



**HAL**  
open science

## Feature Recognition for Virtual Machining

Shixin Xú, Nabil Anwer, Lihong Qiao

► **To cite this version:**

Shixin Xú, Nabil Anwer, Lihong Qiao. Feature Recognition for Virtual Machining. 21st International Conference on Industrial Engineering and Engineering Management 2014, Nov 2014, zhuhai, China. hal-01094274

**HAL Id: hal-01094274**

**<https://hal.science/hal-01094274>**

Submitted on 12 Dec 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Feature Recognition for Virtual Machining

Shixin XU<sup>1,2\*</sup>, Nabil ANWER<sup>1</sup>, Lihong QIAO<sup>2</sup>

<sup>1</sup>LURPA, ENS de Cachan, 94235 France

<sup>2</sup>SMEA, Beihang University, Beijing 100191, China

( \*[sxu@ens-cachan.fr](mailto:sxu@ens-cachan.fr); [anwer@lurpa.ens-cachan.fr](mailto:anwer@lurpa.ens-cachan.fr); [lhqiao@buaa.edu.cn](mailto:lhqiao@buaa.edu.cn) )

**Abstract** – Virtual machining uses software tools to simulate machining processes in virtual environments ahead of actual production. This paper proposes that feature recognition techniques can be applied in the course of virtual machining, such as identifying some process problems, and presenting corresponding correcting advices. By comparing with the original CAD model, form errors of the machining features can be found. And then corrections are suggested to process designers. Two approaches, feature recognition from G-code and feature recognition from In-Process Model (IPM), are proposed and elucidated. Feature recognition from IPM adopts a novel method of curvature based region segmentation and valuated adjacency graph. Recognized machining features are represented in conformance to STEP-NC. Feature recognition can help the virtual machining analysis in revealing potential defections in machining operations.

**Keywords** – feature recognition, virtual machining, in-process model, G-code, STEP-NC

## I. INTRODUCTION

In a CAM environment, using a part's CAD model, after tool path strategies and cutting conditions are settled, the part's NC program can be generated. Then the NC program should be verified or proved in order to lower production risk. NC program proving by trial-and-error cycle on a physical machine is slow and costly. Sometimes, in manufacturing costly aerospace parts, trial-and-error based optimization might be unaffordable. The aim of virtual machining (VM) is to decrease the trials by simulating machining operations in digital environments prior to actual manufacture. VM is performed mainly on the establishment of the mathematical modeling of part-tool engagement geometry, machining process physics, structural and rigid body kinematic motion of the machine and work material properties. The simulation of machining operations in a virtual environment can predict dimensional surface errors left on a part, as well as maximum cutting forces, torque, power, vibration amplitudes in short time [1,2].

We have been trying feature recognition techniques for virtual machining in two ways. One way is to carry out feature recognition on part programs (G-codes in conformance to ISO 6983) to rebuild the machining feature based model of the part. In some cases, the original CAM model could be neither available nor usable. For example, the part's archive is lost/damaged, or incompatible with upgraded systems, or the part is

designed and machined at shopfloors without aids of CAx tools. Therefore the rebuilt model can help understand and check the VM outcome. The other way is to perform feature recognition on the part's in-process model (IPM) resulting from the virtual machining. Then the extracted features can be used to reconstruct solid models and to evaluate form errors for machining and inspection. In both ways, the recognized machining features are expressed as STEP-NC features (ISO 14649 standard) for convenience of subsequent applications. The process diagram is illustrated in Fig. 1.



Fig. 1. Process diagram of feature modeling for virtual machining.

The IPM, which is the geometric output data of the virtual machining, is usually a collection of unordered set of triangles, without topological information. Many NC simulation tools output the IPMs as STL files, like NCSIMUL and VeriCut. In fact, an IPM bears two types of features: surface micro features and machining features (such as holes, pockets, slots, steps). Surface micro features refer to textures, scallops, marks, and veins on machined surfaces of the IPM. Fig. 2 shows examples of surface micro features of a face milling. Surface micro features could be related to some cutting parameters, such as scallop height and stepover. From the patterns of the surface textures, we can determine whether the machining operation is stable, or is under chatter influence. If there are series of marks left on a machined surface, usually at its corners, that indicates jerks, revealing discontinuous feed rates along tool paths. Surface micro feature recognition involves image pattern recognition, etc., and in this paper we do not discuss it. Here we only use feature recognition approach to extract machining features from IPM. Thereafter we can identify potential machining operation problems by analyzing these features.

