Neural Network Based Selection of Optimal Tool - Path in Free Form Surface Machining

Marjan Korosec, Janez Kopac

Abstract:

The purpose of the presented paper is to show how with the help of artificial Neural Network (NN) the prediction of milling tool-path strategies could be performed in order to determine which milling tool - path strategies or their sequences will yield the best results (i.e. the most appropriate ones) of free form surface machining, in accordance with a selected technological aim. Usually, the machining task could be completed successfully using different tool-path strategies or their sequences. They can all perform the machining task according to the demands but always only one of the all possible applied strategies is optimal in terms of the desired technological goal (surface quality in most cases). In the presented paper, the best possible surface quality of a machined surface was taken as the primary technological aim. Configuration of the applied Neural Network is presented and the whole procedure of determining the optimal tool-path sequence is shown through an example of a light switch mould. Verification of the machined surface quality, in relation to the average mean roughness R_a is also being performed and compared with the NN predicted results.

Keywords: *(NN) neural network, CAD/CAM system, CAPP, Intelligent CAM (ICAM), milling strategy*

1. Problem formulation

Many efforts have been made in order to simplify and make NC programming procedures easier. Nowadays, the trend in CAM systems development is to make different CAM systems capable of recognizing particular features which compose a 3D model of the part and then generate the most important machining procedures and parameters [1] according to geometric shape recognition. Some researchers employ Neural Networks and Genetic Algorithms (GA) at this stage but they all face the problem of recognizing very complex free form surfaces, which are far away from being only the basic geometric shapes, such as a cylinder, a cube, a cone etc. So the problem arises how to present a complex surface configuration of free form model to a Neural Network. NN should be capable of predicting the right or optimal machining strategy in order to achieve a high surface quality. So NN must be somehow acquainted with the complex surface configuration of machined workpieces [2]. One possible solution of this problem is shown in this paper. For machining 3D complex surfaces, it is often not enough to use basic tool-path milling strategies only [3]. Specific combinations, which even change during travelling of the cutting tool across the surface, should often be used. Combining milling strategies, there are no simple relations between machining parameters, because they are changing in time and depend on a particular sequence of milling strategies. Their mutual relations are mostly non-linear. The question also arises as to which milling tool-path strategy will be the most adequate to satisfy the demands according to selected technological aims [4]. Generally, it is possible to make optimization according to these main technological aims:

- best possible surface quality,
- minimum tool wear,
- shortest achieved machining time and,
- minimum machining costs.

Since this problem was initiated by the tool shop industry, which produces tools for car lights equipment, our technological aim was to achieve the best possible surface quality of machined workpieces. Different milling strategies can be applied to machine the same complex surface on a workpiece, but surface quality after each different applied combination of milling strategies will differ a lot. It has been realized that by changing the feed rate and cutting speed only, it is very hard to achieve the best possible surface quality in 3D complex surfaces.

2. Current state-of-the-art

Many researchers and developers of CAM systems try to incorporate some intelligence in their applications in order to improve technological knowledge. Some of them are trying to introduce NN, GA and expert systems in their solutions in order to be able to predict crucial machining parameters. A modified Backpropagation NN is proposed for on-line modelling of the milling system and a modified NN is proposed for the real-time optimal control of the milling system [4]. Also a self-organized Kohonen NN is used for path finding and for feed rate adjustment [5,6].

New approaches tend toward integrating CAD, CAPP (computer added process planning) and CAM system [7]. By integrating those three systems, feature-based technology becomes an important tool. The goal of this integration is to replace a typical procedure of manual process planning (Figure 1a.) with CAPP, which represents a basis for automatic generation of NC machining programme (Figure 1b.) [8,9]

In such integrated system, the so-called feature based design or design by features should be used instead of conventional CAD methods. Design feature understands the properties of the region to be machined, such as the geometric shape, the dimensions, the dimensional and geometrical tolerances, etc. However, such features, being primarily design-oriented, have to be converted into manufacturing features in order to build the interface between the feature-based design and automated process planning. According to the definition, manufacturing fea-

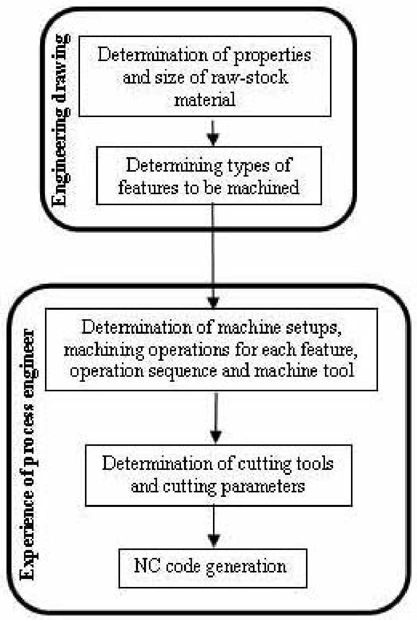

Figure 1a. Manual process planning.

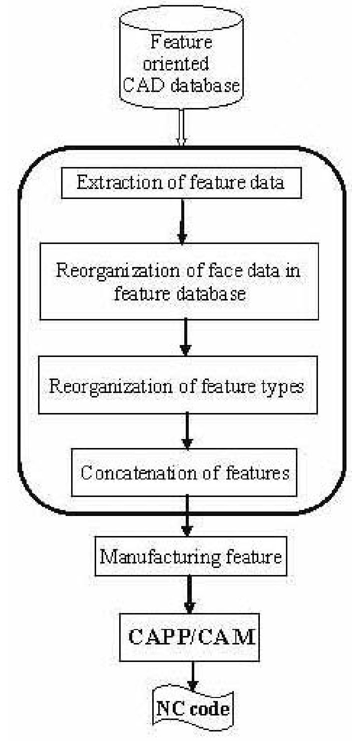

Figure 1b. Extracting of manufacturing features.

tures are surfaces or volumes, which are produced by one or a series of material-removal operations [10]. Without properly defined manufacturing features it is not possible to perform automated NC code generation.

