimal sequence to machine each feature, the jobs of interpreting the die design drawings and development of the manufacturing sequence strategy are left to the CNC programmer on the shop floor. The programmer will be under strict time constraints, as well as the monitoring of other machines on the shop floor. It is likely this will not be the optimal sequence for the part, and it is more likely to be the easiest for him/her to program. This is squandering valuable production time on the machine. To improve the die design and manufacturing, two issues should be considered: design input extracted from a CAD model (e.g. a feature-based model) by feature recognition, and acquisition and representation of process knowledge.

1.1 Feature recognition

Zhang and Atling (1994) have defined features as “the generic shapes with which design and manufacturing engineers associate attributes and knowledge as useful in reasoning about a given product”. As one of major feature technology, feature recognition has been identified as the key tool to integrate design and manufacturing processes. A variety of recognition methods have been forwarded to date. Such as: The logic approach (Sadaiah et al 2002), where a geometric feature extraction is carried out by browsing *.txt files in SolidWorks application protocol interface. Recognition is performed by extracting forms with the patterns in a database. The limitation of this approach come from ambiguous representations and predefined rules needed for multiple features, making it inflexible. Also, the volume decomposition method (Kim, 1992), here the features are decomposed into cells (unit volumes). All cells that have common faces or co-planar faces are merged to get maximum cells and that can be removed in one singular machining operation (Sakurai and Chin, 2002). A limitation of this approach comes from the limited features it is applicable for. Also, the large number of possible cell combination per feature, produces enormous time complexity. These approaches are also limited by the amount of manual interaction/ processing required. These conventional methods have their own advantages and disadvantages, however, there are some common problems hindering their practical applications, such as inability to learn, limited range of recognition, low speed, etc. A die is constructed from various machined features these include: the main product cavity, overflow pockets, air release slots, core block slots, runner slots and a variety of holes for ejection and core pins. The features, such as product cavity, may have high-index surfaces. In addition most features are highly radiused and contain draft angles to aid component ejection. Thus feature interactions and high algorithmic complexity, are still the problems that need to be solved for feature recognition in die manufacture.

1.2 Computer aided process planning

The nature of process planning for manufacturing systems is both dynamic and complex. It presents a complete description of manufacturing capabilities and requirements for processing of product. The core components of CAPP are the selection of suitable set-up plans for process resources and sequencing processing operations so that corresponding objective can be obtained: such as the least processing cost of the part, the minimum tardiness or combined objective (Ding et al., 2005). Normally using a feature-based component model (usually pre-processed by a feature recognition method) as the input, CAPP evaluates the design for its manufacturability and generates a viable process plan through geometric and technological reasoning. During these tasks, machine and cutting tools, jigs and fixtures, etc. also need to be considered. Research by Mamalis et al., (1994) has shown that effective CAPP can result in reduction of up 50% in manufacturing time. This relates to a 30% reduction in production costs. In the contemporary manufacturing environment this is vital to competitive enterprise and ability to respond rapidly to market changes (Zhou et al., 2007). The push and adaption of integration of CAD and CAPP based on feature recognition has recently been shown for injection moulding sets (Alam et al., 2003) and packing forming sets (Hicks et al., 2007). A large amount of work has been carried out in CAPP area, using both retrieval and generative methods, with varying degrees of success, but some questions still need to be answered, such as automated sequencing strategy and evaluation technique for manufacturability. Die manufacturing process planning has its own characteristics; for instance the finishing of the die cavity is normally performed via pre-made electrodes on electro discharge machines (EDM). The area where the advancements can be made lies in the blocking of die and profiling, roughing of the cavity, machining of holes for core and ejector pins and the relevant slots and pockets relating to overflows, core block slots and feeds

Prior work has been performed in this area, by both industry, such as CAMWORKS™, Esprit™ and, academia with Featurefinder (Little et al., 1998; 1999) which is based on an algorithm using adjacency graphs. These systems have achieved varying degrees of success with CAPP and feature recognition, but manufacturer of the commercial packages offered to date can only offer 2½ D features at present. Also, these approaches do not retain the ability to learn which artificial neural networks can offer.

1.3 Paper structure

ANN's have a proven track record in pattern recognition (Kulkarni et al., 1998), and there ability to learn offers an approach to aid both feature recognition and process planning task. Therefore, the authors have identified ANN in offering the best potential for this research. With this mind a review of the ANN's has been needed so the readers of this article can understand the research area in which the authors have surveyed. The motivation in producing this review is to investigate solutions to find the optimal process plans of die-casting dies and their inserts. The contribution of this paper is two-fold. Firstly, it will critically present the current state-of-the-art in using ANN and hybrid solutions to feature recognition and CAPP. And secondly, it will discover the limitations of currently applied approaches and identify the directions for the continuing research to produce automated, optimal process plans for the die-casting dies. The remainder of the paper is structured as follow: section 2 introduces the neural network, its architecture, input requirements, output form, and the varieties of ANN that have previously been applied to this field. Section 3 introduces the feature input preparation techniques. Section 4 presents the application of ANN to CAPP. Section 5 offer discussions of the current state-of-the-art and its potential when applied to die manufacture, and section 6 presents the concluding remarks for the paper.

2. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network is a system loosely modelled on the human brain. The origins of such networks can be traced back to the 1940's when McCullock and Pitts (1943), who presented a mathematical model of the biological neuron. A model of an artificial neuron is shown in Figure 2.

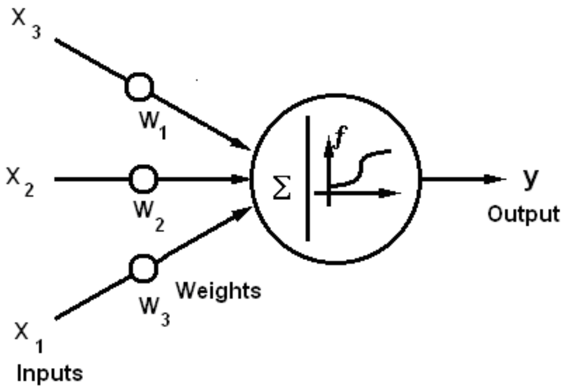

Figure2 Model of artificial neuron

The function of a neural network is to produce outputs when presented with inputs. Within a neural network each neuron is linked to certain of its neighbours with varying coefficients of connectivity these represent the strengths of these connections. Learning is accomplished by adjusting these strengths to cause the overall network to output appropriate results. Figure 3 shows the configuration of a typical three layered neural network, associated with feature recognition. ANN’s can be regarded as an adaptive and learning technique, and has several advantages over other methods employed in feature recognition and CAPP. They can tolerate slight errors from the input during learning or problem solving. In general they are faster, because the process is limited to simple mathematical computations and do not use either a search or logical parse information. ANN’s, also retain the ability to derive rules or knowledge through training with examples and can allow exceptions and irregularities in the knowledge/ rule base. For the evaluation process of ANN for CAPP and feature recognition, we must consider the four key factors of functionality, the network topology, input node characteristics, learning (training) rules, and output node characteristics. The ANN’s performance is closely related with their architecture design; in most papers reviewed the authors have strived to find an optimal architecture in respect of function and performance. The assumption in this paper is that all the research reviewed has found the best network for the input representation strategy and application it has been applied too and so, is reviewed against these criteria.

2.1 Types of artificial neural networks

In general, there are three main ANN architectures have been employed by researchers in the process of feature recognition and computer aided process planning, these are: feed-forward, self organizing maps and recurrent.

2.1.1 Feed-forward networks.

In this type of network, the neurons are strictly feed-forward i.e. towards output, to activate the neurons in the next higher layer. In such networks there are no connecting arcs feeding back 'upstream' neuron. For a typical single-layer feed forward network, its input neurons are fully connected to output neurons, but not vice versa. The input neurons are not connected to each other and the output neurons are not connected to other output neurons. In order to constrain the value of each neuron on the current layer, various transfer functions are applied. For instances, a sigmoid function can be used to limit each neuron ranging from 0 to 1:

$$f(x_l) = \frac{1}{1 + e^{-x_l}} \tag{1}$$

The Bipolar sigmoid function is also commonly used function to make the output fall in a continuous range from –1 to 1. It is closely related to the hyperbolic tangent function, which can be described (approximated) as the following equation 2. Santochi and Dini (1996) found this is the best function for cutting parameters selection, in their three-layered FF network.

$$f(x) = \tanh \sigma x = \frac{1 - e^{-2\sigma x}}{1 + e^{-2\sigma x}} \tag{2}$$

The linear or Ramp function (cf. equation 3) has also been employed

$$f(x) = x \tag{3}$$

While optimizing their network, Deb et al (2006) found a linear transfer function on the input layer and a sigmoid function on the hidden layer best suited the recognition of machined features on radial parts. The general consensus from most the reviewed work states that, the sigmoid functions are generally desirable for the feature recognition problem.

The three-layer FF neural network. This topology (cf. figure 3a) has an input, a hidden and an output layer. Neurons on the output layers are defined from the neurons on the previous layer, the weights and a processing algorithm. For example, in Chuang's system (Chuang *et al.,* 1999), the *l*th neuron on the current layer, N_l can be calculated as:

$$N_l = \sum_{k=1}^{n} u_k w_{kl}, \tag{4}$$

Where u_k is the *k*th neuron on the previous layer, and w_{kl} is the weight representing the strength of the relationship between the *k*th neuron on the previous layer and the *l*th neuron on the current layer.

Further, for the neuron on the output layer, the value is converted into 0 or 1 by an appropriate *thresholding* scheme.

There have been other instances of using three-layer feed-forward neural networks, such as on 3D feature recognition. (Jun et al., 2001), used sets of scanned points to recognise features associated with prismatic parts. Wong and Lam (2000) used the topology to recognize orthogonal and non-orthogonal machined features. Li et al (2000) employed such a network to investigate interacting feature. Ding and Yue (2004) recognized machined features on prismatic components. The core reason quoted by most authors for using this network architecture was its proven track record in pattern recognition and in previous feature recognition research.

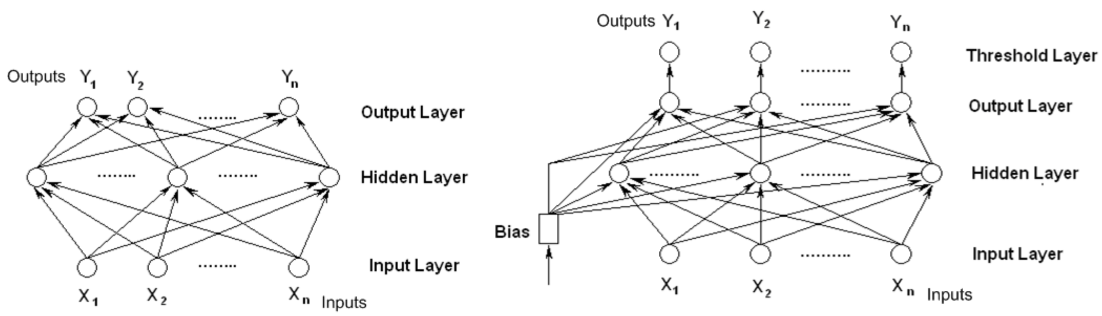

Figure 3 Three and four layered FF networks

The four-layer FF neural network. The three layered network has one hidden layer, this may be sufficient to estimate any continuous function, it is unlikely to be optimal in terms of learning time or implementation effort. If one hidden layer is employed then it may require an infinite number of neurons to approximate a given the function. The use of two hidden layers can avoid this assumption (Chester, 1990). For this reason Ozturk and Ozturk (2001) used a standard four layered ANN for primitive circular and rectangular feature recognition. Nezis and Vosniakos (1997) also used four layers, but with a different topology: an input, a hidden, an output, and a threshold layer which is added to the network as the training is completed (cf. figure 3b). The threshold layer performs the function of activating the neurons of the output layer by a threshold, e.g. 0.5. All elements of the hidden and output layers are connected with a bias element that can be considered as an activation threshold. Although an optimized four-layer feed-forward networks is more versatile than an optimized three-layer feed-forward networks, they train more slowly due to the attenuation of errors through the non-linearities (Principe et al., 2000).