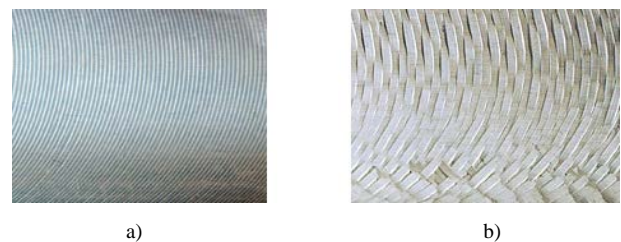


Fig. 2. Examples of surface micro features: a) texture formed by stable milling of a plane; b) chatter marks left by unstable milling.

Common feature recognition approaches take a solid model as their inputs, and search for pattern of faces and edges that obey certain topological or geometrical relationships [3]. But in this work we take a G-code program or the triangle mesh instead as the input of feature recognition. These need to design new methods to treat the G-code or the IPM data.

The remainder of the paper is organized as follows: Section II explains the feature recognition approach from G-code, Section III elaborates the feature recognition approach from IPM. Finally, we draw conclusions in Section IV and give some directions for further research.

## II. FEATURE RECOGNITION FROM G-CODE

### A. Introduction

G-code focuses on programming the cutting tool paths with respect to machine axes, rather than the machining process with respect to the part. Generally there is no information about cutting tools, the rawpiece' shape and dimensions, its location, and the CNC controller type in the G-code, we should first supplement these for ensuring successful recognition. Then by analyzing hints (such as tool changes, speed changes, machining regions) in the G-code, workingsteps can be generated. In these workingsteps all operations are treated as freeform operations and features as toolpaths (which is converted into the data structure "toolpath" of freeform operations.), except those (like canned cycles) that can be easily attached to operations. Finally, we can extract machining features from the toolpath data by analyzing the machining regions, machining strategies, etc. By organizing the extracted feature data by ISO 14649 definitions, the STEP-NC features can be obtained.

### B. From G-Code to STEP-NC

We devised an interpreter for the toolpath generation [4]. The interpreter emulates the execution of the given G-code one block by one block: if it meets a "G0" command, a "rapid movement" entity is created; if it meets one or several consecutive "G1"s or "G2/3"s, a "machining workingstep" which includes a freeform operation is created. The parameters of these commands are used as cutter location data stored in the toolpath list of the freeform operation. Sometimes computation is needed for obtaining the toolpaths.

Next, we deal with the extraction of manufacturing (machining) features from toolpaths and tool's geometry. In this phase, one freeform operation corresponds to one machining workingstep. Many freeform operations might correspond to one manufacturing feature because often there are several layers of a rough machining and a finish machining, which are needed to make a final feature. Hence one major procedure is to merge those freeform

operations that machine the same feature, as well as the relevant workingsteps and rapid movements.

Some features can be easily extracted by the tool used. For example, if the tool type is for drilling, the feature is a hole; if the tool is a facemill, the feature is a planar face. If the tool is an endmill, which is a general and complex case, then analyse the x, y, z-values of the CL (cutter location) data in the toolpath list. If z-value varies and x, y-values keep constant, it mills a hole feature. If x, y, z-values all vary, it is a freeform milling operation for making a region of surface. If z-value keeps constant and x, y-values vary, it is a 2½D milling operation. The main grounds to find the remaining features (which can be planar faces, general outside profiles, closed pockets and open pockets in STEP-NC.) in a 2½D milling operation are the cutting area (the tool's covering region for cutting movements) and the milling strategy, which are computed based on toolpath CL data.

### C. An Example

In the example G-code of the study case part (Fig. 3), 3 cutting tools (a drill, a reamer and an endmill) are used. By the tool types and the canned cycles in the code, we can extract a hole feature. By the cutting area and the milling strategy, we can identify a pocket feature.