Activities in the area of CAPP and intelligent CAM can be divided into four areas:

- -feature recognition,
- -extracting manufacturing features from a feature-based design model,
- -operation selection as part of CAPP and
- -operation sequencing as a part of CAPP.

Four modules are included in obtaining manufacturing features, presented in Fig. 1b. Feature recognition has been one of the major research issues in the area of automated CAD/CAPP interface. So far, it has also been also the basis for applying intelligent CAM (ICAM) systems. Some main approaches in this area include the Cover Set Graph (CSG)-based approach, the graph-based approach and the neural-network-based approach. Most suggested methods for feature recognition apply a solid model as their input, which represents only the purely geometric aspects of the design information. In 1992, a Super Relation Graph (SRG) system, using artificial Neural Networks, was developed for the purpose of feature recognition. The objective of this system is to recognize and extract prismatic features from 3-D CAD databases [10]. This system recognizes some volumetric primitive features and classifies them into: holes, pockets, blind-slots, blind-steps, thru-slots, and steps. Using the techniques of artificial neural networks and computational geometry, the SRG identifies only features based on two types of relationships between faces: super concavity and face to face. Basically, the SRG is a matrix representation of the relationship between the faces. Later, the SRG based system was improved by adding cover set model (CSM) approach. The CSM is built on the SRG system and determines the essential and non-essential features [11].

In the field of operation sequencing, two methods, i.e. knowledge-based evaluation and fuzzy quantitative evaluation, are widely applied. It means that attributes of a feature are quantitatively fuzzified into a number of measures, which can build up a numeric data array for modelling important features. After that, the fuzzy evaluation function created with neural network can be used to automatically execute feature prioritization. It is clearly that NN based approach is advancing very fast in all CAPP segments, mainly because of the ability of incremental learning and the capability of modelling non-linear inter-relationships. The other advantage of NN based approach is providing a more precise technique and representing the complex inter-connections between the fuzzified feature parameters and the manufacturability of the features [12].

All these approaches are needed to provide the necessary link between CAD and CAM systems and integrate them into the CAPP system. The described systems add technological data to the features from some general data bases, but those added machining data are not optimized for every single feature within the CAD model. Actually, there is no need for feature optimization because of their simple topological and geometrical nature (cylinders, spheres and prismatic features only). But in the case of a complex surface recognition, machining parameters have to be optimized, most effectively by the NN approach [13, 14, 15]. The common limitation of all the above-mentioned methods is that they can recognize and define only volumetric features, based on solid models [16]. These methods can still not recognize features in surface models. When talking about complex free form surfaces, it is not possible to simply divide them into some elementary prismatic or cylindrical features because of their irregular shapes. The other problem with free form surfaces is also their non-linear technological and topological properties relationship, which are impossible to be captured with the above-mentioned methods [17,18]. Considering the fact that nowadays mould design industry uses circa 70% of surface modellers, it is obvious that there is a need for a more general method of manufacturing feature recognition.

The method presented in this paper is different from the above-mentioned approaches especially in the following attributes: it is applicable in solid, as well as in surface 3D free form models, the concept of NN training takes account of all geometrical and topological non-linear relationships, it can easily represent free form surface to the NN, it does not have to be supported from the feature based design concept (therefore it can be used in any CAM system, or in the frame of a widely used CAPP system), and the method is adaptive according to the selected technological goal.

3. Presentation of free form surfaces to the neural network

As mentioned before, many machine-technological parameters depend on the workpiece surface configuration. But for a successful tool-path optimization with the use of NN, the main problem remains: how to present the surface configuration to the Neural Network. So the workpiece surface configuration must first be recognized by NN.

3.1 Selection of representative 3D models and their corresponding milling-path strategies

According to the programming of machining with CAM system (Hypermill-Open Mind), we created five representative 3D models shown in Figure 2. They are the most frequently used tool path strategies and surface forms in our tool-shop company. These selected milling strategies proved to be the best for selected 3D models according to the checked surface quality (mean roughness Ra). For machining material, we used 54 HRc steel. The selected tool path strategies for models presented in Figure 2 are:

- Combination of Profile finishing and Z finishing (slope mode option) representing model NN1. First, flat surfaces are machined in the "profile finishing" mode - surfaces which have the slope angle smaller than the boundary set angle - and then the rest of the surface is machined in the "Z finishing" mode.
- Profile finish, or 3D finish, representing NN2 model.
- Profile finish (scallop height mode), representing NN3 model.
- Z level finish, representing NN4 model.
- Profile finish, (equidistant machining, in feed is constant on the whole surface area), representing NN5 model.

Shapes of the shown 3D models were machined with the proposed milling strategies and in this way the best surface quality results were achieved.

3.2 Multiplication of basic 3D models

In order to get enough training data for NN learning data-base and in order to make the application more universal, a C++ executable programme named Saturnus.exe was written. Among other tasks, it also rotates every basic model in increments by 10° (degrees), starting from -50° to +50°. After the first ten rotations, the basic model was turned upside down by 180° degrees and rotated again by 10 degrees increments. The starting, final and incremental angles are arbitrarily chosen by the user as an input in our written programme (the input file organization will be presented later in this paper). In this way, each basic model produced 20 additional sub-models. So each basic model together with the rotated sub-models provided 24 different models, thus 24 NN model vectors. With five basic models, we provided 120 NN model vectors.