The five-layer, perceptrons quasi-neural network. Prabhakar and Henderson (1992) developed a five-layer, perceptrons quasi-neural network system called PRENET. The system has five layers which respectively, consist of N nodes, N groups of M nodes, N nodes with a threshold non-linearity, M nodes corresponding to the M conditions for a feature, and one node, where N is the number of faces in the test part and M is the number of conditions required for the feature. Their four stage approach involved converting the input vectors of a row into single integers called codes. Searching for the integers corresponding to that feature definition. From this definition, finding the faces that satisfy the conditions specified for the feature. Then, producing the recognition result to the node in the 5th layer by an AND operation.

2.1.2 Self organizing maps. The self-organizing map (SOM) is closely related to feed forward neural networks. Its topology is a single layer network where the inputs are connected to all output neurons. These output neurons are arranged in low dimensional grid. A weight vector with the same dimensionality as the input vectors is associated with every neuron. The main attributes of SOM and their variants are their ability for competitive learning and to cluster data. Two variants of SOM have been identified in feature recognition, Kohonen (Kohonen, 1987) and Adaptive Resonance Theorem (ART) (Carpenter and Grossberg, 1987) Meeran and Zullfifi (2002) used a Kohonen self-organizing feature map (SOFM), the structure of which is shown in Figure 4a. This is capable of recognition of non-orthogonal interacting features. The SOFM was used to cluster the vertices of interacting features, by combining Boolean operations to breakdown the interacting feature to extract volume data for the feature. Lankalapalli et al., (1997) used an ART2 network to recognise nine machined features. The structure of an ART network can be seen in Figure 4b, which comprises bi-directional interconnections between a set of input nodes neural network. The use of this network gives the potential to enable decomposition and recognition of a variety of interacting features but the method does not suffer from combinatorial explosion.

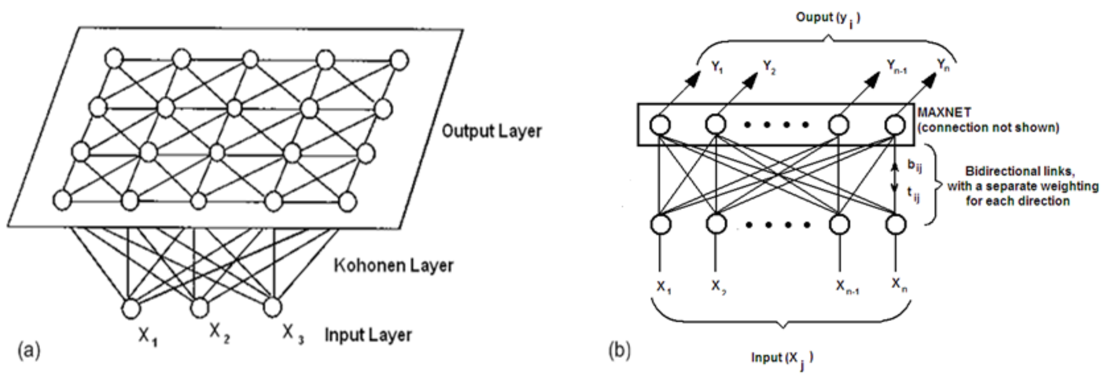

Figure 4 Self organizing maps

2.1.3 Recurrent networks.

Recurrent networks differ from the feed forward neural network in that it has at least one feedback. Recurrent networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. The term of recurrent is also referred to feedback architectures, although recurrent is often used to denote feedback connections in single-layer organisations (Picton, 1994). Typically recurrent networks include Hopfield network, Maxnet, and recurrent back propagation net.

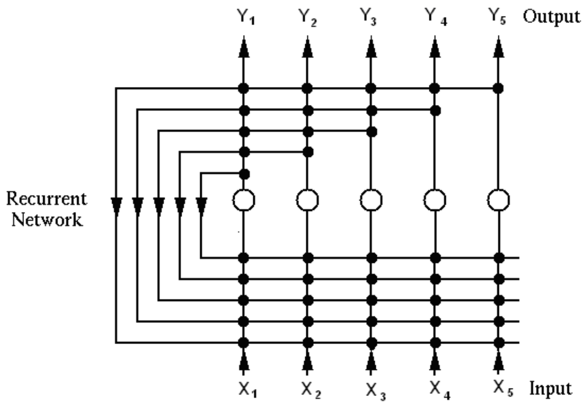

Figure 5 Hopfield network

Hopfield network. The Hopfield network is a single layer recurrent network that uses threshold process elements and an interconnect symmetric matrix as shown in Figure 5. The Hopfield network has only one layer and the nodes are used for both input and output (Hopfield and Tank, 1985). A minimum point or attractor has been demonstrated to be existence in this network, which corresponds to one of the stored patterns. The dynamics of the Hopfield network can be described by the state of an energy function which eventually gets to a minimum point. Wang and Liu (1993) exploited the ability of a Hopfield net to recognize basic features on a CAD drawing. At the time of their publication this network offered superior processing for their approach against contemporary algorithms

Brain-State-in-a-Box (BSB). As a discrete-time recurrent network with a continuous state, the output values of a BSB, consists of interconnected neurons, which are dependant on: the learnt patterns, the initial values of given patterns and the recall coefficients. The motion of a BSB network can be described by the following equation (Principe et al. 2000):

$$y_i(n+1) = f(x_i(n) + \alpha \sum_{j=1}^{N} w_{ij} x_j(n)) \tag{5}$$

$$f(u) = \begin{cases} 1 & if \quad u \geq 1 \\ u & if \quad -1 \leq u \leq 1 \\ -1 & if \quad u \leq -1 \end{cases} \tag{6}$$

A BSB can be used as a subnet for decision feedback applications, because it amplifies the present input until all neurons saturate, and eventually converges to one of the corners of the hypercube $[-1,1]^n$. The prime limitation of the BSB network comes from the fact that, the location of the attractors must be predefined as the vertices, on the aforementioned hypercube.

MAXNET. MAXNET is a competitive network, in which only one neuron will have a non-zero output when the competition is completed. The network consists of interconnected neurons and symmetric weights. There is no training algorithm for MAXNET and the weights are fixed (Fausett 1994). Its application procedure includes two steps: activation and initialisation of weights, and updating the activation of each unit until only one unit responds. MAXNET is suitable for situations where more information is needed than can be incorporated.

Similar to feature recognition, the above ANN architectures are also widely adopted in the applications of CAPP, which will be further discussed in section 4.

2.2 Input and output node characteristics

2.2.1 Input node characteristics

Neural networks, typically, although not necessarily, receive an n-unit vector. Each unit of the input vector could be a) integer value, which is given a particular integer instead of the original value; b) real value, which is encoded with numerical values ranging from 0 to 1; and c) in binary form, which is represented by only two characters (i.e. 0 and 1). The problem then is how to convert product model (i.e. geometry and topology) and engineering information (i.e. tolerances, materials and operations) to a format suitable for neural network input in a convenient and efficient way. There are three aspects to be resolved:

1. *Complete and precise information:* It is extremely important that this representation describes input information completely, and does not distort any information.

2. *An identifiable format:* Each piece of the input information, such as a feature belonging to different class, or an attribute leading to different decisions, must have a unique input representation without overlaps.

3. *Numerical encoding:* Based on the characteristics of neural networks, the input

parameters need to be converted into numerical values, like the three general formats mentioned above: integer value, real value and binary form. The type of encoding relates back to the recognition technique employed.

Various examples of this process have been: Afzal and Meeran (2006) transformed pixcelated *.AVI pictures of machined features into a chain of code of integers. Ding et al (2005) produced a vector of integers related to recognized feature, Deb et al (2001; 2006) used a vector of binary number for the same purpose. In order to determine feature clusters, Chen and LeClair (1993) represented a feature with a *(6+n)*-unit vector in binary form, which defines the *six* approach directions and *n* tool types. Park et al. (2000) defined the input values from 0 to 1 according to its real values, e.g. 0.5000 for a hardness unit for a real value of 225 BHN, and 0.4366 for a cutting speed unit for a real value of 80 m/min. Park et al. (1996) used 15 input parameters concerning seven factors, such as the feature type, ratio of feature width to depth, tool length, and tool material. A class number is given for each parameter based on its real value, e.g. the class number is 6 if the ratio of selected tool length over standard length is 2. Osakada and Yang (1991) converted the cross-sectional shape data of the product into standardized image data for the input. They use 12 "colours" to represent 12 outer or inner geometric primitives, such as cylinder and cone. Half of the product shapes are converted into a 16*16 "colour" data image. These 256 units are regarded as the input to the neural network. This representation can only be used for rotationally symmetric products. The input representation is crucial for the success of neural network; the conversion of input format for feature recognition will be further discussed at Section 3.

2.2.2. Output node characteristics

In the reviewed literature, five main output formats have been proposed: Output vector in ordered binary form, output vector with special values, one-unit output in binary form, one-unit output in integer form and output matrix.

Output vector in ordered binary form. The output vector is usually applied for operation selection, machine or cutting tool selection. Commonly, it consists of a number of neurons, each with a value (i.e. 0 or 1) showing whether the corresponding item (machining operation, machine tool or cutting tool belongs to the process plan or not. For instance as output vector consists of eight neurons representing respectively drilling, reaming, boring, turning, taper turning, grooving, grinding and precision. If the value of a neuron is '1', the corresponding operation is needed for the feature; otherwise, the value is '0', e.g. a hole requires the drilling operation, so the first neuron is assigned the value '1'. The sequence of the vector represents the sequence of the operations, e.g. reaming is usually performed after drilling. Li *et al* (1994) used a *4*-neuron vector corresponding to the abrasive type, grade, grit size and bond, for grinding. Le Tumelin *et al* (1995) designed a 23-neuron vector. Joo et al. (2001) used a 11-neuron vector corresponding to Machine type, Table size(L), Table size (W), Table load capacity, Stokes, x,y and z, Spindle power, speed range, feed-range and minimum accuracy. Ahmad and Haque (2002) had 10-neurons relating to rough training, semi-finishing turning, finish turning, facing, taper turning, chamfering, form turning, cut-off, grinding and lapping.

