Optimization of the STEP-NC compliant online toolpath generation for T-spline surfaces using convolutional neural network and random forest classifier

O Zavalnyi\textsuperscript{1}, G Zhao\textsuperscript{1,2}, Y Liu\textsuperscript{1} and W Xiao\textsuperscript{1,2}

\textsuperscript{1}School of Mechanical Engineering & Automation, Beihang University, China
\textsuperscript{2}MIIT Key Laboratory of Aeronautics Intelligent Manufacturing, Beihang University, China

E-mail: xiaowenlei@buaa.edu.cn

Abstract. Deep implementation of principals of the intelligent STEP-NC compliant manufacturing implies enabling a CNC system with integrated CAM functionality to perform all tasks autonomously. This means that most of the responsibilities of a CAM engineer need to be delegated to the CNC system. When it comes to manufacturing of freeform surfaces, the implementation of this approach becomes very challenging. One of the most important issues is the autonomous online toolpath generation. Making a choice of optimal machining strategies and other manufacturing parameters, that might be a trivial task for an expert, turns out to be not an easy problem for a machine, and often can be hardly solved using the traditional approach of explicit programming. Therefore, this paper proposes a method to optimize toolpath generation for T-spline surfaces, in particular, the process of choosing an optimal machining strategy for a given surface region using Machine Learning. The selection model has been trained to make a choice of the appropriate freeform machining strategy (two different strategies have been considered) based on the shape of an input surface.

1. Introduction
Feature oriented data representation provided by STEP-CN and the possibility of bidirectional dataflow allow to build CNC systems with integrated CAM functionality [1]. Advantages of the comprehensive integration of STEP-CN into manufacturing processes can be fully estimated while developing intelligent STEP-CNC systems. These systems are expected to perform much of their intelligent functionality (including online toolpath generation) autonomously, and the human-machine interactions have to be minimized.

Enabling such systems to machine freeform surfaces represents a significant challenge for developers. Efficient online toolpath generation for complex freeform surfaces is one of the biggest problems that need to be solved in order to build an intelligent CNC. Zhao et al. [2] presented results of the development of a STEP-compliant CNC with T-spline enabled toolpath generation capability. The reason for choosing the T-spline surface representation for integration within the STEP-CNC is its advantages over other freeform surface representations [3]. These advantages ensure more safety and flexibility during the online toolpath generation processes performed inside a CNC system, and therefore make T-spline a good choice for the development of the intelligent STEP-NC-based manufacturing.
Thus, having completed the development of a real T-spline enabled STEP-CNC system that can strategically support online toolpath generation for freeform machining of simple T-spline surfaces using different freeform strategies, this paper proposes a method to select the optimal freeform machining strategy for a surface region based on its shape using Machine Learning. This task has to be solved in order to enable a machine to make a decision about manufacturing parameters (choosing the machining strategy) based on the previous experience, imitating intellectual abilities of a human (an expert). Providing the machine with this ability allows to optimize toolpath generation, and to perform it more autonomously.

To achieve this result, the strategy selection model has been trained on a comparatively small generated training dataset (5100 pictures) of simple freeform surfaces. For each surface, an optimal machining strategy has been assigned by an expert. Two different architectures of the model have been used for training, Convolutional Neural Network (CNN) and Random Forest classifier, and their results have been compared. The developed algorithm allows to select the appropriate machining strategy (its ID) from a list for any given simple freeform surface (the input is a STL file). The input surface is being transformed into a 2D array (200x200), and the trained model can be used to perform the selection.

2. STEP-NC compliant toolpath generation for T-spline surfaces

First toolpath generation and machining results for simple T-spline surfaces were presented in Gan et al. [4]. The introduction of STEP-compliant data models for T-splines [5] made possible defining them in terms of STEP-NC and their further integration within a STEP-CNC system [2].

2.1. T-spline surface definition

T-spline surface representation, first introduced in Sederberg et al. [3], is a modification of the NURBS representation with much more topological flexibility. It has significant advantages over the NURBS such as less control points, gap-free model representation capability and localized refinement operations.