TABLE I  
AN EXAMPLE G-CODE

G54 G90 G21 G40 G49 M5 M9	Y100.
G0 Z100. (Move to the secure plane)	X85.
(To drill and ream a thru hole)	Y40.
T2 M6 (Use a spiral drill, diameter 20mm)	X55.
G43 H2 (Length compensation by 70mm)	Y100.
M8 M3 F900. S720	X70.
G0 Z30.	Z0.
G90 G99 G81 X20. Y60. Z-18. R10.	G0 X69.532 Y47.815 (End of 1st layer)
G99 G81 X20. Y60. Z-36. R10. F1800.	..... (Code of next 4 layers omitted)
G99 G81 X20. Y60. Z-60. R10. F1350.	
G1 Z10. F1800.	(To finish pocket in 6 layers. 5mm/layer)
G80 G49 M5 M9 (end of drilling cycle)	(Bottom allowance 0.5, side allowance 1)
T3 M6 (Use a reamer, diameter 22mm)	G0 Z30.
G43 H3 (Length compensation by 50mm)	X74.890 Y60.285
M8 M3 S1080	Z15.
G90 G99 G85 X20. Y60. Z-60. R10.	(First 2 blocks: to run helical approach)
G80 G49 M5 M9 (End of reaming cycle)	G2 X77.2 Y55. Z-2. I-4.891 J-5.285
	G2 X70. Y55. I-3.60 J0.
	G1 Y93.
(To cut a pocket, rough & finish)	X78.
T1 M6 (Use an endmill, diameter 18mm)	Y47.
G43 H1 (Length compensation by 50mm)	X62.
M8 S1200 M3 F2400.	Y93.
G0 Z30.	X70.
X64.754 Y50.069	Y101.
Z15.	X85.
(To rough pocket in 5 layers, 5.9/layer)	G2 X86. Y100. I0. J-1.
(First 2 blocks: to run helical approach)	G1 Y40.
G2 X77.2 Y55. Z-5.9 I5.246 J4.932	G2 X85. Y39. I-1. J0.
G2 X70. Y55. I-3.60 J0.	G1 X55.
G1 Y90.	G2 X54. Y40. I0. J1.
X75.	G1 Y100.
Y50.	G2 X55. Y101. I1. J0.
X65.	G1 X70.
Y90.	Z0. (End of 1st layer)
X70.	.....(Code of rest of layers omitted)
Y95.	G2 X55. Y101. I1. J0. (now Z-30.)
X80.	G1 X70.
Y45.	Z15. (End of finishing)
X60.	M30
Y95.	
X70.	

After the recognition from the G-code, we got two features: a hole and a pocket. Structuring the feature data by the STEP-NC standard, we can get the part21 file for the recognition result, like in the following excerpt.

```

.....
#11= MACHINING_WORKINGSTEP ('WS DRILL HOLE1',#60,#21,#32,$);
#12= MACHINING_WORKINGSTEP ('WS REAM HOLE1',#60,#21,#33,$);
#13= MACHINING_WORKINGSTEP ('WS ROUGH POCKET1',#60,#22,#34,$);
#14= MACHINING_WORKINGSTEP ('WS FINISH POCKET1',#60,#22,#35,$);

#21= ROUND_HOLE ('HOLE1 D=22MM',#2,(#32,#33),#67,#70,#111,$,#25);
#22= CLOSED_POCKET ('POCKET1',#2,(#34,#35),#69,#71(),$,#26,$,#112,#27);
#25= THROUGH_BOTTOM_CONDITION ();
#26= PLANAR_POCKET_BOTTOM_CONDITION ();
#27= RECTANGULAR_CLOSED_PROFILE ($,#113,#114);

#32= DRILLING ($,$,'DRILL HOLE1',15.,$,#44,#54,#51,$,$,$,$,#55);
#33= REAMING ($,$,'REAM HOLE1',15.,$,#47,#54,#51,$,$,$,$,#56.,T.,$,$);
#34= BOTTOM_AND_SIDE_ROUGH_MILLING ($,$,'ROUGH POCKET1',15.,$,#40,
#57,#51,$,$,#58,6.5,5.1,.0.5);
#35= BOTTOM_AND_SIDE_FINISH_MILLING ($,$,'FINISH POCKET1',
15.,$,#40,#57,#51,$,$,$,#59,2.,10.,$,$);
.....

```

In fact, the G-code makes a part shown in Fig. 3.