Output vector with special values. Each neuron in the output vector has a possible value that the corresponding parameter may assume. Santochi and Dini (1996) developed a system for selecting the eight technological parameters of a cutting tool. For example, to select a normal clearance angle α_n, the number of output neurons is 5 which represent 4°, 5°, 6°, 7°, 8° respectively. The neuron with the value '1' represents the optimal value and '0.5' a second choice. Many feature recognition systems adopt the output vector, each neuron of which represents a feature class, such as the work of Chen and Lee (1998), Nezis and Vosniakos (1997) and Ding and Yue (2004). In addition, Hwang (1991) applied six neurons output vector, each of which representing the class, name, confidence factor, the main-face name, the list of associated faces of the face found, and the total execution time.

One-unit output in binary form. This output format has only one unit whose value is either 0 or 1 (Mei et al. 1995). The output shows which surfaces should be used as manufacturing datum's. For instance, '1' means that the surface will be used for the part setups and '0' means that it has nothing to do with part setups.

One-unit output in integer form. Each discrete integer is concerned with a special class (Chen and LeClair 1993). The output integer represents a cluster of features according to the approach directions and the tool types.

Output matrix. Shan et al. (1992) devised a binary incidence matrix V (n*n) in which the rows denote operations and the columns correspond to sequences. The value '1' indicates that a specified operation is performed. Because each operation is performed only once and only one operation is carried out at a time, one and only one of the entries in each row and column should take the value of 1 whereas the rest should be set to 0. Zulkifli and Meeran's system (1999) is another typical example: the output is a binary matrix $O=[b_{ij}]$, $1 \le i \le 2$, $1 \le j \le 5$, with b_{1j} representing the code for the feature recognised, and b_{2j} showing the tool accessibility to machine the feature, namely $+x$, $-x$, $+y$, $-y$ and $-z$ directions.

2.3 Network learning strategy

The learning ability of an ANN is established by its architecture, and the method chosen for training. The term machine learning algorithm has been applied to ANN, since altering the connection weights causes the network to learn the solution to a problem. The strength of connection between the neurons is stored as a weight-value for the specific connection. The system learns new knowledge by adjusting these connection weights. The training process is classified as supervised learning or unsupervised learning.

2.3.1. Supervised training. In producing this survey, the authors noted, that most ANN's employed for feature recognition use supervised training with a back propagation (BP) algorithm (Ding and Yue, 2002, Zulkifli and Meeran, 1999, Marques et al., 2001). With BP algorithm, the network gives reinforcement for how it is performing on a task. The actual and desired outputs are compared, and the network's error is calculated as the difference between its output and target. Information about errors is also filtered back through the system and is employed to modify the connections between the layers, giving improved performance. After a number of iterations, the output will converge towards the target (Nezis and Vosniakos 1997, Ding and Yue 2004 and Chen and Lee 1998). Basically, the BP algorithm consists of two basic steps:

- initialisation of weights; for example, in Gu et al.'s work (1997) all the weights were initially randomly set in the range 0 to 0.1; and

- repetition of training until the error is acceptably low. For instance, Gu et al. (1997) mapped the selected pattern pairs to reinforce the weights until the deviation between the training output and the target of each sample converged to a pre-defined error goal (e.g. 0.05).

Back-propagation methods have proven highly successful for applications (Ahmad and Haque 2002, Deb et al. 2001). Currently, four major groups under BP methods are adopted:

The Delta Rule. One of the back-propagation learning algorithms is the delta rule, which is based on cumulative error; known as the least mean squares. This learning rule changes the connection weights, so as to minimise the mean squared error between the network output and the target over all training patterns. Sakakura and Inasaki (1992) chose the delta rule for both a feed-forward network and a BSB network. In the three-layer feed-forward network, the weight connecting neuron j in the hidden layer to neuron k in the output layer is updated as follows:

$$\Delta w_{kj} = \eta_f \sum_p \delta_{pk} o_{pj} \tag{7}$$

$$\delta_{pk} = (t_{pk} - o_{pk}) f'(a_{pk}) \text{, where} \tag{8}$$

$f()$ is the output function of neuron, η_f is the learning coefficient of the FF network, p is the learning pattern number, t_{pi} is the learning value of neuron i for learning pattern p, a_{pi} is the status value of neuron i for learning pattern p, and o_{pi} is the output value of neuron i for learning pattern p.

For the BSB network, the modified value of the weight which interconnects neuron m and neuron n is calculated as the following:

$$\Delta w_{mn} = \eta_b \sum_p (t_{pm} - w_{nm} t_{pn}) t_{pn} \text{, where} \tag{9}$$

η_b is the learning coefficient of BSB network, and t_{ji} is the learning value of neuron i for learning pattern j.

The principle drawback of this method is its sensitivity to the result of initialization of the synaptic weights, and the slowness of the convergence rate (Johannet et al., 2007). To overcome this, the Levemberg-Marquart approach is normally employed.

Levemberg-Marquardt Approximation. A back-propagation algorithm using the approximation of Levemberg-Marquardt is also used in some applications (Santochi and Dini, 1996). This algorithm allows a better performance in terms of training time. However, it may require a very large storage space for some complex situations. The matrix of the connection weights is updated through the following equation:

$$w = (J^T J + \mu U)^{-1} J^T e \text{, where} \tag{10}$$

J is the Jacobian matrix of derivatives of the errors to each weight $w_{i,j}$, μ is a scalar, U is the unit matrix, and e is the error vector of the network.

Conjugate Gradient Algorithm. Conjugate gradient algorithms make a search along conjugate directions, which produces generally faster convergence than in the steepest directions. In general, weights are updated by an optimal distance (learning rate, α_k) along the current search direction

$$W^{(k+1)} = W^{(k)} + \alpha_k d_k. \tag{11}$$

d_k is the steepest descent direction, which is chosen as a linear function of the current gradient (g_k) and the previous search direction (d_{k-1})

Batch Training. Either the delta rule or the Levemberg-Marquardt approximation is used as the on-line learning rule. The batch training is an off-line training process. Rather than adjust the weights after each pattern presentation, batch training accumulates the errors over the whole training set, and adjusts each weight according to the accumulated errors. It can generally be expressed as follows (Principe *et al.,* 2000):

$$\Delta w_{ji} = \eta \sum_{p} \delta_{out} H_{in} \ , \quad \text{where} \tag{12}$$

the subscripts *in* and *out* refer to the net input and output signals associated with a given unit, and *i* and *j* refer to the connection from unit *i* to unit *j*.

The form of δ varies depending on the type of layer to which the formula applies. In some cases it is advantageous because of its smoothing effect on the correction terms and increasing of convergence to a local minimum. Because of this effect Devireddy and Ghosh (1999) trained a system with a batch training back-propagation algorithm.

2.3.2 Unsupervised learning algorithm.

With an unsupervised learning algorithm, the training set only contains input samples; no desired or sample outputs are available. The neural network must construct an internal model that captures regularities in input training patterns, instead of measuring its predictive performance for a given input. Hence this method is also called self-organisation. With self organizing maps (SOM), as used by Meeran and Zulkifli (2002), the training process is to associate different parts of the SOM lattice to respond similarly to certain input patterns. The weights of the neurons are initialized either to small random values or sampled evenly from the subspace spanned by the two largest principal component eigenvectors. The latter alternative speeds up the training significantly because the initial weights already give good approximation of SOM weights. When a training sample is given to the network, the Euclidean distance to all weight vectors is computed. The neuron with weight vector most similar to the input is called the Best-Matching-Unit (BMU). This can also be termed as the 'winning node' (Picton, 1994). The weights of the BMU and neurons close to it in the SOM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and is smaller for neurons physically far away from the BMU. The update formula for a neuron with weight vector Wv(t) is:

$$Wv(t+1) = Wv(t) + \Theta(v,t)\alpha(t)(D(t) - Wv(t)) \tag{13}$$

where α(t) is a monotonically decreasing learning coefficient and D(t) is the input vector. The neighborhood function Θ(v,t) depends on the lattice distance between the BMU and neuron v. In the simplest form it is one for all neurons close enough to BMU and zero for others, but a Gaussian function is a common choice, too. Regardless of the functional form, the neighborhood function shrinks with time. At the beginning, when the neighborhood is broad, the self-organizing takes place on the global scale. When the neighborhood has shrunk to just a couple of neurons, the weights are converging to local estimates.

In CAPP applications, a logical AND/OR operation-based unsupervised learning approach is used. Chen and LeClair (1993) clustered features based on the approach direction and tool type and then generated a process plan using an Episodal Associative Memory (EAM) approach. The AND operation was applied to solve multiple approach directions for some features. If the digit is 1 for the corresponding approach direction, the update weight for the cluster *j* is

$$^{a}b_{ij}(s+1) = {}^{a}x_i^{(p)} \ AND \ {}^{a}b_{ij}(s) = {}^{a}x_i^{(p)} \, {}^{a}b_{ij}(s), \quad \text{where} \tag{14}$$

$^{a}x_i^{(p)}$ is the approach direction sub-pattern, $<+x,+y,+z,-x,-y,-z>$, of pattern *p*.

In the meantime, the OR rule is used to update the weight so that the probability of common tools can be increased. If the digit is 1 for the corresponding tool, then $b_{ij}\ (s+1)$ is modified according the following equation (Chen and LeClair, 1993):

$$^{t}b_{ij}(s+1) = {}^{t}x_i^{(p)} \ OR \ {}^{l}b_{ij}(s) = f({}^{l}x_i^{(p)} + {}^{l}b_{ij}(s)), \quad \text{where} \tag{15}$$

$^{t}x_i^{(p)}$ is the tool sub-pattern, and $f(\eta)=1$ if $\eta/1$, else $f(\eta)=0$.

2.4 Software environments employed

Friendly user-interface is another issue that needs to be considered for CAD/ CAM system to be taken up in an industrial environment. In reviewing the published research it can be seen that a variety of specialized neural network toolkits and software packages have been employed.

2.4.1 Specialize or dedicated software:

- Mathworks, Matlab has a dedicated toolkit for neural network, this tool was used by Ding et al. (2005), Ahmad and Haque (2002), Wong and Lam (2000), Chakraborty and Basu (2006);

- Tragan-neural networks was employed by Ozturk and Ozturk (2001) and Ozturk and Ozturk (2004) in their research in feature recognition;
- Neuframe4 was employed by Deb et al. (2001; 2006);
- Neuralware was used by Korosec and Kopac (2006).

2.4.2 Standard programming software:

- Jun et al. (2001) developed there own environment using C++ and Open GL languages for geometric feature recognition. What is not evident from the hybrid approaches is how the ANN and other intelligent agents have been integrated. Using each tool individually, does not offer a sensible option for a tool being developed for a production environment;
- Joo et al. (2001) implemented their neural network algorithm in the C programming language.

What cannot be ascertained from the literature survey, is why the researchers have selected these software options? It can only be assume that for ease researchers have used the specialized software whereas the standard programming software offer the user more flexibility i.e. the need to develop or adapt the algorithm.

3. FEATURE INPUT PREPARATION

In the production of solid models there are two representations: Boundary representations (B-rep) Braid (1974), and Constructive Solid Geometry (CSG) (Mäntylä, 1988). B-rep stores a solid model with low level entitles such as vertices and faces, where CSG stores a 'tree' of low level primitive volumes, with respective Boolean details used to construct the solid model. CSG is limited due to the non-uniqueness of the CSG tree, because of this, B-rep is deemed more useful for the applications presented in this paper (Allada and Anand, 1996). Manufacturing feature recognition is a complicated process, for which entire information in a solid model including both geometric and topologic information needs to be input. However a set of integer values is the normal representation that is offered to the neural network. This raises the problem of how to convert a solid model to a format suitable for neural network input in a convenient and efficient way. Basically, there are three characteristics for a satisfactory input representation (Yue and Ding, 2001):

- the solid model information (e.g. faces, edges and vertices) for feature recognition;
- a format identifiable by the input layer; and
- an unique input representation without overlaps.