3.3 Projection of points in training models

This task was also automatically performed in our C++ executable program. At first, the programme clipped models in the smallest possible rectangular shape, preserving the a/b relations of rectangular sides constant for each model. After that, models were transferred into the so-called "model space", where they were lifted over the rectangular ground plane and strewed with a raster of points, on the upper part of the model. The strewed points must be settled in appropriate raster, which is arbitrary. In our case, it was 1 mm in the X and Y directions. The more points there are, the more precise the interpolation, which is later performed in the same programme. The whole set (raster) of points was directly projected on each of the five basic models. The number of points and their raster must be equal for every model. After the points were projected, the models were removed and only the distributed points remained in the picture. This was done because later, in the appropriate data transfer format, only points were presented, which was actually of our interest. Namely, we used coordinates of those points in order to get information about 3D surface configuration. As a software tool for all described manipulations with models, Mechanical desktop V4.0 was used.

The purpose of the point's projection across the model was to gain up the surface configuration of the models in the Z direction, which has proved to have the biggest influence on the important technological parameters (feed rate and cutting speed).

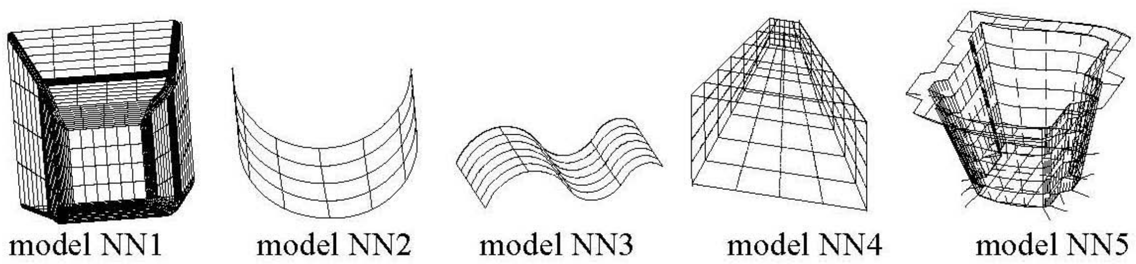

Figure 2. Five models representing basic milling tool-path strategies.

3.4 Preparing VDA files for input into NN

The strewed points were translated with the VDA file format, and then those VDA files are used as an input in our executable programme Saturnus.exe, which performed interpolation between points and their reorganization , and produced ASCII data files, organized in a way to be convenient for entering into NN as input data.

Therefore, our programme performed two very important tasks: it made an interpolation between strewed points so the necessary amounts of data points were reduced (but the Z height configuration of surface was still retained), and it clipped, strewed and rotated the models. With the interpolation, the starting amount of points was reduced to a rectangle having 15 points in the Y direction and 15 points in the X direction for each model.

3.5 Structure of model vectors as an input to the NN

The programme Saturnus.exe produced ASCII files organized as model vectors suitable for direct entering into NN. Each model vector consists of an input and output part (variables).

Concatenation of both vectors gives the original model vector. This can be written as:

$$\mathbf{mv} = \mathbf{P} \oplus \mathbf{Q} = \left(m_1, m_2, \ldots, m_M, m_{M+1}, \ldots, m_L\right) \tag{1}$$

and in a matrix form:

where the shadowed part belongs to the output part of the model vector.

In the presented case, the input part of the model vector consists of 225 (15x15) interpolated coordinate points, and the output part consists of eight probability variables, for one milling strategy each. So there are 120 (5x24) model vectors, each of them consisting of 225 input variables, and one discrete output variable. This presents a learning base for our NN. Training model vectors and their organization are shown in Table 1.

3.6 Structure of output part of model vectors

Looking at Table 1, it is noticeable that the discrete output variable has eight variables of probability. Five of them are used to designate the milling path strategy. Three output variables are left in case of expanding the number of milling path strategies from five to eight. The meaning of the variables of probability is as follows:

(out 1) 10000000....profile finish + Z finish (slope mode option)

(out 2) 01000000....3D finish, respectively profile finish

(out 3) 00100000....profile finish (scallop height mode)

(out 4) 00010000....Z level finish

(out 5) 00001000....profile finish (equidistant machining, constant infeed)

$\mathbf{mv}_1$	=	m_{11}	m_{12}	...	m_{1M}	$m_{1,M+1}$	...	m_{1L}
$\mathbf{mv}_2$	=	m_{21}	m_{22}	...	m_{2M}	$m_{2,M+1}$	...	m_{2L}
...		...						...
...		...						...
$\mathbf{mv}_N$	=	m_{N1}	m_{N2}	...	m_{NM}	$m_{N,M+1}$	...	m_{NL}

Figure 3. Structure of model vectors in a matrix form.

Model vector	Out 1	Out 2	.	Out 5	Out 8	In 1_1	In 1_2	.	.	In 1_15	In 2_1	.	In 2_15	.	In 9_15
Mv1_1	1	0	.	0	0										
Mv1_2	1	0	.	0	0										
Mv1_3	1	0	.	0	0										
.	.	.		.	.										
.	.	.		.	.										
Mv1_24	1	0		0	0										
Mv2_1	0	1	.	0	0										
Mv2_2	0	1	.	0	0										
.	.	.		.	.										
.	.	.		.	.										
Mv2_24	0	1	.	0	0										
.	.	.	.	0	0										
.	.	.	.	0	0										
Mv5_24	0	0	.	1	0										
.	.	.	.	.	.										
Mv8_24	0	0	0	0	1										

Table 1. Organization of model vectors in the NN training data base.

These are five most often used milling strategies for finish machining. Those strategies will not be described in this paper because this is not the intention of this paper. However, their details can be found in almost every CAM system.