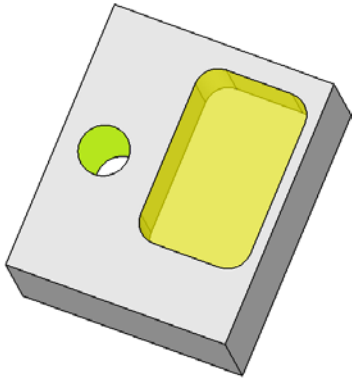


Fig. 3. The part made by the example G-code.

### III. FEATURE RECOGNITION FROM IPM

#### A. Introduction

The widely used mesh-based virtual machining simulation refers to that the IPM uses triangle mesh for processing and dynamic display, and hence the IPM can be output as collection of triangles. Most of commercial NC simulators provide an STL export interface for outputting the IPM data. During machining process simulation, one popular method is that the workpiece and the cutting tool are represented as discretized triangles for simplifying the computation and display. The geometric modeling of the cutter-workpiece engagement along the tool path are very important to the variation in chip thickness, and axial/radial depths of cut that are needed to evaluate force, torque, power, vibration and other process states along the tool path. The IPM mesh data has the following characteristics.

1) The IPM is a collection of triangles, among which have no sequence and topological relationship. The IPM sometimes contains offcuts, which should be identified and eliminated in the beginning of feature recognition. The mesh data usually contains degenerated triangles (i.e. isolated vertices and edges) and incorrectly

oriented triangles, or lacks of triangles data resulting a gap on the surface.

2) Compared with the mesh data of a faceted CAD model, the IPM mesh data is more irregular and complex. For example, a rectangular planar face in a CAD model may be faceted as 4 triangles; but the counterpart in IPM might be like a waved surface composed by numerous triangles after machining. This is due to the scallops left on the face, forming small ridges and ruts, which are represented by tiny triangles in the IPM.

The overall method we designed is as follows: first, build a surface B-rep model by processing the IPM data; then, recognize machining features from the B-rep model.

Two approaches are considered to the feature recognition for virtual machining in our work. One is to attach attributes to triangles while carrying out the virtual machining. This approach is simple while practical when the number of the triangles is small. The other approach is to analyze the pure geometric mesh data of the simulation, on basis of discrete segmentation [5,6]. This is a general approach, not rely on a certain NC simulation tool. In this paper, we will focus on the latter.

#### B. Description of key procedures

##### 1) B-rep triangular model building

We use the Open CASCADE [7] toolkits, an open source geometry development platform, to build the B-Rep triangular model from the IPM data. Each triangle in the STL file is transformed into a triangular face. When completed, a manifold B-Rep triangular solid model is generated. This model has the following characteristics: In ruled surface regions and in planar regions at its curved boundaries, the model has many long and thin triangles resulting from the discretization; Vertices of triangles in planar and ruled regions are mostly on the part's boundary edges; Triangles in flat regions are sparse and relatively large, while dense and uniform in highly curved regions.

##### 2) Curvature based region segmentation

Discrete curvature calculation is based on vertex vicinity. In triangle sparse areas, most of vertices are on boundary edges, belonging to two or more regions. Discrete curvature estimation in this case is not reliable, resulting in wrong segmentation. To avoid this, we detect the part's sharp edges. After the B-Rep triangular model construction, we compute the dihedral angle between two adjacent triangles. All the sharp boundary edges can be found by a threshold on dihedral angle. Using a propagating approach, preliminary regions demarcated by sharp edges can be obtained. Follow-up curvature calculation will be confined to use vertices only within such a region. Curvature calculation is improved and is more reliable since the neighborhood of each vertex is not disturbed by vertices in other regions.

On a surface the local shape around its point P is characterized by maximum and minimum principal

curvatures ( $\kappa_{\max}$ ,  $\kappa_{\min}$ ) and by the two principal directions ( $d_{\max}$ ,  $d_{\min}$ ) corresponding to the tangent vectors for which the principal curvatures are obtained. The method used here adopts Cohen-Steiner's [8] and Zhao's [9] to compute the curvatures of a discrete shape. Shape index and curvedness [10] are two indicators derived from the principal curvatures. Shape index, ranged [-1,1], is a quantitative measure of local surface type. Curvedness is a positive value that specifies the amount or intensity of the surface curvature.