The CAD representations are presented in two forms, either 2D drawing/ pictures in first or third projections such as Afzal and Meeran (2006). Or, in 3D orthogonal views such as Marquez et al (1999) and Chakraborty and Basu (2006). With 3D orthogonal views, face, edge, and vertex values are derived, because of this, the orientation of these models is immaterial (Ding et al, 2005). Orientation can be can issue these values are not derived in the process as shown by Wang and Liu(1993), resulting in wrong feature classification. While investigating the published literature, the input representation can be broadly classified into the following types:

2D feature representation. The standard representation of parts on computer generated drawings, are in the wire-frame profiles. These can be subdivided into connected loops of edges. Peters (1992) proposed an ordered triplet (C_i, A_i, L_i) to represent each edge of a connected loop, where C_i, A_i and L_i are the curvature, interior angle and arc length of the *i*th element respectively. An encoded feature vector of the triplet (C_i, A_i, L_i) for a given profile is used as the input. Chen and Lee (1998) produced an encoded feature vector, in which the representation of each edge is 7-tuple in the form: (Li Ai Ci Ji OLi OAi OCi) where Ci, Ai and Li are the curvature, interior angle and arc length of the *ith* element respectively, Ji is the intersection type between the line segment and its subsequent line segment, and OLi, OAi and OCi are the ordinal values assigned to Li, Ai, Ci respectively. The ordinal values are assigned to the parameter in order to capture the magnitudes. The input layer has thirty-five neurons corresponding to five edges, seven neurons representing each edge. The number of neurons in the layer will need to be increased as the number of edges increases.

Graph matching method. Graph matching method organises a B-rep model of a part into a stereotypical sub-graphs structure where the nodes represent faces, edges or vertices and the arcs represent the relationships of any two entities. Joshi and Chang (1988), De Floriani and Bruzzone (1989), Lentz and Sowerby (1993) have pursued this method. The graph-based recognition approach has an advantage over the others due to the graph nature of B-rep-based solid model (Lam and Wong, 2000). It is effective, but suffers from two significant drawbacks: the large computational expenditure of dealing with complex components, and the deficiency of dealing with interacting features.

Face adjacency matrix code. A face adjacency matrix is a 2D array of integer vectors converted from a solid model. Each integer vector represents a face and its relationship to another face, i.e. adjacency or common edge. The length of an integer vector depends on the number of parameters considered for the recognition of a feature. In Prabhakar and

Henderson's work (1992), the vector has eight integers indicating characteristics such as edge type, face type, face angle type, number of loops, etc. This method is limited to features defined by a primary face and a set of secondary faces. It cannot differentiate between features with the same topology but different dimensions of compound faces.

Face score vector The concept of face score was originally presented by Hwang and Henderson (1992) and employed by Lankalapalli et al. (1997) where the face score is defined as Fs = *f*(Fg , Eg, Vg, At), given that Fg , Eg, Vg are the information about the face , edges and vertices, and At gives the adjacency among the faces, edges and vertices (cf. table 1). A modified faced vector value assignment was proposed by Marquez et al. (2001) who highlighted the differences between concavity and convexity of face and edges. The edge scores for all surfaces the vertex score (V) can be calculated from the following equation.

$$V = \sum_{i=1}^{m} E_i \tag{16}$$

Table 1 Assignment of value to obtain face values *Value of face type*

EDGE Scores	
Convex edge	+0.5
Concave edge	-0.5
Face geometry scores Fgs	
Planar surface	0.0
Convex surface	+2.0
Concave surface	-2.0
Spline surface	0.0

By adding the average value of vertex scores in the face and face geometrical value (Fgs), the face score (Fs) can be calculated using the following equation.

$$F_s = \sum_{j-1}^{n} \frac{V_j}{n} + F_{gs} \tag{17}$$

The face score approach was also employed by Ozturk and Ozturk (2001) where the features are defined as input vectors in terms of vertices, edges and faces and presented to neural network as follows:

$$Feature_j(t) = \sum wij(t)F_i - \theta \qquad (18)$$

Where *Fi* is the boundary representation based face score, *t* is the number of training patterns and θ is the threshold. The core limitation of this approach comes from the fact that this representation can recognise a very limited number of compound features, and there is no one-to-one correspondence between feature patterns and features.

Attributed adjacency matrix. An attributed adjacency matrix (Nezis and Vosniakos1997) describes the geometry and topology of a feature pattern is converted from the attributed adjacency graph (AAG) (Joshi and Chang, 1988), these are used to produce adjacency matrices (AM). In Nezis and Vosniakos' research (1997), the AM is a 2D, square, binary matrix with two triangular areas: an upper and a lower which are the convex and concave spaces respectively. *AM*[*i, j*] and *AM*[*j, i*] indicates the connection between the *i*th and *j*th faces of the object. One of them belongs to the concave space and the other to the convex space. The representation vector is formed by firstly breaking the AAG into sub-graphs which are converted into AM using a heuristic method. Then each matrix is converting into a representation vector (RV) by interrogating a set of 12 questions about the AM layout and the number of faces in the sub-graph; and a binary vector is formed combining the 12 positive answers and the other 8 elements corresponding to the number of external faces linked to the sub-graph. Major limitations to this method come from the fact it can recognize planar and simple curved faces, but not features related to secondary feature faces, such as dove-tail slots. And, although simple interacting features can be recognized, the system does not consider interacting features that share a common bottom face.

F-adjacency matrix and V-adjacency matrix. To solve the problems of AAG noted above, an input representation with two matrices is proposed by Ding and Yue, (2004). The input scheme is based on the topological and geometrical information of a feature as a spatial virtual entity (SVE), which is an equivalent to the volume removed from the initial material stock in order to obtain the final boundary of the feature.

a) *F-adjacency matrix*: is defined as $I_F = [a_{ij}]_{i\times j}$, where 1<=*i*<=5 and 1<=*j*<=5. The middle elements of I_F, i.e. a_{ii}, show the type of the *ith* face, (given in Table 2). Other elements of I_F (i.e. a_{ij}, where $i \neq j$) indicate the connection between the *i*th and *j*th faces of the object. A numerical value between 0 and 9, as shown in Figure 6, is allocated according to the relationship between the two faces. The layout presentation of I_F is symmetrical so that the input format consists of 15 nodes, $a_{11}, a_{12}, \ldots, a_{15}, a_{22}, a_{23}, \ldots, a_{25}, \ldots, a_{55}$. All of these faces are sequenced, firstly, according to their types and relationships with adjacent faces. An example of F-adjacency matrix is given in Figure 7.

Table 2 Value of face type

Face type	Value	Face type	Value
Cylindrical face	1	Semi-spherical face	5
Part-cylindrical face	2	Planar face	6
Conical face	3	Linear-group	7
Part-conical face	4	Circular-group	8

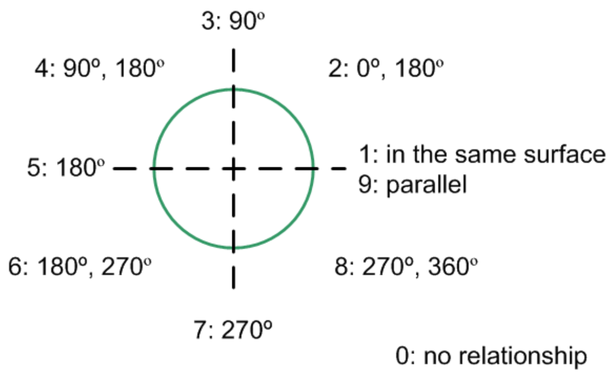

Figure 6 Values of relationship between two faces

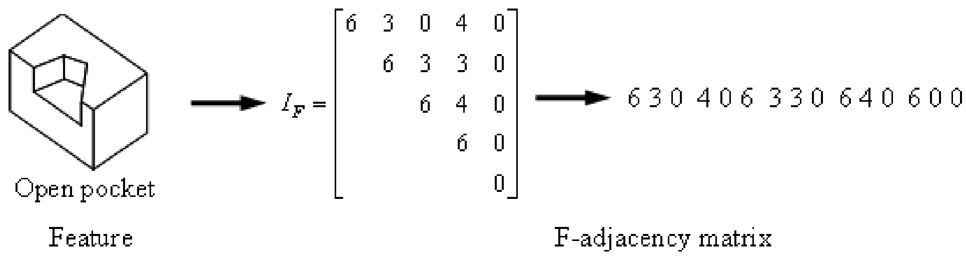

Figure 7 An example of F-adjacency Matrix

b) *V-adjacency matrix:* is defined as I_V =[*bij*]6x6, I_V showing the relationships between VF faces in the SVE. Where VF refers to a face that forms the boundary of its SVE, but does not physically constitutes the basic shape of the feature. Each row and column of I_V represents 6 directions:+x, -x, +y, -y, +z, -z. The middle element, b_{ii}, shows whether there is a VF in the corresponding direction. If in the ith direction (e.g. $+x$), the given SVE exists a VF face, then $b_{ii} = 1$; if not, $b_{ii} = 0$. Other elements, b_{ij} ($i \neq j$), describe whether the two VFs, corresponding to direction i and direction j, are connected or not (i.e. 1 or 0). Similarly, the symmetric characteristic of V-adjacency matrix is used to simplify the input. A vector consisting of 21 codes is input to the neural network, that is, $b_{11}, b_{12}, \ldots, b_{16}, b_{22}, b_{23}, \ldots, b_{26}, \ldots, b_{66}$. Figure 8 shows an example of V-adjacency Matrix.

To avoid a big size of matrix, if the number of faces that physically constitute features is larger than 5, two rules for simplification are given to generate, linear group and circular group.

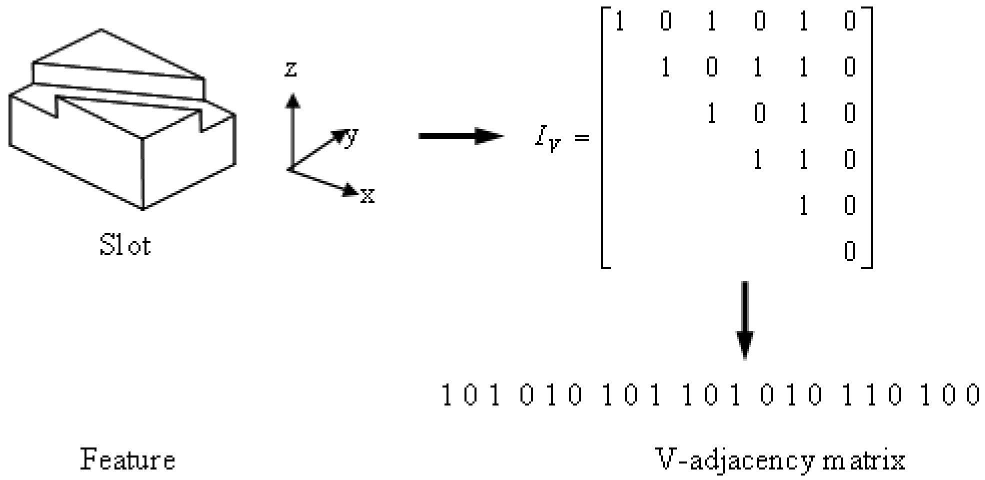

Figure 8 An example of V-adjacency Matrix

2D input patterns of 3D feature volume. Zulkifli and Meeran (1999) presented an input matrix based on a cross-sectional method. The B-rep solid model is searched through cross-sectional layers and converted into 2D feature patterns, which are then translated into a matrix appropriate to the network. Four input matrices correspond to four feature classes: simple primitive, circular, slanting, and non-orthogonal primitive features. There are several disadvantages, e.g. simple primitive features are limited to four rectangular vertices, and features with non-orthogonal faces in the z direction cannot be dealt with.