4. Neural network setup and algorithm used for probability prediction of milling toolpath strategies

The NN algorithm used for solving the problem of surface recognition is different from the traditional artificial Neural network, in the sense that it is derived from the probabilistic approach and it uses a new self-organizing system algorithm. It is based on a self-organizing system, called the neural network-like system, presented by Grabec [19,20]. It is similar to the method of the nearest neighbour, Learning Vector Quatization Network [21], and also to the probabilistic neural network, proposed by Specht [22]. All of the above-mentioned methods have the same foundation and similar rules for describing various phenomena. On the other hand, they are different compared to each other, similarly to differences among various paradigms used in backpropagation artificial NN, or among various types of artificial NN. Most NN that can learn to generalize effectively from noisy data are similar or identical to statistical methods. For example, probabilistic neural nets are identical to kernel discriminant analysis. On the other hand, Kohonen's self organizing maps have no close relativities in the existing statistical literature, but self-organization of neurons, proposed by Grabec, is very similar to Kohonen's self-organization process and is based on statistical principles [21,23]. Also, feedforward nets are a subset of the class of non-linear regression and discriminant models.

Neural network can learn from cases. It predicts probability in %, which determines milling strategies or their combination that is to yield the best machining results, according to surface roughness. It means that the machining time of predicted strategies is not necessarily the shortest time because our technological aim was to yield the best possible quality of machined surface.

4.1 Definition of probability assumption

However, it is very unlikely that a perfect match exists in reality. Thus, a second probability-based assumption is needed. It states that if the input parts of model vectors P and C are "near", there is a high probability that the output part of C is similar to the output part of P. Conversely, if the input parts of P and C are "far", there is only a low probability as shown in Figure 4. [24].

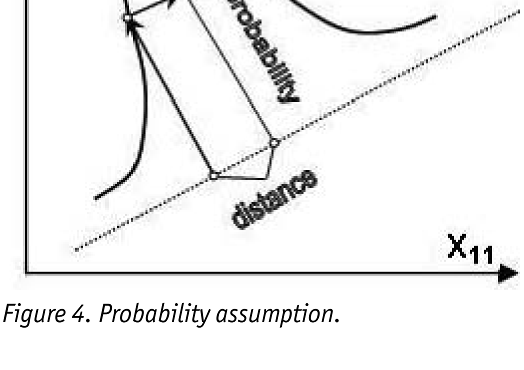

Figure 4. Probability assumption.

The words "near" and "far" from assumption are then converted into numbers. Thus, two vectors are near if the vector norm of their difference is a small value. Usually, an Euclidean norm is used in such cases. Equation (5) shows the Euclidean norm for the difference (distance) of the vectors P and C [25].

$$d_{PC} = \sqrt{\sum_i (x_{Pi} - x_{Ci})^2} \tag{5}$$

where:

d_{PC} is the distance between the input parts of model vectors P and C

x_{pi} is input part of model vector P

x_{ci} is input part of model vector C

When the distances between the model vectors are defined, a Gaussian probability function can be selected. Moreover, if the probability function and the distance are known, the similarity can also be calculated. Thus, the similarity between P and C is represented by [26, 27]:

$$S_{PC} = e^{\frac{-d_{PC}^2}{\alpha}} \tag{6}$$

where:

S_{PC} is the similarity between model vector P and C

α is the penalty coefficient, replacing the standard deviation value

Equation (6) is a slightly modified Gaussian function because the standard deviation cannot be calculated. Therefore, the standard deviation is replaced by a constant value, which is called "the penalty coefficient", and has a significant influence on the shape of the probability function. The penalty coefficient is selected a priori by the user.

For each model vector in M, its distance from P and their similarity can be calculated. To simplify the final calculation of the output part of vector P, the similarity coefficient must be normalized; that is, their sum must equal 1 [26]:

$$\overline{s_{px}} = \frac{s_{px}}{\sum s_{pi}}; \rightarrow \sum_i \overline{s_{pi}} = 1.0 \tag{7}$$

where:

$\overline{s}_{pi}$ normalized similarity coefficient of model vector P

$\overline{s}_{px}$ normalized similarity coefficient of model vector X

Once the similarity coefficients are normalized, the final result is obtained by a combination of the output parts of all model vectors :

$$P_0 = \sum_x \overline{s_{px}} \cdot X_0 \tag{8}$$

where:

P_0 is the final calculation of the output part of model vector P

The index *X* in Equation (8) runs over all model vectors in the model. It should be stressed that the most important thing the user has to do is choosing a penalty coefficient that minimizes the mean square errors from the output variables. The programme uses a method, which prevents "over-training". Namely, it first deletes a case and then uses the remaining cases for training. This "trained" NN is then used to validate this deleted case. This operation is reiterated until all cases have been processed [22, 26].

4.2 Neural network construction

In this particular NN, the training phase is very quick and corresponds to the presentation of the model vectors (loading the database) to the network (*Manual from NeuralWare*, Neural, 1991). The prediction phase corresponds to the calculation of values of processing elements and to the calculation of unknown output values of prediction vector (in case of prediction) or output values of model vectors (in case of filtration of verification to determine the penalty coefficient value). The weights on connections equal either one or zero. The expression for weight adaptation can be written as:

$$w_{ij} = \overline{w}_{ij}\delta_{kj} \tag{9}$$

where $\overline{w}_{ij}$ equals 1.0, and ij is defined as:

$$\delta_{ij} = \begin{cases} 1; i = j \\ 0; i \neq j \end{cases}$$

NN has two hidden layers (layer B and layer C). The number of neurons in layer B equals the product of the number of all model vectors N and the number of input variables M (N.M), while the number of neurons in layer C equals 2 times the number of model vectors. Graphical presentation is shown in Figure 5.