Ten surface types (spherical cup, trough, rut, saddle rut, saddle, spherical cap, dome, ridge, saddle ridge, and plane) are defined based on shape index and curvedness [11] (Fig. 4). Every vertex, except vertices of sharp edges, of the shape is assigned a surface type label during local surface type recognition. We use the curvedness value to detect smooth edges within a preliminary region. Vertices on smooth edges and transition regions have higher curvedness value, as thus a threshold can be given to detect them.

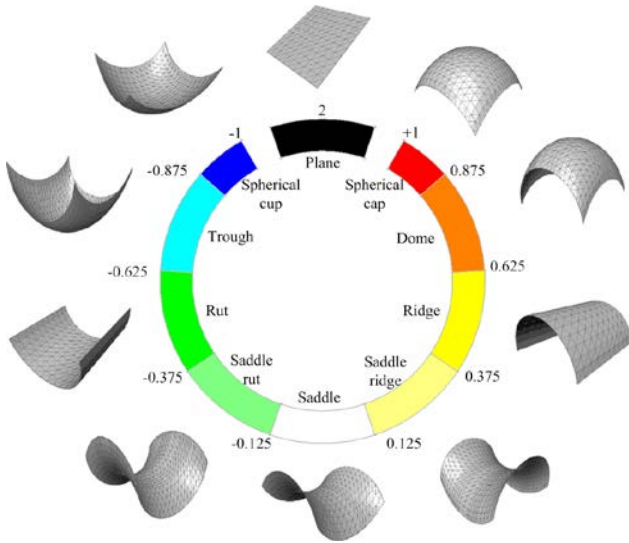
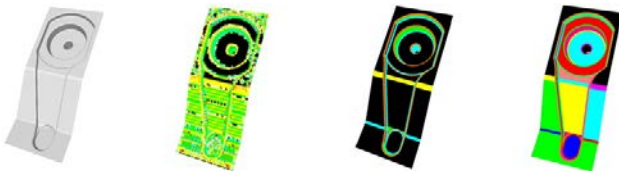


Fig. 4. Surface types, shape index scales and color scales.

Then connected regions are generated from these clustered vertices. Two operations are performed: connected region growing to generate initial segmentation result, and region refining, which aims to reduce over-segmented regions and to improve the segmentation result. When a vertex yet unassigned with a region label is met, the vertex is marked with a region label by its neighbor condition. With this associated region label, the vertex is added to an existing region or creates a new region. We adopt the approaches reported in [5,11] for the operations. Fig. 5 shows segmentation examples.



a) b) c) d)

Fig. 5. Segmentation example: a) discrete shape; b) initial clustering; c) cluster refining; d) segmentation.

### 3) Machining Feature Recognition

#### ① Surface B-rep model building

To obtain the boundary representation, we use the meshes' additional information, which help to know which meshes belong to the same surface and what is the type of the surface to be constructed.

Edge loops that form the surface are identified and built. This can be done using the topology of the already built polyhedral B-rep model. The process starts with finding a boundary edge whose vertices are on the surface. Following the first edge direction, find the next edge that links the first edge and is on the surface, repeat the similar search until all edges that can form a loop are found. CASCADE is used to build the curve from these edges, and all loops of the surface can be built in this way. Thus a continuous surface of the part is built. When all regions are exhaustively searched, the part's surface B-rep model is built. At present the surface types of the model can be planar faces, spheres, cones, cylinders and tori.

#### ② Adjacency Graph Based Feature Recognition

Machining feature recognition is performed on the surface B-rep model obtained above. Here we adopt the valued-adjacency graph approach [12], in which the B-rep model is converted into a planar graph where, e.g., its nodes represent faces and its arcs represent edges. Additional information, such as edge-xity (a measure of concavity/convexity), is incorporated into the graph. The xity  $m$  is defined as follows (cf. Fig. 6):

$$m = (\vec{n}_1 \times \vec{n}_2) \cdot \vec{a}_1 \quad (3)$$

where:  $\vec{a}_1$  is the orientation of edge  $e$  of face  $F_1$ .  $\vec{n}_1$  &  $\vec{n}_2$  are the normals at point  $P$  of  $e$  for  $F_1$  &  $F_2$ , respectively.