Partitioned view-contours of a given object. The given object is represented by nine partitioned view-contours from $+x$, $-x$, $+y$, $-y$, $+z$, $-z$, x, y and z respectively. The vector is built in three steps.

- A graph with a representative ring code is defined from a partitioned view-contour in which the nodes represent the regions and the arcs represent the adjacency relations among the regions; the representative ring code is a cyclic string of digits formed for each region based on both the graph and a two-layer octal coding system;

- Based on the weighting value computed with the representative ring code, the graphs are converted to a reference tree in which each node is associated with 6+m values using heuristics from several experiments, assuming each graph node has at most m+1 adjacent nodes;
- The vector is then generated with the first 6+m elements for the tree root and the next 6+m elements for the second tree node ranked, and so on.

This method has shown only to be suitable for block-shaped objects with rectangular view-contour boundaries. Feature classes were defined as: slot, step, pocket, protrusion, blind-slot, corner-pocket, through hole and blind hole, as shown in Chuang (1999).

Simplified skeleton. A simplified skeleton is a tree structure with line segments (Wu and Jen 1996) represented by an input vector that is formed in the following process:

- a standard tree structure in which each parent branch has the same number of descendants is predefined;
- a simplified skeleton with several standard trees is represented;
- six attributes of a branch in the standard tree to describe each real link (non-null assignment branch) and the spatial relationships among them are defined; and
- the standard tree is converted into a vector in which each element corresponds to a branch; there can be several standard trees for a simplified skeleton.

Although this representation can be used to classify 3-D prismatic parts, a disadvantage comes from the fact that only the contour information of the part is considered.

A summary of the achievements of previous feature recognition research is presented in table 3.

Table 3 Capabilities of ANN-based feature recognition systems

Authors	Limits to features recognised
Afzal and Meeran (2006)	2-D features: rectangle, slot, trapezoid, parallelogram, V-slot and triangle
Chakraborty and Basu (2007)	Slots, Circular and semicircular pockets
Chuang (1999)	3-D block-shaped components
Chen and Lee (1998)	2-D features: rectangle, slot, trapezoid, parallelogram, V-slot and triangle
Ding and Yue (2004)	3-D Prismatic parts
Lankalapalli et al (1997)	Simple primitive features such as blocks, step, slot, blind slot, hole and pocket
Jun et al (2001)	Simple primitive features such as blocks, step, slot, blind slot, hole and pocket
Marquez et al (1999)	Features such as slot, blind slot, step, pocket and hole which only have planar faces
Marquez et al (2001)	Features such as slot, blind slot, step, pocket and hole which only have planar faces
Meeran and Zulkifli (2002)	1. Simple primitive features defined by four rectangular vertices, such as step, slot, blind slot and pocket 2. Circular features 3. V slanting features, such as tapered pockets, wedges and V-slots 4. Non-orthogonal faces in the x and y directions
Nezis and Vosniakos (1997)	1. Features such as slot, blind slot, step, pocket and hole which only have planar faces 2. Simple curved faces
Ozturk and Ozturk (2001)	1. Simple primitive features defined by four rectangular vertices, such as step, slot, blind slot and pocket 2. Circular features
Wong and Lam (2000)	Primitive machined features
Zulkifli and Meeran (1999)	1. Simple primitive features defined by four rectangular vertices 2. Circular features 3. Non-orthogonal faces in the x and y directions

4. ANN CAPP INTEGRATION

Computer aided process planning belongs to a group of constrained optimization problems that are NP-hard. These constraints are defined as *hard* and *soft* (Matthews et al., 2006).

Where the *hard constraints are* concerned with manufacturing precedence's i.e. The hole must be drilled before it is reamed, and, at a higher level, s*oft constraints* can impose restrictions on performance criteria, tardiness, cost etc. A variety of numerical techniques have been employed to propose solutions to such problems, ANN are one such technique. Research using ANN's for CAPP aligns itself very closely to another NP hard problem, that of scheduling. A recent study by Akyol and Bayhan (2008) on ANN's and production scheduling, will give the readers a good comparison to a related topic. ANNs also offer an encouraging approach to CAPP due to their learning ability. This section details the ANN techniques used in CAPP.

Gu et al. (1997) employed a three-layer feed-forward network with a 5-neuron hidden layer for manufacturing evaluation. Santochi and Dini (1996) proved in their experiment that a three-layer feed-forward network with a suitable number of neurons for each layer is the best architecture for selecting technological parameters for a cutting tool using the hyperbolic tangent sigmoid function. Park et al. (1996) developed a four-layer neural network to modify cutting condition based on several tests. Their network has a 15-neuron input layer, two 15-neuron hidden layers and a single-neuron output layer. A common factor that links approaches is the inputs nodes are the features attributes and the number of nodes relates to the number of features attributes the respective approach offers. The output layers relates to the number of feasible machining operations. Le Tumelin et al. (1995) proposed a 5-layer feed-forward network to determine appropriate sequence of operations for machining holes.

It was shown by Hopfield and Tank (1985) that if an energy function can represent the optimization problem, then a Hopfield network that relates to this energy function can be used to minimize this function and provide a near optimal solution. Chang and Angkasith (2001), Yan and Qiao (2004) and Zhao et al. (2002) employed a Hopfield network for operational sequence planning of prismatic parts and EDM sequencing respectively. Supposing the number of operations is *n*, the network is then composed of n^2 neurons, each identified by double subscripts: the operation and the sequence to be executed.

$$E_1 = \frac{1}{2}A\sum_i \sum_j \sum_{l \neq j} v_{ij}v_{il} + \frac{1}{2}B\sum_j \sum_i \sum_{k \neq i} v_{ij}v_{kj} + \frac{1}{2}C[(\sum_i \sum_j v_{ij}) - n]^2 \tag{19}$$

$$E_2 = D\sum_i \sum_j \sum_{k \neq i} \sum_{i > j} p_{ki}v_{ij}v_{kl} \tag{20}$$

$$E_3 = F\sum_j^{n-1} \sum_i \sum_{k \neq i} t_{ki}v_{ij}v_{k,\,j+1}, \quad \text{where} \tag{21}$$

A, B, C, D and F are constants, v_{ij} is the output of neuron in position (i, j) of the matrix, and t_{ki} is tool travelling time form the position i to k.

The change in energy ΔE_{ij} due to a change in the state of neuron is:

$\Delta E_{ij} = [\sum_{l} \sum_{k} w_{kl,ij} v_{kl} + I_{ij}] \Delta v_{ij}$, where I_{ij} is a bias weight. The weight (22)

connecting neurons kl and ij can be found as the following: $w_{ij,kl} = -A\delta_{ki}(1-\delta_{lj})-B\delta_{lj}(1-\delta_{ki})-C-Dp_{ki}\phi_{lj}(1-\delta_{ki})-Ft_{ki}\delta_{lj+1},$ where

$$\delta_{xy} = \begin{cases} 0 & if \quad x \neq y \\ 1 & if \quad x = y \end{cases} \tag{21}$$

$$\phi_{xy} = \begin{cases} 0 & if \quad x \leq y \\ 1 & if \quad x > y \end{cases} \tag{23}$$

The Hopfield network provides one of the strongest links between information processing and dynamics. However, spurious memories limit its capacity to store patterns. Another issue using ANN for such problem is the energy function generally can only define the hard constraints of any CAPP problem.

BSB and MAXNET are typically used in multi-type architectures. Sakakura and Inasaki (1992) used a BSB with a three-layer feed-forward network in a CAPP system. The numbers of neurons assigned for the dressing depth of cut, dressing feed and surface roughness are: ‘5’, ‘5’ and ‘9’ respectively. The initial values are given by a feed-forward network run at the same time. The BSB repeats, performing a calculation until the output value of each neuron converges to a certain value. Knapp and Wang (1992) utilized a co-operating architecture combining a three-layer feed-forward network and a MAXNET, where the MAXNET is used to force a decision between the competing operation alternatives. Yahia et al. (2002) proposed an approach, which is composed of two related feed forward with a parallel structure (cf. figure 9). NN1 is capable to select machining operations and NN2 is capable to select machining tools to be used. The limitation of these networks is that the location of the attractors must be predefined as the vertices of the hypercube. Not always practical for the applications in this paper.

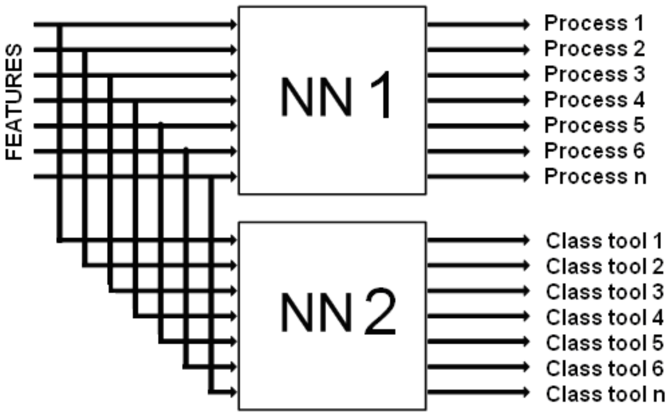

Figure 9 Hybrid topology

Table 4. Realization of ANN-based CAPP systems

Author(s)	ANN type	Functions	Process
Ahmed and Haque (2002)	3-Layer FF network	Operation selection and operation sequencing	Surface machining of cylindrical components
Ben Yahia et al (2002)	Twin	Operation sequencing	Machining operations for prismatic components with regular machining features
Chambers and Mount-Campbell (2002)	3-Layer FF network	Buffer size selection for operation sequencing	Batch manufacturing operations
Chang and Angkaisth (2001)	Hopfield	Operation sequencing	Machining operations for prismatic components with regular machining features
Chen et al (2005)	ART1	Operation sequencing	Generic process planning
Deb et al (2002)	3-Layer FF network	Operation sequencing	Machining operations for rotational axis-symmetric components
Deb et al (2006)	3Layer FF network	Operation sequencing	Machining operations for rotational axis-symmetric components
Devireddy and Ghosh (1999)	3-Layer FF network	Operation selection and operation sequencing	Machining operations for rotational components

Devireddy and Ghosh (2002)	3-Layer FF network	Operation selection and operation sequencing	Machining operations for rotational components
Gu et al (1997)	3-Layer FF network	Operation sequencing	Machining operations for prismatic components with regular machining features
Joo et al (2001)	Multi layered FF network	Operation sequencing	Machining operations
Korosec et al (2005)	3-Layer FF network	Tool path strategy and sequencing	Machining operations
Korosec and Kopac (2006)	Bespoke SOM algorithm	Tool path strategy and sequencing	Machining operations
Ming and Mak (2000a)	Kohonen SOM	Set-up planning	Machining operations
Park et al (1996)	4-layer FF network	Generation of modified cutting conditions	Milling and turning for sheet metal
Park et al (2000)	Fuzzy ARTMAP network	Generation of cutting conditions	Milling operations
Yan and Qiao (2004)	Hopfield	Operation sequencing	General machining operations
Zhao et al (2002)	Hopfield	Operation sequencing	Electro discharge machining
Zhong et al (2004)	Back-propagation	Operational sequencing	Marine pipe machining

The CAPP approaches noted above employed ANN's to identify machine, parameters or condition for manufacture. Although they aim to aid the investigation for an optimal setup and manufacture, they do not optimize the actual operation and the networks identified are limited for this purpose. Later research has investigated the use of ANN for the constrained optimization problem. Table 4 presents the achievements of ANN-based approaches to CAPP.