To calculate a point on a T-spline surface in three-dimensional space, the following point-based equation can be used:

\[
S(s, t) = \frac{\sum_{i=1}^{n} P_i B_i(s, t) w_i}{\sum_{i=1}^{n} B_i(s, t) w_i}
\]

(1)

where Pi are the control points, wi are weights, and Bi(s, t) are the T-spline blending functions:

\[
B_i(s, t) = N_i(s) N_i(t)
\]

(2)

B-spline basis functions Ni(s) and Ni(t) are associated with the individual knot vectors in s and t directions of the parametric domain correspondingly. The components of the knot vectors si and ti for each basis function are deduced from the T-mesh, which is a grid that may have T-junctions [3] (simple T-spline surfaces with T-junctions are shown in Figure 1).

2.2. STEP-NC compliant freeform machining strategies

ISO 14649-11 defines the freeform strategy as a machining strategy used for milling operations. There are four freeform strategies presented in the standard: UV strategy, Plane Cutter Contact (PCC) strategy, Plane Cutter Location (PCL) strategy, and Leading Line strategy. All these strategies have been implemented in terms of STEP-NC definitions and integrated within the developed STEP-CNC system capable to perform online toolpath generation in order to machine simple T-spline surfaces.

Unlike any other strategy, UV strategy defines the toolpaths to follow parametric lines in the local (u, v) coordinate system (Figure 1 (a)). According to the definition of the PCC strategy, the cutter contact paths can be generated by intersecting the target surface with a family of parallel planes, as shown in Figure 1(b). The only difference of the PCL strategy is that the intersections are performed with the target surface, offset by the cutter radius.
Figure 1. Toolpath generation results for simple T-spline surfaces in terms of two different freeform machining strategies defined in ISO 14649-11.

3. Scope of the investigation

Considering the complexity of the problem domain and the need to have a sufficient dataset in order to train strategy selection models using Machine Learning, some limitations defining the scope of the investigation have to be imposed.

Thus, we consider only simple (in terms of the shape and topology) freeform surfaces as an input for the strategy selection algorithm. Though most of the surfaces to be machined are not simple, some existing algorithms [6, 7] allow to subdivide any complex surface into a set of feature-based regions. These region-based tool path generation methods are based on dividing the freeform surface into regions by identifying meaningful features, including surface shape and machinability. Moreover, the possibility to use local refinement operations (introduced with T-splines) makes the subdivision process even more efficient. Therefore, the problem of toolpath generation for a complex T-spline surface can be reduced to the problem of toolpath generation for the set of simple surface regions.

The number of strategies available for selection by the trained model has also been limited. In this paper, two main freeform machining strategies presented in the standard have been considered, UV strategy and PCC strategy. These two strategies have one significant difference which makes it very important to choose the one strategy and not the other in each specific case. Under the UV strategy, toolpath lines are defined by iso-parametric lines. Therefore, despite that the distance between two neighboring toolpath lines in the parametric domain is equal, this distance may vary considerably in the 3D space depending on the surface shape and topology. Considering the need to maintain the constant scallop-height between consecutive trajectories on the manufacturing part, the toolpath length (and machining time) may increase. Whereas, toolpaths generated using the PCC strategy do not have this problem because cutter contact paths are defined as the intersection of the target surfaces with parallel planes in 3D space. However, toolpath generation in terms of the UV strategy is simpler and computationally cheaper than it is for the PPC strategy. Thus, the choice of the optimal machining strategy for each specific case has to be made based on the surface region shape.

4. Brief introduction to machine learning

Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed [8]. It allows a computer program to learn from experience and improve its performance with respect to some task. Recently, Machine Learning has been experiencing a never seen increase in the number of its applications in different fields, including manufacturing [9]. Deep learning is part of
a broader family of machine learning methods, and having as its core different architectures of artificial neural networks (ANNs), it allows to solve very complex Machine Learning tasks.

The problem of the machining strategy selection is a classification problem. After training the model on a set of the labelled training data, this model can be used to define the class of any new sample. Many different model architectures could be used to solve this problem, and in this paper Convolutional Neural Network (CNN) and Random Forest classifier have been considered. CNNs have provided state-of-the-art results in many computer vision problems [10], but they usually require a large number of training samples. Random Forest classifiers can also directly classify instances into multiple classes [8], and it is often possible to achieve good accuracy results even on a smaller number of training samples.