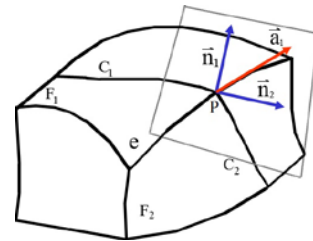


Fig. 6. Definition of the edge-xity.

A machining feature is regarded as a set of connected faces satisfying certain geometrical and topological conditions on the workpiece's B-rep. These faces have topological links of concavity that form cavities or protrusions and are related to a machining volume. For each form feature, it is possible to define a set of convex edges that are the limits of the feature: boundary edges. Note that the definition of a feature is only based on its topology. Seven generic classes for form feature

classification have been proposed [12,13]. General and specific rules to define the classes are based on properties of planar graphs.

K-Protrusion class definition depicts formal rules (Fig.7c) that have been developed with strong emphasis in graph theory. The classification rule consists of the identification of a base face which is connected to all the wall faces by concave edges. The wall faces are linked to each other with either convex or concave edges that form inner protruding volumes (Fig.7a). The rule can be easily described using cyclomatic and co-cyclomatic numbers of extracted planar graphs [12].

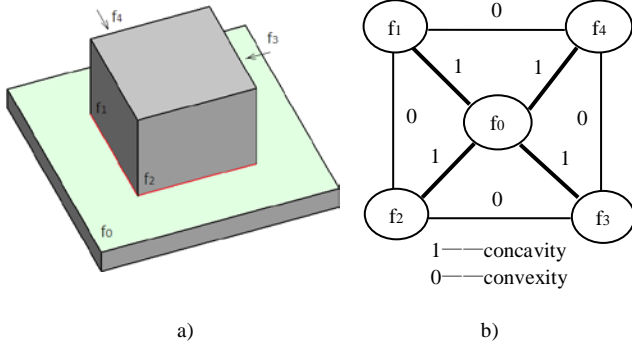


Fig. 7. Adjacency graph: a) a part; b) the adjacency graph; c) K-Protrusion classification rule.

The type and data structure of a machining feature adopt STEP-NC's definitions, so that the recognized feature data can be stored in conformance to the standard [14,15]. So far the types of machining features that can be recognized are closed pockets, open pockets, slots, steps, holes (Fig. 8). A machining feature is then regarded as a set of connected faces satisfying certain geometrical and topological conditions. For each feature type, its feature template is defined. A graph-based and rule based manufacturability analysis approach is used for validating the results. For example, as for the manufacturability of a step, multiple interpretations due to 3 potential machining directions are available in the extracted feature data.

From the reconstructed surface B-rep model or the machining feature based model, we can easily obtain form parameters to be controlled, and compare them with the

part's original CAD/CAM model. Hence we can evaluate the form errors under current machining operations.

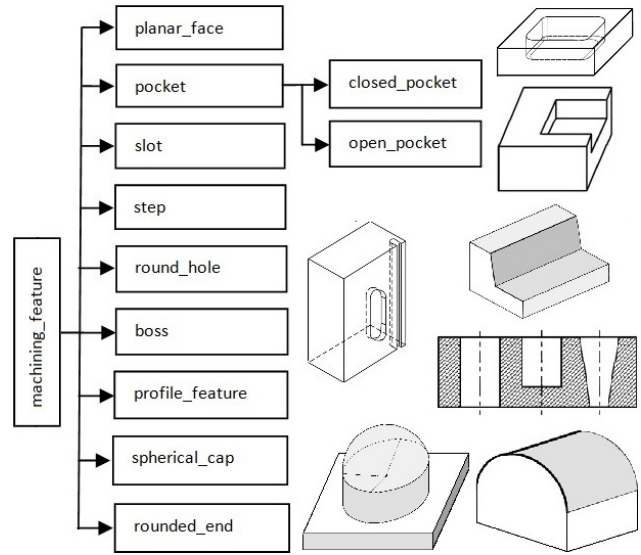


Fig. 8. STEP-NC features (adapted from [11]).

### C. An Example

The part in Fig. 9 has planar faces, holes, pockets, slots, etc. This figure shows some feature recognition results for the test part.