4.2 Hybrid approaches to CAPP using ANN

ANN techniques used to solve constrained optimization problems have major limitations, specifically:

- Lack of systematic investigation of the framework and methodology of CAPP. Most of solutions are designed for specific activities (e.g. tool parameters selection, cutting condition generation) and specific applications (e.g. cold forging, grinding operations), which cannot be used in different industrial environments;
- Low efficiency and quality of process planning. For a process planning system based on the machined surface, with the number of machined surface increasing, the

efficiency and quality of reasoning can not guarantee, especially for a complex component; and

- Limited previous research has considered prismatic components. It is mainly due to the complex geometrical representation of the 3D prismatic components and the intricate nature of cutting mechanism in milling. It is difficult for a CAPP system to plan a solution for all possible components.

To overcome some of these drawbacks, researchers have considered the integration of other 'intelligent' techniques. This section presents some of these.

Expert and ANN. Expert systems employ explicit rules, such as manufacturing and production rules. However, CAPP is not only concerned with explicit judgements but also implicit judgements. For example, how does a system run when it cannot guarantee all manufacturing rules are satisfied at the same time? On the contrary, neural networks are an implicit reference method, which is formed through a training process with a set of examples. Therefore, the incorporation of expert system and neural network techniques in a CAPP system can benefit from the advantages of both and make the system more flexible and adaptive. Ming et al. (1999) proposed a hybrid intelligent inference model combining an expert system and a neural network for CAPP consisting of inference functions, global inference control strategy, hybrid control manager, cooperative communication processor, hybrid process knowledge base, and CAPP inference methods. The hybrid control manager is first executed to judge the initial condition, and to select the appropriate control strategy (by the expert system or neural network) or other functions (through the calculation function or optimisation function). Other examples of this kind of hybrid approach include the systems proposed by Kandel and Langholz (1992) and Medsker and Liebowitz (1994).

Fuzzy logic ANN hybrids There is often uncertainty or intangible factors in CAPP, especially during manufacturing evaluation, such as geometrical complexity and manufacturability. Fuzzy set theory may be employed as a solution to uncertainty. Another consideration comes from the fact that new manufacturing methods and technological developments may influence process planning, (e.g. new machine tools purchased). Thus, adaptability is needed for CAPP. Based on the above requirements, it is useful to incorporate fuzzy logic techniques to perform certain CAPP tasks. Chang and Chang (2000) developed an artificial intelligent CAPP system integrating neural network, fuzzy logic and expert system techniques. Their system consists of a back-propagation neural network for evaluating the manufacturability of important features of the component. A fuzzy logical back-propagation neural network (FL-BPN) for evaluating the suitability of the existing plans stored in the database. The FL-BPN has five layers: the input layer, membership function layer which fuzzifies the crisp input values, AND layer where each neuron represents the premise part of a rule and is connected with an 'AND' operator, OR layer where each neuron represents the conclusion part of a rule and is connected with an 'OR' operator, and defuzzification layer which defuzzifies the final evaluating result functional modules for process planning using an expert system, such as manufacturing process selection, machine selection, cell selection, fixture selection, part setup determination, cutting tool selection, machining parameters calculation, and final operations sequencing.

Amatitik and Engin Killic (2007) presented an intelligent process planning system using STEP features (ST-FeatCAPP) for prismatic parts. The contribution of this work was its ability to negate complex feature recognition and knowledge acquisition problems highlighted in section 3. They employed three three-layer FF network for machine operation, cutting tool section and feature selection, and a fuzzy logic to select machining parameters. Their approach maps a STEP AP224 XML data file, and produces the corresponding machining operations to generate the process plan and corresponding STEP-NC in XML format.

Genetic algorithm ANN hybrids. A limitation of ANN's is their tendency to become trapped in local minima. To solve this, ANN have been combined with Genetic algorithms (GAs), an algorithm with proven success in solving global problems. GA's are combinatorial algorithms for search technique in solving optimisation problems based on the mechanics of the survival of the fittest, which have been developed from analogies of the works of Charles Darwin and his theories of natural selection and preservation of favoured race in the life struggle. Generally, a GA starts with some valid solutions generated randomly, then makes a random change to them and accepts the ones whose fitness functions reduced, and the process is repeated until no changes for fitness function reduction can be made. The disadvantages of using such evolutionary techniques are: evolution training can be slow for complex process plans and such techniques are computationally expensive.

When problems have multi-objectives, such as minimum tardiness, minimal tool changes and best working practice, it becomes difficult to assign the correct weights to the GA's fitness function. To assist in this problem, Ding et al. (2005) used a GA to find optimal sequence plans for machining, and applied ANN to allocate relative weights for different evaluation factors of variant components for process sequencing.

The hybrid approach is also employed for the pre-task of CAPP, such as feature recognition. Orturk and Ozturk (2004) employed a hybrid approach of ANN and Genetic algorithm. The GA is used to find the optimal combination of inputs to be presented to the ANN, thus reducing computational time via the requirement of less neurons in the network. Zhang et al.,

(2006) used a micro GA, in combination with an ANN to obtain the optimal variables for sequence design of cold extruded parts. Hybrid CAPP combines the advantages of neural network and other techniques, e.g. expert systems, fuzzy logic and genetic algorithm (c.f. table 5), and therefore make CAPP more effectively and more adaptively.

Table 5. Hybrid ANN approaches to CAPP

Authors	**System architecture**	***Function***	***Function of hybrid system***
Amatitik and Killic (2007)	Rule-based, fuzzy logic and FF ANN	Process planning, including, machine selection and cutting parameters	ANN for machine tool selection, recognition of STEP AP224 XML files (requires no complex feature recognition) The output is a STEP NC file.
Chang and Chang (2000)	Expert system, FL-BPN	Process planning, including process, machine selection. Part setup determination, cutting tool selection and machine parameters	ANN for evaluation manufacturing ability of important features of components. FL-BPN for evaluating the suitability of the existing plans.
Ding et al (2005)	GA, ANN	Machining operations for components with regular machining feature	ANN used to adapt the relative weights for the evaluating factors for process sequence
Kandel and Langholz (1992)	Expert system, ANN	Process planning	Appropriate manufacturing strategy selection
Ming and Mak (1999)	Expert system, ANN	Process planning	Judging the initial condition and selecting the appropriate control strategy.
Ming and Mak (2000b)	ANN and GA	Optimal process planning	ANN for feature recognition and GA for optimization
Medsker and Liebowitz(1994)	Expert system, ANN	Process planning	Appropriate manufacturing strategy selection
Orturk and Ozturk (2004)	GA, ANN	Machining feature recognition for machined components	Reduced computational time in recognition
Zhang et al (2006)	GA, ANN	Optimal selection of feasible forming sequences of cold extrusion parts	To obtain optimal variables corresponding to the optimal target

5. DISCUSSION

When readdressing the objective of this work, i.e. optimal automated process plans for die-casting dies the initial factor noticed is the lack of research into this specific area when it comes to the feature recognition. The following discusses the findings of this review.

5.1 Feature recognition

The recognition of features to be machined using ANN can be broadly divided into two sections: the feature input representation and the network architecture.

While reviewing previous research, it has been shown that the core advantages of using ANN for feature recognition lies in the fact, they have the ability to learn. They can recognize and classify features; this negates the need to define all elements of feature when addressing new parts:

- Its efficient knowledge acquisition capability owing to its ability to implicitly derive the rules from sample machining cases presented to the neural network;
- Its capability to generalize beyond the original machining cases to which it is exposed during the training and face intermediate situations with reasonably good accuracy with respect to those proposed during the training; and
- High processing speed once the neural network is trained.

The most widely used network architectures are, the three and four layered FF networks. The most common stated reason for this was their proven track record in pattern recognition and in previous feature recognition research. Although the three is most popular, four layered is preferred in the situation when single hidden layer is unlikely to be optimal in terms of learning time or implementation effort.

A disadvantage of ANN's-based approaches which has lead previous researchers to investigate non ANN recognition solutions (Babic et al., 2008), comes form the fact that the recognition process is perceived as not transparent. This has lead to ANN approaches being called "black box processes", where an input is forwarded and an output response received without the ability to observe processing. An active area of research, aims to cure this issue (Johannet et al., 2007; Oussar and Dreyfus, 2001). The search for a 'transparent', or even 'grey' boxes has the potential to show users the available knowledge as the network functions. To promote this, the main technique is that of rule extraction. Here algorithms are employed which mimic the behaviour of the ANN, but allow a comprehensible description of the recognition process. A comprehensive review of such algorithms can be found in

Huysmans et al (2006) and Jacobsson (2005). Although the issues of related to knowledge extraction in ANNs has been raised. It must also be stated that in the author's opinion such techniques are not particularly suited to this particular problem, and the emphasis such be placed on the feature input presentation.

Another disadvantage of ANN approaches comes from the need for pre and post processing of data, which can add considerable time to the overall process. As presented in section three many of the techniques employed for pre-process of the input data are limited to primitive machined features, rendering them unsuitable for complex die manufacturing processes. The approach by Ding and Yue (2004), has potential as it offers recognition of 3D features that interact, also, some researchers have investigated the process of non-standard feature recognition such b-spline curves and surfaces and ellipses (Öztürk and Öztürk., 2001; 2004) which are more applicable to die cavity construction. The latest research has intended to use standard features (i.e. STEP AP224) and standard markup language (i.e. XML), such as Amaitik and Engin Killic (2007) who employed STEP AP244 XML data files of the features in their process planning research. However, previous research (Ding, 2003) into this area found there are still some limitations of feature definition and classification in AP224, including the definition of machining features is not precise (e.g. STEP defines machining features as a volume of removal material while protrusion features, which are not removed volumes, are included); the classification is incomplete and does not include all primitive machining features. Once the standard has been improved to cover all features, this approach will negate the need for complex feature recognition.