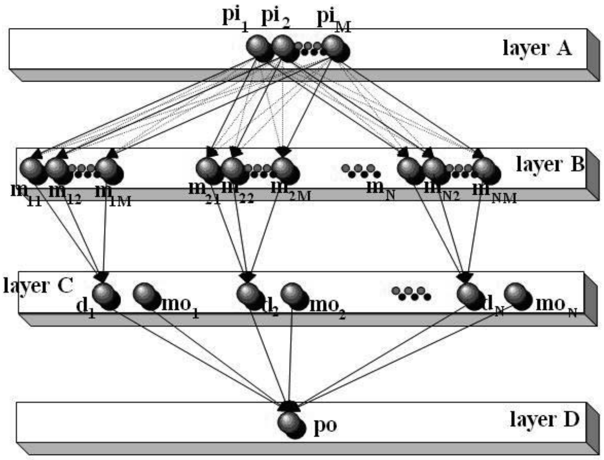

Figure 5. Construction of neural network.

Notations in Figure 5 have the following meanings:

- p prediction vector,
- m model vector,
- i indicates the neuron, belonging to the input variable,
- o indicates the neuron, belonging to the output variable.
- N number of model vectors,
- M number of input variables of the phenomenon,

The processors (neurons) are connected by unidirectional communication channels, which carry numeric data: this can be seen in Figure 5. It should be mentioned that connections (weights) are not changed; they have values either 0 or 1 [22]. All four layers have linear transfer functions and the final output of the neuron on layer D is :

$$po = Y^D = f\left(X^D, \overline{X}^D\right) = \frac{X^D}{\overline{X}^D} \tag{10}$$

where:

X^D and $\overline{X}^D$ mean neuron value before and after weight adjustment.

The presented procedure corresponds to the associative recognition of some unknown properties of the phenomenon on the basis of incomplete observation or experience, obtained by previous complete observation.

5. Training of neural network

The form of model vectors prepared with executable program Saturnus.exe is shown in Table 1. This NN does not learn in a conventional manner but it actually learns simultaneously during the prediction phase. In conventional NNs, a lot of time is spent for training (determination of weights) the NN. But once it has been trained, it predicts quickly. Here it is just the opposite: it predicts a little bit slower, but it learns very quickly. Partially, it happens also because the presented NN uses only two weight values: 0 and 1.

Before running the test set, all vectors had to be normalized and penalty coefficient α must be chosen. The training test was also applied in order to determine the right value of the penalty coefficient. The actual value of the penalty coefficient depends on the density of model vectors. Automatic determination of the penalty coefficient is based on minimizing the RMS verification error. Penalty coefficient is strongly correlated with the learning error in the back propagation neural network (BPNN) and is a very important parameter.

5.1 Results of a training test with known data

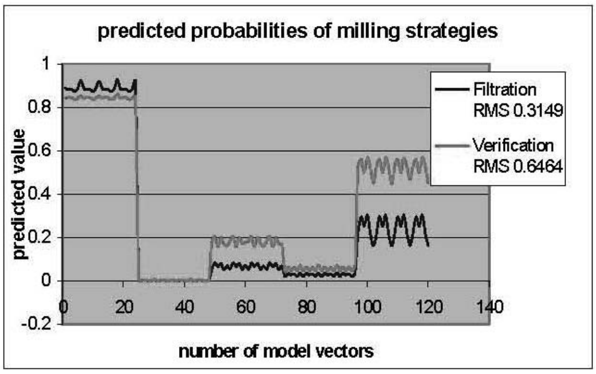

Figure 6. Predicting probabilities of milling strategies in a training set for the first 3D model.

First, data points are normalized. Since the variable values are not falling within particular limits, the statistical normalization was used. Figure 6 shows probability values for the first 3D model (NN1 shown in Figure 2) using data base of all 120 model vectors. It is clearly seen that NN proposes using the first milling path strategy (profile finish + Z finish in slope mode option, as marked in Figure 2) for machining the model since the first 24 model vectors (which represent the first strategy) achieved the highest probability (0.8 to 0.85 in the verification curve in Figure 6) and as such, will give the best surface roughness results. This prediction is quite correct considering the fact that in the learning set, the first 3D model (i.e. the first 24 model vectors) is machined with profile finish + Z finish strategy as the most convenient strategy. In Figure 6, two curves are presented. The verification curve actually excludes the model vectors for which the milling path strategy is predicted. The filtration tool also includes the model vectors for which the prediction is being made. If the filtration and verification curves are getting along fairly well it means that the data noise is small, and vice versa [24,26]. The second proposed milling strategy according to the predicted probability in Figure 6 is equidistant machining with constant infeed over the surface area (verification probability 0.4 to 0.45), and so on. In later experiments with models, which NN hasn't "seen" yet, only their VDA files are needed. The procedure of preparing the VDA files for NN is done automatically, before running the NN.

6. Testing Neural Network model

The proposed NN model is tested on a two experimental 3D models, which have never before been "seen" by NN. The milling path strategies will be predicted with a view to the best possible surface quality. Both 3D models represent the upper part of a mould for plastic injection, and are taken out of the tool-shop practice. The projection of point set was made on a 3D model as shown in Figure 7a and 7b. Points were projected from a rectangular net, lifted over the 3D model as described in chapter 3. In this particular case the spacing between points arrayed on a rectangular raster was 1 mm in the X and 1 mm in the Y direction, because of very steep walls. The spacing between points is picked out arbitrary and depends on a model surface configuration. The results of projecting the points are shown in Figure 7. The points were projected and strewed as described in chapter 3.