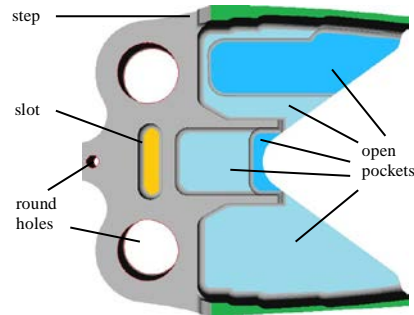


Fig. 9. A test part.

## IV. CONCLUSION

This paper proposes that feature recognition techniques can be applied to virtual machining for identifying some machining operation problems. Two types of features exist in in-process model of a simulation: machining features and surface micro features. We focus on machining features, especially on feature recognition from IPM, which involves B-rep model construction, curvature based region segmentation, and valuated adjacency graph based feature recognition. By comparing with the original CAD/CAM model, form errors (such as over/undercut, out-of-tolerance dimensions) of the features can be found. So performing feature recognition

on the VM output model can help the VM analysis in finding potential machining operation problems.

The work is in an initiating phase. Future works include: for the work of recognition from G-code, to increase the capacity of machining feature recognition, especially for region features; for the recognition from IPM, to develop specific approaches of region segmentation for other quadric/free-form surfaces and to test the approach on more complex parts, such as parts that are manufactured by five-axis and mill-turn machines. Future work will also verify that feature recognition can play a part for virtual machining.

#### ACKNOWLEDGMENT

This work is a part of the ANGEL FUI project funded by the French Inter-ministerial Fund and endorsed by top French competitiveness clusters (SYSTEMATIC PARIS REGION "Systems & ICT", VIAMECA "Advanced Manufacturing" and ASTECH "Aeronautics & Space").

#### References

- [1] Y. Altintas, P. Kerting, D. Biermann, E. Budak, B. Denkena, and I. Lazoglu, "Virtual process systems for part machining operations," *CIRP Annals – Manuf. Tech.*, 2014
- [2] Y. Zhang, X. Xu, Y. Liu, "Numerical control machining simulation: a comprehensive survey", *Int. J. of Computer Integrated Manufacturing*, vol.24, no.7, 2011, pp.593–609
- [3] S. Joshi, "Graph-based heuristics for recognition of machined features from a 3D solid model". *Computer-Aided Design*, vol.20, no.2, 1988, pp.58–66
- [4] S. XU, N. Anwer, S. Lavernhe, "Conversion of G-code part programs for milling into STEP-NC", in *Proc. of Joint Conf. on Mech., Design Eng. & Adv. Manuf.*, Toulouse, France, June 2014
- [5] S. XU, N. Anwer, M.-S. Charyar, "Machining feature recognition from in-process model of NC simulation" (Accepted for publication). *Computer-Aided Design & App.*
- [6] H. Zhao, N. Anwer, P. Bourdet, "Curvature-based registration and segmentation for multisensor coordinate metrology", *Procedia CIRP*, 10, 2013, pp.112-118
- [7] CASCADE, <http://www.opencascade.org>
- [8] D. Cohen-Steiner, J.-M. Morvan, "Restricted Delaunay triangulation and normal cycle", In *19th Annual ACM Symposium on Computational Geometry'03*, 2003, 312-321
- [9] H. Zhao, N. Anwer, P. Bourdet, "Curvature-based registration and segmentation for multisensor coordinate metrology", *Procedia CIRP*, 10, 2013, pp.112-118
- [10] J.J. Koenderink, A.J. Doorn, "Surface shape and curvature scales", *Imaging and Vision Computing*, vol. 10, no. 8, 1992, pp. 557–565
- [11] N. Anwer, Y. Yang, H. Zhao, O. Coma, J. Paul, "Reverse engineering for NC machining simulation," In *IDMME'2010—Visual Concept 2010*, Bordeaux, France
- [12] N. Anwer, "Méthodologie d'analyse de raisonnement pour la génération automatique des gammes d'usinage en fraisage", Ph.D. dissertation, ÉNS de Cachan, France, 2000
- [13] N. Anwer, A. Chep, B. Anselmetti, "Feature modeling and machinability analysis for CAD/CAPP Integration", IEEE Int. Conf. on Intelligent Eng. Sys., INES98, Austria, 1998
- [14] ISO 14649 Part 10: General process data, 2002
- [15] ISO 14649 Part 11: Process data for milling, 2002