5.2 Computer Aided Process Planning

The key issue to identify optimal process plan, is optimizing process sequencing. Although for the die manufacturing process, research that is limited to primitive features can be disregarded from this point forward. In reviewing the published research it can be seen that the use of ANN techniques can improve the performance of CAPP systems, such as operation selection, generation of cutting conditions. These allow empirical rules to be learnt through typical examples. Faster processing makes systems more effective, especially in parallel environments (Ding and Yue, 2004). Although some work has been performed in process sequencing, the results on general application are constrained as the limitations of neural network structure and representation. The only effective constrained optimization of process plans has been performed using the Hopfield network. As noted in section 1, die machining has its own characteristics; the finishing of the die cavity is normally performed via pre-made electrodes on electro discharge machines (EDM). The area where the advancements to be made lie in the blocking and profiling of the die/ insert, roughing of the cavity, machining of holes for core and ejector pins and the relevant slots and pockets relating to overflows, core block slots and feeds. Thus ANN-based CAPP has the potential to be used for die manufacture, although no previous work has been performed specifically for die work. Furthermore, using the learning and recognition strength of ANN in combination with expert systems, and the proven strengths of evolutionary algorithms such as genetic algorithm seem to offer a more robust solution for the manufacturing environment. From the reviewed literature it can be seen that the main advantages of ANN-based hybrid approach are:

- It enables a CAPP system to have self-learning ability;
- It overcomes the problem of approaches which solely rely on ANN, specifically getting trapped in local minima;
- It enhances the adaptability and consistency of a CAPP system, this is a must in a dynamic manufacturing environment;
- It provides suitable tools to deal with uncertainty problems and utilise expert experiences. Thus, the intelligent functions in CAPP are formed efficiently.

6. CONCLUSIONS AND FUTURE DIRECTIONS

This paper has provided an extensive literature review on the application of ANN's to the problem of feature recognition and process planning of casting dies. Putting the reviewed limitations aside, ANN approaches still offer considerable potential in the recognition of features of die-casting dies. As a constrained optimization tool for CAPP, their effectiveness is limited to Hopfield networks and its variants. Although, dedicated optimization algorithms in combination with ANN (hybrid approaches) offer the some of the best solutions to the overall problem when considering, cutting and machine tool parameter investigation and optimal sequence planning.

In this review it has been shown that the mode of initial representation is expanding, not only including drawings, such as Afzal and Meeran (2006). Also, with the current global trend in collaborative and distributed, design and manufacturing, there has been a large volume of research in lightweight representations as the means of information transfer (Fuh et al., 2005, Li et al., 2004 and Ding et al 2007). Most lightweight representations, such as U3D (ECMA-363, 2007), do not use either B-rep or CSG. Instead, they employ facet representations to reduce the file size, but do so at the cost of losing the geometry identifiers within the CAD models. It is the next step to investigate the feature recognition abilities of established techniques on these kinds of representations.

7. ACKNOWLEDGMENTS

The work reported in this paper has been supported by a number of grants for Engineering and Physical Sciences Research Council (EPSRC), involving a large number of industrial collaborators. In particular, current research is being undertaken as part of the EPSRC Innovative Manufacturing Research Centre at the University of Bath (reference GR/R67507/01). The authors gratefully express their thanks for the advice and support of all concerned.

8. REFERENCES

Ahmad, N and Haque, A (2002) artificial neural network based process selection for cylindrical surface machining. Proceedings of the International conference on manufacturing ICM2002, August 2002, Dhaka, Bangledesh. 321-326.

Afzal, M. T and Meeran,S (2006) Recognizing features from orthographical images using neural networks: a framework for CADCAM and AVI integration. International journal of Production Research. 44(22), 4959-4986.

Akyol, D. E and Bayhan, G. M (2007) A review on evolution of production scheduling with neural networks, Journal of Computers and Industrial Engineering. 53, 95-122.

Alam, MR, Lee, KS, Rahman, M and Zhang, YF (2003) Process planning optimization for the manufacture of injection moulds using a genetic algorithm. International journal of computer integrated manufacturing. 16(3), 181-191.

Allada, V and Anand, S (1996) Machine understanding of manufacturing features. International Journal of Production Research. 34(7), 228-243.

Amaititik, S. M and Engin Killic , S (2007) An intelligent process planning system for preismatic parts using STEP feature. International Journal of Advanced Manufacture. 32, 978-993.

Babic B., Nesic, N., Miljkovic Z (2008) A review of automated feature recognition with rule-based pattern recognition, Computers in Industries, 59(4): 321-337.

Balic, J and Korosec (2002) Intelligent tool path generation for milling of free surface using neural networks. Journal of Machine Tools and Manufacture. 42, 1171-1179.

Braid, I.C. Designing with Volumes (2nd ed.). Cantab Press, 97 Hurst Park Ave, Cambridge, U.K., 1974.

Burh, R. O (2003) Feature recognition in 3D surface models using self organizing maps. PhD Thesis. Rand Afrikaans University. South Africa.

CAMWORKS™, http://www.teksoft.com/camworks/camworks.htm (accessed April 2008).

Carpenter, G. A and Grossberg, S. (1987) ART2: self organization of stable category recognition codes for analogue input patterns. Applied optics. 26, 4919.

Chakraborty, S and Basu, A (2006) Retrieval of machining information from feature patterns using artificial neural networks. 27, 781-787

Chang, C.A and Angkasith, V (2001) Using Hopfield neural networks for operational sequencing for prismatic parts on NC machines. Engineering Applications of Artificial Intelligence 14, 357–368

Chang, P.T. and Chang C.H.(2000)An integrated artificial intelligent computer aided process planning system, International Journal of Computer Integrated Manufacturing, 13(6), 483-497.

Chuang, J.H., Wang, P.H. and Wu, M.C. (1999), Automatic classification of block-shaped parts based on their 2D projections, Computers & Industrial Engineering, 36(3), 697-718.

Chambers, M and Mount-Campbell, C.A (2002) Process optimization via neural network metamodeling. International Journal of Production Economics 79 , 93-100.

Chen, Y.H. and Lee, H.M., (1998) A neural network system for 2D feature recognition, International Journal of Computer Integrated Manufacturing, 11(2), 111-117

Chen, C.L.P. and LeClair, S.R., (1993) Unsupervised neural learning algorithm for setup generation in process planning, Proceedings of International Conference on Artificial Neural Networks in Engineering, 663-668.

Chen, H., Wang, T and Sun, B (2005) Application of intelligent CAPP systems based on the ANN. Micro-computer Information. 25.

Chester, D (1990) Why two hidden layers are better than one. Proceedings of the International Joint Conference on Artificial Neural Networks. San Diego, June 17-21, 265-268.

De Floriani L. and Bruzzone E. (1989) Building a feature-based object description from a boundary model", Computer Aided Design, 21(10), 602-610.

Deb, S., Ghosh, K and Deb, S. R. (2001) Machining process planning for rotational components by using a neural network approach. Proceedings of the 29th International conference on Computer and Industrial engineering. Montreal, Canada. 117-122.

Deb, S., Ghosh, K and Paul, S (2006) A neural network base methodology for machining operations selection in computer aided process planning for rotationally symmetrical parts. Journal of Intelligent Manufacture. 17, 557-569.

DellCam Powermill 6 http://www.powermill.com/ (accesses April 2008).

Devireddy, C.R. and Ghosh, K. (1999), Feature-based modelling and neural networks-based CAPP for integrated manufacturing, International Journal of Computer Integrated Manufacturing, 12(1), 61-74.

Devireddy, C.R., Eid, T and Ghosh, K (2002) Computer-aided process planning for rotational components using artificial neural networks. International journal of Agile Manufacturing. 5(1), 27-49.

Darwin, C. The origins of species by means of natural selection or preservation of favoured races in the struggle for life. The book league of America 1929. USA

Ding, L. and Yue, Y., (2002) A Novel Input Representation for ANN-Based Feature Recognition, Frontiers in Artificial Intelligence and Applications, 82, Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies, KES 2002, Part I, 311-315.

Ding, L., 2003, Feature technology and its applications in computer integrated manufacturing. Ph.D Thesis. Luton University, UK.

Ding, L and Yue, Y (2004) Novel ANN-based feature recognition incorporating design by features Computers in Industry, 55(2), 197-222.

Ding, L., Yue, Y., Ahmet, K., Jackson, M and Parkin, R. (2005) Global optimization of a feature-based sequence using GA and ANN techniques. International Journal of Production Research, 2005, 15(1) 3247-3272.

Ding, L., Ball, A., Matthews, J., McMahon, C.A and Patel, M. Product representation in lightweight format for product lifecycle management (PLM) Proceeding of Digital Enterprise Technologies (DET2007). September 2007, Bath, United Kingdom. 87-95. Eds P Maropoulos and S Newman, 637-645. ISBN 978-0-86197-141-1.

Donaldson I.A. and Corney J.R. (1993) Rule-based feature recognition for 2.5D machined components. International Journal of Computer Integrated Manufacturing, 6(1-2), 51-64.

ECMA-363 (2007). Universal 3D File Format. 4th ed. url: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-363%204th%20Edition.pdf (2008-02-01).

ESPRIT® CAD/CAM Software. DP-Technology corporation. http://www.dptechnology.com/en/products.asp (accessed April 2008)

Fausett, L. Fundamentals of Neural Networks: Architectures, Algorithms and Applications, Prentice Hall International, Inc, 1994.

Fuh J.Y.H and Li, W. D. Advances in collaborative CAD: the-state-of-the art, Computer-Aided Design, 2005, 37(5),571-581.

Gu, Z., Zhang, Y.F. and Nee, A.Y.C., (1997) Identification of important features for machining operations sequence generation", International Journal of Product Research, 35(8), 2285-2307.

Hicks, B.J., Mullineux, G., Matthews, J and Medland, A. J. Towards an integrated CADCAM process for the production of forming shoulders with exact geometry. Proceedings of IMechE part B, Journal of Manufacturing Engineering, 2007, 221(10), 1521-1531.

Hopfield,J.J and Tank, D. W (1985) Neural computation of decision in optimization problems. Biological Cybernetics 52(3), 141-152.

Hwang, J.L.Applying the perceptron to 3D feature recognition, PhD Thesis, Arizona State University, USA, 1991.

Hwang, J.L and Henderson, M.R (1992) Applying the preceptron to three-dimensional feature recognition. Journal of design manufacture, 2(4), 187-198.

Huysmans, J.,Baesens,B and Vanthienen,J. ITER: an algorithm for pre-dictive regression rule extraction. In 8th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2006), pages Springer Verlag, lncs, 2006.

Huysman, J., Beasen, B and Vanthienen, J (2006) Using rule extraction to improve the comprehensibility of predictive models. FETEW Internal research. K.U. Leuven.

Jacobsson, H. Rule extraction from recurrent neural networks: A taxonomy and review. Neural Computation, 17, 1223-1263, 2005.

Joo, J., Park, S and Cho, H. (2001). Adaptive and dynamic process planning using neural networks. International Journal Production Research. 39(13), 2923-2946.

Joshi, S.B. and Chang, T.C. (1988) Graph-based heuristics for recognition of machined features from a 3D solid model, Computer Aided Design, 20(2), 58-66.

Johannet , A., Vayssade, B and Bertin, D (2007) Proceedings of world academy of science, engineering and technology. 24, 162-169.

Jun, Y., Raja, V and Park,S (2001). Geometric feature recognition for reserves engineering using neural networks. International Journal of Manufacturing Technology. 17, 462-470.

Kandel, A, Langholz, G. (1992), Hybrid Architectures for Intelligent Systems, CRC Press, Boca Raton, FL.