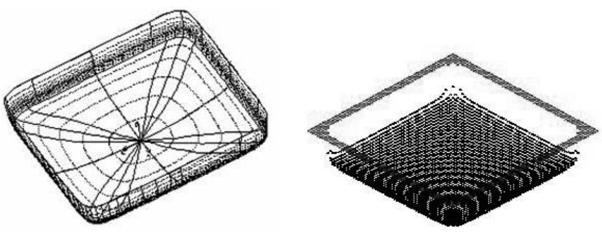

Figure 7a. 3D free form model and point model of a light-switch body.

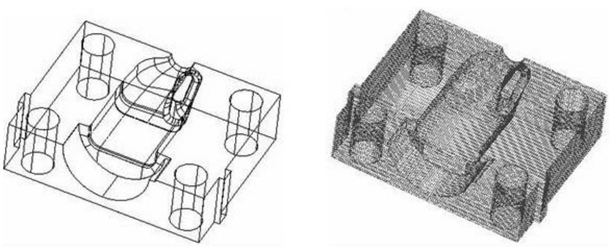

Figure 7b. 3D free form model and point model of a water tank body.

When DWG files of point set were translated into a VDA files, a record of 3832 strewed points (x, y, z coordinates) in a case of light switch body and 18.235 points in a case of water tank came out. Those points describe the 3D surface configuration, which is then imported into NN.

6.1 Results of the test set

When the training model has proved to work well, the milling path strategies for both models from figure 7. are predicted.. It has to be emphasized one more time that these models have never been "seen" before by the NN (they are not included in the training model from Table 1) and therefore represent a really serious proof for NN model.

	milling path strategies				
model vector	out 1	out 2	out 3	out 4	out 5
Mv 1	0.28	0.15	0.07	0.05	0.45
Mv 2	0.25	0.18	0.13	0.05	0.39
Mv 3	0.35	0.15	0.10	0.05	0.35
Mv 4	0.28	0.15	0.10	0.05	0.42
Mv 5	0.13	0.40	0.10	0.05	0.32
Mv 6	0.29	0.15	0.10	0.05	0.41
Mv 7	0.30	0.15	0.10	0.05	0.40
Mv 8	0.15	0.28	0.10	0.05	0.42
Mv 9	0.28	0.17	0.08	0.05	0.42
Mv 10	0.18	0.35	0.10	0.05	0.32
Mv 11	0.28	0.25	0.10	0.05	0.32
Mv 12	0.16	0.27	0.10	0.05	0.42
Mv 13	0.15	0.28	0.10	0.09	0.38
Mv 14	0.28	0.35	0.10	0.05	0.22
Mv 15	0.20	0.19	0.19	0.20	0.22
sum total	**3.56**	**3.47**	**1.57**	**0.94**	**5.46**

Table 2. Predicted probabilities of milling path strategies for model 1.

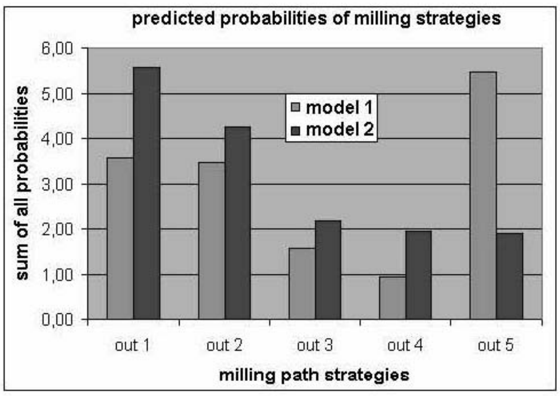

Figure 8. NN results for predicted probabilities of tool path strategies for light switch model and water tank model.

The following finish milling tool-path strategies were used:

out 1......profile finishing + Z finishing (slope mode option)
out 2......3D finishing
out 3......profile finish (scallop height mode)
out 4......Z level finish
out 5......profile finish (equidistant machining, constant infeed)

Looking at Table 2, and observing the prediction for the light switch model, it is very obviously that NN gave the highest probability to strategy number 5 (Mv 1 gave 0.45, and total sum 5.46), that is equidistant machining with constant infeed. The second and the third predicted strategy probability are almost the same (total sum 3.56 and 3.47). Figure 8 shows a graph of predicted probabilities of milling strategies. When observing the prediction for for water tank (model 2), the highest probability was given to the profile finishing + Z finishing with slope mode option (this is strategy "out 1" in Figure 8). The second and third probabilites were 3D finishing ("out 2") and profile finishing with scallop height mode ("out 3"), which were actually used by NC programmers.

In order to get sufficiently satisfactory predicted results, the basic 3D model of a light-switch was rotated around each axis (step of rotation in degrees and axes are arbitrary selected, before running our executable program Saturnus.exe) and strewed with points again. In this way, we got 15 model vectors, and hence the prediction is more reliable. It is important to notice that in most predicted model vectors the strategy with the highest probability is the same, in our example the milling path strategy number 5 (see Table 2). This also implies that the predicted probability might be correct. If the highest predicted probability were divided among different milling path strategies in each model vector the result would probably be doubtful, and the step in the X and Y directions should be changed or the learning model should be redefined. In the presented example this was not the case.

6.2 Checking machined surface quality for NN predicted tool-path strategies and their assessment

Usually, the most frequently used strategy for finish machining of proposed experimental parts used by NC programmers was a conventional 3D finish (strategy in output 3 in Table 2). Sometimes some NC programmers also used a combination of 3D finish with Z level finish machining (strategy in output 1). The part was machined with all 3 milling path strategies, and the centreline average roughness was compared. Finish machining was performed using a ø 2 mm ball mill two flutes cutter, with machine parameters: n = 18.000 min^{-1}, v_f = 1900 mm/min, a_a = 0.04 mm, p_f = 0.2 mm, employing 3+2 axes simultaneously. The parameters were picked out as appropriate for a high-speed-cutting process and a machining workpiece of 54 HRc. A new tool was used for every milling strategy to exclude the influence of tool wear on surface roughness. The results are shown in Figure 9 and Figure 10.