Compared to the traditional Artificial Neural Networks (ANNs), CNNs are primarily designed and used in the field of pattern recognition within images [11]. Therefore, some image-specific features, which make the network more suited for image-focused tasks, can be encoded into the architecture, and the parameters required to set up the model can be further reduced. The layers within the CNN are comprised of neurons organized into three dimensions, the spatial dimensionality of the input (height and the width) and the depth (the third dimension of an activation volume). CNNs usually contain three types of layers: convolutional layers, pooling layers and fully-connected layers [12]. The convolutional layers are used to extract features from the input images, then the dimensionality can be reduced by the pooling layers, and the fully connected layers act as classifiers. If a sufficient training dataset is available, CNNs almost always will provide the best results in pattern recognition and image classification tasks.

Random Forests classifiers are classifiers based on decision-tree that use ensemble algorithms (algorithms which combine more than one algorithms of same or different kind for classifying objects) [13]. They operate by constructing a number of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. The training process is generally carried out via the bagging method [8]. Random Forests are well suited for both classification and regression tasks, and therefore may be used in various machine learning applications.

5. Preparation of the dataset for training

Proposed strategy selection algorithm takes as an input a surface given in STL format (often used format in CAM applications). The integrated within the developed STEP-CNC system T-spline kernel supports STL format for the data exchange, therefore the algorithm can be easily implemented inside the system architecture in the future. The geometrical information contained in the STL files cannot be directly used in the training processes, because different surfaces may be represented by the different number of triangles. The input for the most of models has to be a fixed-size sequence of features (a fixed number of input neurons for CCNs). Therefore, in order to train strategy selection models, the original surface has to be represented as an array of fixed size. The best way to do this is to transform the surface into a voxelized 3D model located within a grid box of fixed size. Then the dimensionality of this box can be reduced, and the model can be represented as a 2D matrix. This approach is possible due to the assumption that the input is a simple surface, and therefore it can be easily mapped on the XY-plane of the local coordinate system. By subdividing a complex surface into a set of simple surface regions, we can fulfill these requirements in the most cases. Otherwise, the 3D matrix representation can be used, and 3D convolutional neural networks have to be employed correspondently.

As an example, the transformation of a grid box (15x15x15) with one single voxel into a 2D matrix (15x15) is presented in Figure 2. X and Y-coordinates of the voxel in the box define the row and the column numbers in the matrix (12 and 8 correspondently). Z-coordinate of the voxel, 11, defines the normalized value, \( Z_n \), of the corresponding element of the matrix (0 ≤ \( Z_n \) ≤1): \( Z_n = (1/15) \times 11 = 0.73 \). The rest of the elements of the matrix have zero values.

In order to be used for training, each machining strategy is assigned a unique ID number (0 – for UV strategy and 1 – for PCC strategy). These ID numbers can be used to label training samples.
First, 340 unique simple T-spline surfaces of different size and shape have been created and saved in the STL format (sample surfaces are shown in Figure 3 (right)).

![Figure 2. Transformation of a grid box with one voxel into a 2D matrix.](image)

For each surface, the optimal machining strategy has been assigned by an expert, and each sample was labelled with the corresponding strategy ID number. Then, the surfaces have to be transformed into the 2D matrix representations, as described in this section (see Figure 3 (left)). A grid box (200x200x200) with the cell size in all three dimensions equal to 0.5 mm is created, and the surface, being voxelized, is centered in the box. All surfaces must be scaled to fit into the box. At the same time, the surface is transformed into a 2D matrix representation (see Figure 3 (left)).

![Figure 3. Transformation of the surface representation (left). Examples of different surfaces and prepared training samples (right).](image)
time, augmentation (by rotating the surface about the X, Y, and Z –axis by a small random angles) is
performed to expand the number of training samples. To manipulate the surface transformations, open
source package “trimesh” written in Python has been used. Finally, each sample is represented as a 2D
data matrix (200x200), which can be viewed as a fixed-size image (see Figure 3). Thus, the final training
data set after augmentation contains 5100 images of the size (200x200) and 5100 labels which define
the corresponding strategy IDs.

6. Training of the strategy selection models and the test results
To perform the training, Tensorflow library has been used. Different architectures and combinations of
hyper-parameters have been tested to get good results.