Kim Y.S., (1992) Recognition of form features using convex decomposition, Computer Aided Design, 24(9), 461-476.

Knapp, G. M., & Wang, H. P. (1992). Machine fault classification: A neural network approach. International Journal of Production Research, 30(4), 811-823.

Kohonen, T (1987) Adaptive, association and self organization functions in computing. Applied Optics. 26 , 4910

Korosec, M., Balic, J and Kopac, J (2005) Neural network based manufacturability evaluation of free form machining. International Journal of Machine Tools and Manufacturing, 45, 13-20.

Korosec, M and Kopac, J (2006) A method of choosing an optimal tool-path in free form surface machining. Proceeding of the sixth International Symposium on Tools and Methods of Competitive Engineering TMCE , 875-796. I Horvath and J Duhovnik eds..

Kulkarni, S.R., Lugosi, G and Venkatesh, S. S (1998). Learning Pattern Classification—A Survey IEEE Transactions on information theory, 44(6).2178-2205 .

Lam, S.M and Wong, T. N (2000) Recognition of machining features—a hybrid approach, International Journal of Production Research. 38, 4301–4316.

Lankalapalli, K Chatterjee, S and Chang, T. C (1997) Feature recognition using ART2: a self organizing neural network. Journal of Intelligent Manufacturing. 8, 203-214.

Lentz D.H. and Sowerby R. (1993) Feature extraction of concave and convex regions and their intersections, Computer Aided Design, Vol. 25, No. 7, 421-437.

Le Tumelin, C., Garro, O. and Charpentier, P. (1995), “Generating process plans using neural networks”, Proceedings of 2nd International Workshop on Learning in Intelligent Manufacturing Systems, Budapest, Hungry

Little, G., Tuttle, R., Clark, D. E. R and Corney, J. R (1998) Delta volume decomposition for multi-sided components. Computer Aided Design. 30(9) 695-705.

Little, G., Tuttle, R., Clark, D. E. R and Corney, J. R (1999) The Heriot Watt FeatureFinder: CIE97 results, Computer Aided Design, 30(13) 991-996.

Li WD, Ong SK and Nee AYC (2000) recognition of overlapping machining features based on hybrid AI Techniques. Procs IMECHE pt B. 214, 739-744.

Li WD, Ong SK, Fuh JYH, Wong YS, Lu YQ, Nee AYC (2004) Feature-based design in a collaborative and distributed environment. CAD 36(9):775–797.

Li, Y., Mills, B., Moruzzi, J.L. and Rowe, W.B., Grinding wheel selection using a neural network, Proceedings of the 10th National Manufacturing Research Conference, Loughborough, 597-601, 1994.

Looney, C. G (1997) Pattern recognition using neural networks: theory and algorithms for engineers and scientists. Oxford University Press. New York, USA.

Mäntylä, M. (1988) An Introduction to Solid Modeling. Principles of Computer Science. Computer Science Press, Maryland, U.S.A

Mamalis, V., Ganosvsky, S and Guenova, V (1994) System based approach to the development of a CAPP system for automated discrete production. International Journal of Advanced manufacture.

Marquez, M., White, A and Gill, R (2001) A hybrid neural network, feature-based manufacturability analysis of mould reinforced plastic parts. Proceeding of the Institute of Mechanical Engineer Part B: Journal of Maufacturing Engineering, 215, 1065-1079.

Marquez, M., Gill, R and Whit, A (1999) Application of neural networks in feature recognition of mould reinforced plastic parts. 7(2), 115-122.

Matthews, J., Singh, B., Mullineux, G., Medland, (2006). A constraint-based approach to investigate the ‘process flexibility’ of food processing equipment. Journal of Computers and Industrial Engineering. 51(4), 809-820.

McMahon, C and Browne , J (1993) *CADCAM from principles to practice*. Addison-Weasley. Workingham, UK.

McCulloch, W. S and Pitts, W (1943) A logical calculus of the ideas imminent in the nervous activity. Bulleting of mathematical biophysics, 5, 115-133

Medsker, L., Liebowitz, J. 1994. *Design and Development of Expert Systems*, Macmillan, Basingstoke,.

Mei, J., Zhang, H.C. and Oldham, W.J.B. (1995) A neural network approach for datum selection in computer-aided process planning, Computers in Industry, 27(1), 53-64

Meeran, S and Zulkifli, A.H (2002) Recognition of simple and complex interacting non-orthogonal features. Pattern Recognition, 35, 2341-2353

Ming, X.G., Mak, K.L. and Yan, J.Q., (1999) A hybrid intelligent inference model for computer aided process planning", Integrated Manufacturing Systems, 10(6), 343-353.

Ming, X.G. and Mak, K.L (2000a) Intelligent set up planning by neural network based approach, Journal of Intelligent Manufacturing, 11, 311-331.

Ming, X.G., and Mak, K.L (2000b), A hybrid Hopfield Network-Genetic approach to optimal process plan selection, International Journal of Production Research, 38(8), 1823-1838.

Neuframe version 4, Getting started manual (2000), Southampton. Neuscience Intelligent solutions, UK.

Neural Computing Manual, (1991), Neural Ware, Inc, Wiley.

Nezis, K. and Vosniakos, G., “Recognising 2.5D shape features using a neural network and heuristics”, Computer Aided Design, 29(7), 523-439, 1997.

Oussar, Y and Dreyfus, G (2000) How to be a Gray Box: Dynamic Semi-physical Modeling Neural Networks, invited paper, 14, 1161-1172.

Öztürk N. and Öztürk F. (2001). Neural network based no-standard feature recognition to integrate CAD and CAM. Computers in Industry, 45,123-135.

Öztürk N. and Öztürk F. (2004). Hybrid neural network and genetic algorithm based machining feature recognition. Journal of Intelligent Manufacturing. 15, 287-298

Osakada, K. and Yang, G.B., (1991) Neural networks for process planning of cold forging, Annals of the CIRP, Vol. 40, No. 1, 243-246.

Park, M.W., Rho, H.M. and Park, B.T.(1996) Generation of modified cutting condition using neural network for an operation planning system, Annals of the CIRP, 45(1), 475-478

Park, M.W., Park, B.T., Rho, Y.M. and Kim, S.K. (2000), Incremental supervised learning of cutting conditions using the Fuzzy ARTMAP neural network, Annals of the CIRP, 49(1), 375-378.

Peters, T.J., (1992) Encoding mechanical design features for recognition via neural nets. Research in Engineering Design, 4(2). 67-74.

Picton, P (1994) Introduction to neural networks, MacMillan Press London, UK

Prabhakar, S. and Henderson, M.R. (1992) Automatic form-feature recognition using neural-network-based techniques on B-rep of solid models, Computer Aided Design, Vol. 24(7), 381-393.

Principe, J.C., Euliano, N.R. and Lefebvre, W.C. (2000), Neural and adaptive system: Fundamentals through Simulations, John Wiley & Sons, Inc.

Sadaiah, M., Yadav, D. Y ., Mohanram, P. V and Radhakrishnan P(2002) A generative CAPP system for prismatic components, International Journal of Advanced Manufacturing Technology 20 , 709–719.

Santochi, M. and Dini, G., (1996) Use of neural networks in automated selection of technological parameters of cutting tools, Computer Integrated Manufacturing Systems, 9(3), 137-148.

Sakurai, H and Chin, C (1994) Definition and recognition of volume features for process planning, in J Shah, M. Mantyla, D. Nau (eds), Advances in feature based manufacture. Elsevier, 65-80.

Sakakura, M. and Inasaki, I.,(1992) A neural network approach to the Decision-making process for grinding operations, Annals of the CIRP, 41(1),353-356.

Shan, X.H., Nee, A.Y.C. and Poo, A.N., (1992)Integrated application of expert systems and neural networks for machining operation sequencing, Neural Networks in Manufacturing and Robotics, ASME, PED 57, 117-126.

STEP AP224, Sharma R, Gao JX (2002) A progressive design and manufacturing evaluation system incorporating STEP AP224. Computers in Industry 47:155–167.

STEP (ISO 10303-1:1994) Industrial automation systems and integration Product data representation and exchange - Overview and Fundamental Principles, International Standard, ISO TC184/SC4, 1994.

STEP-NC (ISO 14649-1:2001) Industrial automation systems and integration Physical Device Control-Part 1: Overview and Fundamental Principles, Draft International Standard ISO TC184/SC4, 2001.

Trajan-neural network simulator. (2004)User manual. Trajan software limited, UK.

Wang, Z., Wang, N and Chen, Yu (2004). Resource decision in integration of CAPP and production planning control. Jisuanji Jicheng Zhizao Xitong-Computer Integrated Manufacturing Systems, 10 (6), 646-650.

Wang, J and Liu, S (1993)Hopfield Neural Network-based Automatic Recognition for 3-D

Features Proceedings of 1993 International Joint Conference on Neural Networks IJCNN93. Nayoto Japan.

Woo Y. and Sakurai H (2002) Recognition of maximal features by volume decomposition, Computer-Aided Design,34, 195-207.

Wong, T.N and Lam, S.M (2000) Automatic recognition of machining features from computer aided design part models. Proceeding of Institute of Mechanical Engineers. Part B. 214, 515-520.

Wu, M.C. and Jen, S.R., (1994) Global shape modelling and classification of 2D work pieces. International journal of computer Integrated Manufacturing. 7, 261-275.

Wu, M.C. and Jen, S.R., (1996) A neural network approach to the classification of 3D prismatic parts, International Journal of Advanced Manufacturing Technology, 11(5), 325-335.

Yahia, N.B.; Fnaiech, F.; Abid, S.; Sassi, B.H., (2002) Manufacturing process planning application using artificial neural networks, IEEE International Conference on Systems, Man and Cybernetics , 5, 6-9

Yan, H and Qiao, L (2004) A Hopfield neural network algorithm for machining operation sequencing. Seventh International Conference on Progress of Machining Technology. ICPT2004. Suzhou, China. 879-884

Yue, Y., Ding, L., Ahmet, K., Painter, J and Walters, M (2001) Study of neural network techniques for computer integrated manufacture. Engineering Computation. 19(2), 136-157.

Zhang HC and Atling, L (1994) Computerised manufacturing process planning system. Chapman and Hall, London

Zhang, X.Q, Peng, Y.H., Ruan, X.Y and Yamazaki, K (2006) Feature based integrated Intelligent sequence design for cold extrusion. Journal of Materials processing technology, 174 74-81.

Zhao, W., Liu, X., Chi, G and Li, Z (2002) Study on the intelligent system for EDM. Jixie Gongcheng Xuebao/ Chinese Journal of Mechanical Engineering. 12, 70-74.

Zhong, Y-G., Qiu, C-H and Shi, D-Y (2004) Application of neural network methods to process planning in ship pipe machining. Journal of Marine Science and Application. 3(2). 42-45.

Zhou, X., Qui,Y., Grangru, H Huifeng, W and Ruan, X (2007) A feasible approach to the integration of CAD and CAPP. Computer-Aided-Design. 39, 324-338.

Zulkifli, A.H. and Meeran, S. (1999), Feature patterns in recognising non-interacting and interacting primitive, circular and slanting features using a neural network, International Journal of Production Research, 37(13), 3063-3100.