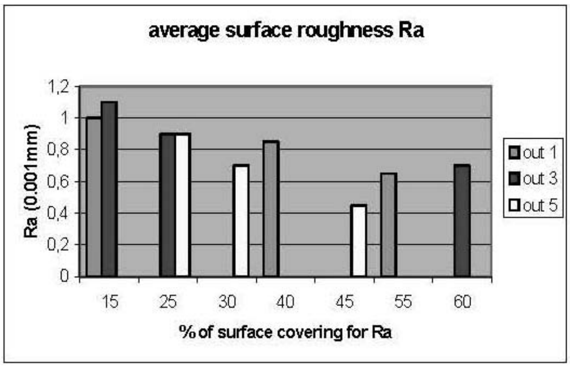

Figure 9. Surface roughness achieved using three milling tool-path strategies for model 1.

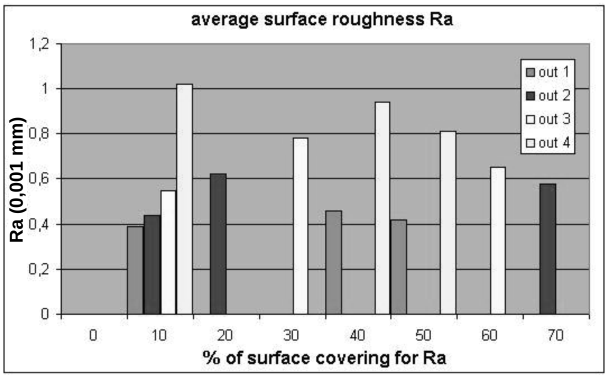

Figure 10. Surface roughness achieved using four milling tool-path strategies for model 2.

In the presented case, NN predicted the probabilities in order of precedence, according to the achieved centreline average roughness R_a on a machined surface. The best R_a at model 1 (light switch) was achieved with the milling strategy "out 5". In this strategy almost 50% percent of machined surface has the R_a value of about 0.45 µm, 30% achieved $R_a \cong 0.7$ µm, and 25% of machined surface achieved the value of $R_a \cong 0.85$ µm. The next two strategies according to the surface quality were milling strategy "out 1" and "out 3". As shown in Figure 9, the machining results are well in agreement with the predicted results from NN in Figure 8. Despite selecting a rather simple machined surface, it is obviously that NC programmers mostly selected milling strategies "out 1" and "out 3", which satisfied the requirements in the tool shop industry, but yielded worse results with a view to surface quality (see Figure 8) than strategy "out 5", proposed by the developed NN. In such a cases, NN can be of great help for NC programmers, operators and technologist in tool-shops, especially in the sphere of fine machining of 3D complex and functional surfaces, where the surface quality plays a major role.

Inspecting the model 2 (water tank) for average surface roughness Ra, it is noticed that the smallest Ra is achieved by using profile + Z level finishing ("out 1") tool-path strategy. By using this strategy, almost 50% of surface has the Ra value of about 43 µm. The second best strategy regarding the achieved Ra is 3D finishing ("out 2"). Comparing results with predicted probabilities in Figure 8, it is clear that actually tool-path strategy "out 1" and "out 2" yielded the highest probability, which is in agreement with surface roughness results from figure 10.

7. Conclusion

A method for optimal choosing and optimizing the milling tool – path strategies based on the use of NN has been presented. The presented method could be used in solid or surface models and can be applied in all modern CAM systems. The surface quality was set as the primary technological aim, and it was focused on it, considering the tool-shop industry. Of course, one may wish to set up a different technological aim, such as achieving the smallest tool wear, or shortening the machining time etc. When changing the technological aim, the learning model should be reorganized according to the new technological aim and NN should be trained again. The more stirring and confabulated machined surface, the more complex and interlaced are the machining parameters, more difficult is to combine the right order of precedence for milling strategies, or to select the most suitable strategies to achieve the best possible machining surface results. In the case of large complex surfaces, reliability is improved when surfaces are divided into technologically and geometrically reasonable subsurfaces before applying the method. It must be also stressed that representative models for NN training phase depends on the type of milling path strategies which will be used inside particular CAM system. So representative models must be choosen in accordance with the capabilities of CAM system which will be used for machining.

After predicting milling path strategies, it is also possible to make a new NN learning model for predicting maximum possible feedrate and rotational speed of spindle.

Input variables in a new NN are: probability of predicted milling path strategies, hardness of workpiece material and stock allowance left during machining. Output variables in a new NN are: maximum possible feedrate and rotational speed of spindle [28]. Predicted feedrate and rotational speed could be then used as the upper limit value for milling path optimization in applications such as OPTIPATH or OPTIMILL (Vericut v. 5.0, CG Tech Ltd).

For solving those problems, NN can serve as an ideal tool for helping NC programmer make the right decision, or at least serving as an orientation tool. The advantage of NN based approach presented in the paper is its ability to learn and recognize all possible complex non-linear topological and geometrical relationships, which cannot be recognized by other graph based or similar techniques. In this way, it is also possible to save time because many additional post machining operations are reduced to a minimum or even zero amount of time.