The CNN-based training model is a 5-layer neural network with 3 convolution layers, 200x200x1
input layer and the output layer of size 2 (two classes of the machining strategy). The model has the
following architecture: one channel input layer, convolution layer of output depth 4, convolution layer
of output depth 8, convolution layer of output depth 16, fully connected layer, output layer (one-hot
encoded labels). The (2 × 2) kernel has been. The learning rate was set to 0.003, the activation
function was a ReLU, and the Adam optimizer has been used during the training process.

To train the model based on the Random Forest classifier, the Tensorflow contributed library has
been used. Its model has the following structure: the input is a vector of 40000 features (from the
200x200 input matrix), the output has 2 classes (2 strategies) and number of trees is 35 (with limited to
maximum 400 nodes). However, different numbers of trees and nodes may give the same results.

To estimate the accuracy of the trained models, several new sets of test samples have been created
and labelled by an expert. The models demonstrated different results for different sets of test data, as
shown in Table 1:

The model based on the Random Forest classifier demonstrated slightly better results. Moreover, as
the training dataset is comparatively small, the CNN-based model is very prone to overfitting and
therefore requires more tuning manipulations.

| Model type          | Accuracy, % |
|---------------------|-------------|
| CNN (5 layers)      | 85 – 94,8   |
| Random Forest classifier | 87,5 – 95.8 |

Sometimes, a surface may have the shape, based on which it is hard to conclude that the one
machining strategy is an exclusive and the only possible choice, and the other is totally wrong
(multiple choice is possible). In this case, the trained model might make a “wrong decision”, but it still
can be an acceptable choice in terms of machining results. Considering this fact, it is concluded that
the achieved accuracy results are satisfactory and potentially may be used for the online toolpath
generation applications. However, to achieve better and more stable results, a larger amount of
training data has to be used.

7. Conclusions
The results of this research demonstrate the real possibility to use Machine Learning for the machining
strategy selection. The trained model, after further optimization (possibly, retraining on larger datasets)
can be integrated inside the STEP-CNC system and be used during the online toolpath generation. As
the future work, authors consider adding more machining strategies available for the selection.
Training modes for the automated selection of cutting tools and some other manufacturing parameters
might be also considered in the future. Presented in this paper results are very important because they
show that Machine Learning can potentially help to transfer the responsibilities to choose any
manufacturing parameters from a human to a machine, as long as a sufficient training datasets are
available.
Acknowledgements
This work was supported by Special Program of Ministry of Industry and Information Technology, China.

References
[1] Newman S T, Allen R D and Rosso Jr R S U 2003 *Int. J. of Computer Integrated Manufacturing* 16 pp 590–597
[2] Zhao G, Liu Y, Xiao W, Zavalnyi O and Zheng L 2018 *Int J Adv Manuf Technol* 94 pp 1799–1810
[3] Sederberg T W, Zheng J, Bakenov A and Nasri A 2003 *ACM Transactions on Graphics* 22 pp 477–484
[4] Gan W F, Fu J Z, Shen H Y, Chen Z Y and Lin Z W 2014 *Computer-Aided Design* 52 pp 51–63
[5] Xiao W, Liu Y, Li R, Wang W, Zheng J and Zhao G 2016 *Computer-Aided Design* 79 pp 36–47
[6] Chen Z C, Dong Z and Vickers G W 2003 *Computers in Industry* 50 pp 319–331
[7] Elber G 1995 *Computer-Aided Design* 27 pp 465–470
[8] Géron A 2017 *Hands-On Machine Learning with Scikit-Learn and TensorFlow* O’Reilly Media Inc. p 541
[9] Wuest T, Weimer D, Irgens C and Thoben K D 2016 *Production & Manufacturing Research* 4 pp 23–45
[10] Keshari R, Vatsa M, Singh R and Noore A 2018 Learning Structure and Strength of CNN Filters for Small Sample Size Training *CoRR* abs/1803.11405
[11] O'Shea K and Nash R 2015 An Introduction to Convolutional Neural Networks *CoRR* abs/1511.08458
[12] Canziani A, Paszke A and Culurciello E 2016 An Analysis of Deep Neural Network Models for Practical Applications *CoRR* abs/1605.07678
[13] Ho T K 1995 Random Decision Forests in *Proceedings of the 3rd International Conference on Document Analysis and Recognition* 1 pp 278–282