AUTHORS

Marjan* Korosec, Janez Kopac – University of Ljubljana, Faculty of Mechanical Engineering
Askerceva 6, 1000 Ljubljana; Slovenia.
E-mails: Marjan.Korosec@lecad.uni-lj.si,
Janez.Kopac@fs.uni-lj.si.
*Corresponding author

References

[1] A.C. Lin, S.Y. Lin and. S.B. Cheng, "Extraction of manufacturing features from a feature-based design model", *Int. J. Prod. Res.*, vol. 35, 1997, no. 12, pp. 3249-3288.
[2] G.A. Stark, K.S. Moon, "Modeling surface texture in the peripheral milling process using Neural network", *Journal of Manufacturing science and Engineering*, ASME, ISSN 1087-1357, May 1999.
[3] S.H. Suh, Y.S. Shin, "Neural network modeling for tool path planning of the rough cut in complex pocket milling", *Journal of Manufacturing Systems*, ISSN 0278-6125, 1996, pp. 295-304.
[4] J. Balic, A. Nestler, G. Schulz, "Prediction and optimization of cutting conditions using neural networks and genetic algorithm", *Journal of Mechanical Engineering*, Association of Mechanical Engineers and Technicians of Slovenia, ISSN 0039-2480, 1999, pp. 192-203.
[5] Y.M. Liu, C.J. Wang, "Neural network based adaptive control and optimisation in the milling process", *International Journal of Advanced Manufacturing Technology*, ISSN 0268-3768, vol. 14, no. 11, 1999, pp. 791-795,
[6] M. Korosec , "Optimization of free form surface machining, using neural networks", Doctor thesis, 2003, University of Maribor, Faculty of technical engineering.
[7] Sankha Deb, Kalyan Ghosh; S. Paul, "A neural network based methodology for machining operations selection in Computer-Aided Process Planning for rotationally symmetrical parts", *Journal of Intelligent Manufacturing*, vol. 17, no. 5, October 2006 , pp. 557-569(13).
[8] M. Brezocnik, I. Pahole, J. Balic, "Feature recognition from boundary model of a part" (intelligent CAD-CAP interface), in: *Proc. International Conference Design to Manufacture in Modern Industry*, Bled, Slovenia, 29th-30th May 1995, pp. 395-404.
[9] J. Dong and S. Vijayan, "Feature extraction with the consideration of manufacturing processes", *Int. J. Prod. Res.*, vol. 35, no. 8, 1997, pp. 2135-2155.
[10] T.N. Wong and Wong K.N., "Feature-based design by volumetric machining features", *Int. J. Prod. Res.*, vol. 36, no. 10, 1998, pp. 2839-2862.
[11] K. A. Aldakhilallah and R. Ramesh, "Recognition of minimal feature covers of prismatic objects: A prelude to automated process planning", *Int. J. Prod. Res.*, vol. 35, no. 3, 1997, pp. 635-650.
[12] C. Bishop, *Neural networks for pattern recognition*, Oxford Press, 1995.
[13] I. Grabec, "Optimization of kernel-type density estimator by the principle ofmaximal self-consistency", *Neural Parallel & Scientific Computations*, no. 1, 1993, pp. 83-92.
[14] S. G. Wang, Y. L. Hsu , "One-pass milling machining parameter optimization to achieve mirror surface roughness". In: *Proceedings of the I MECH E Part B Journal of Engineering Manufacture*, vol. 219, no. 1, 2005, pp. 177-181(5) .
[15] J. Wang, "Multiple-objective optimisation of machining operations based on neural networks", *The Int. J. of Advanced manufacturing technology*, Springer London, vol. 8, no. 4, July 1993, pp. 235-243.
[16] C.K. Mok and F.S.Y. Wong, "Automatic feature recognition for plastic injection moulded part design", *The International Journal of Advanced Manufacturing Technology*, Springer: London, vol.34, no. 5-6, September 2007, pp. 1058-1070.
[17] G. Jung Hyun Han, Inho Han, Eunseok Lee, June-ho Yi, "Manufacturing feature recognition toward integration with process planning systems", *Man and Cybernetics. Part B*, *IEEE Transactions on*, vol. 31, issue 3, June 2001, pp. 373 – 380.
[18] Helen L. Locket, Marin D. Guenov, "Graph based feature recognition for injection moulding based on a mid-surface approach", *Computer –Aided Design*, vol. 37, issue 2, 2005, pp. 251-262.
[19] I. Grabec, "Self-Organization of Neurons Described by the Maximum Entropy Principle", *Biol. Cybern.*, no. 63, 1990a, pp. 403-409.
[20] I. Grabec, "Modeling of Natural Phenomena by a Self-Organizing System", *Proc. of the ECPD NEURO COMPUTING*, vol. 1, no. 1, 1990b, pp. 142-150.
[21] D. F. Specht , Probabilistic Neural Networks for Classification, Mapping or Associative Memory, ICNN-88, Conference Proc., 1988, pp.525-532.
[22] C. Principe, R. Euliano, *Neural and adaptive systems*, John Wiley&Sons, 2000.
[23] T. Kohonen *et al.*, "Statistical Pattern Recognition with Neural Networks: Benchmark Studies". In:

Proceedings of the 2nd Annual IEEE International Conference on Neural Networks, 1988, vol. 1.

[24] J. Guh *et al*, "Predicting equilibrated postdialysis BUN an artificial neural network in high-efficiency hemodialysis", *Am. J. Kidney Dis* , no. 31 (4), April 1998, pp. 638-46.

[25] S. Prabhu, "Automatic extraction of manufacturable features from CADD models using syntactic pattern recognition techniques", *International Journal of Production Research*, vol. 37, issue 6, 1999, pp. 1259-1281.

[26] Neural Ware, Inc., *Neural Computing Manual*, Wiley, 1991.

[27] T.C. Li, Y.S. Tarng, M.C. Chen, "A self-organising Neural network for chatter identification in milling", *International Journal of Computer Applications in Technology*, ISSN 0952-8091, vol.9, no.5-6, 1996, pp. 239-248.

[28] M. Korosec, J. Balic, J. Kopac, "Neural network based manufacturability evaluation of free form machining", *I. Journal of Machine Tools & Manufacture*, no.45, 2005, pp. 13-20.