Proceedings of the 2009 IEEE
International Conference on Mechatronics and Automation
August 9 - 12, Changchun, China

An Integrated CAD/CAPP System for Bases

Bin Zhu and Yantao Wang
School of Electromechanical Automobile Engineering
Yantai University
Yantai, Shandong Province, China
zbtnyt@sina.com, tomsmarter@163.com

Abstract **- CAPP is expected to bridge the gap between CAD and CAM, in which feature recognition is an active research and a critical component. Much research works have been done to attempt to realize a generic CAPP system, however it proves difficult because of the complexity of objects. This paper proposes a specialized CAD/CAPP integration system, called BaseCAM, for components, bases, in auto-body welding lines to reduce user interaction as much as possible. Its overall structure and some key techniques such as feature recognition, feature-based data exchange, numeric control programming strategies and tool path planning for hole machining are discussed. This system is implemented on the platform of the commercial software, UGS/UG and has been used in the design of auto-body welding lines very successfully.**

Index Terms **- bases, machining feature recognition, data exchange, tool path optimiaztion.**

I. INTRODUCTION

With the rapid development of the automobile industry, the fierce competition of market urges auto enterprises to reduce the manufacturing cycle of cars so as to react to the change of market as soon as possible. The precision and time of the design and manufacturing of auto-body welding lines affect the manufacturing cycle of cars directly[1]. Fig. 1 shows a station in an auto-body welding line, in which the base functions as a platform to support jigs. All of jigs are fixed on bases through brackets using bolts and pins. The commercial CAD/CAM system, UG(Unigraphics), is used in the design and manufacturing of bases, but due to its limit, manufacturing engineers have to pick up thread holes and pin holes, which will be denoted together as location holes for brevity, in a base in a manual method, so the efficiency of the numeric control (NC) programming for it is very low. In this paper, an integrated CAD/CAPP system called BaseCAM has been developed to improve the design efficiency of bases. This paper is organized as follows: The base building process is given in Section II. The architecture of the system is described in Section III. Section IV describes the recognition of machining features in a base. The feature-based data exchange is shown in Section V. Machining strategies for a base and the tool path optimization for the machining of location holes are described in Section VI. Conclusions are drawn in Section VII.

II. BASE BUILDING PROCESS

With the traditional approach, at first a designer finishes the design of a base with UG/CAD, and then sends the result file to a manufacturing engineer, who picks up location holes in the base one by one with a mouse to get the center coordinates and is responsible for the generation of the following types of NC code files:

- A NC file for all of location holes;
- NC files for pin holes with different diameters respectively;
- A NC file for slots.

All of NC files are made in a manual manner, in which picking up hole center coordinates is the most time-consuming job. Especially if a wrong hole were selected, the manufacturing engineer would have to resume the process.

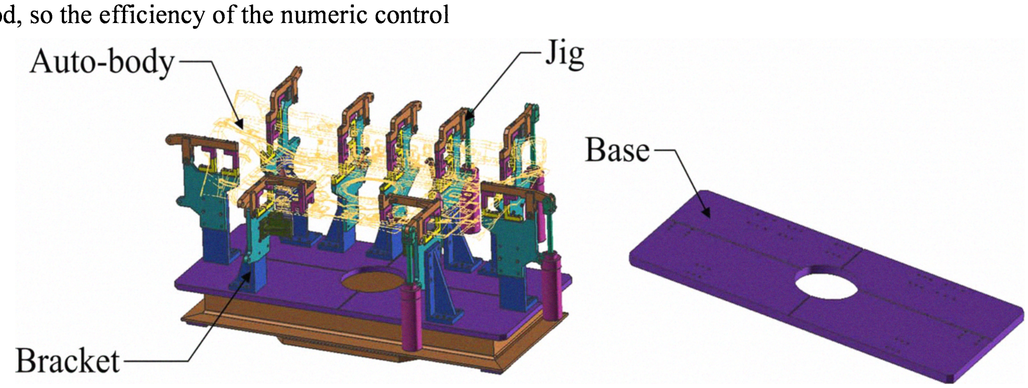


Fig. 1 Instance of a station in an auto-body welding line.

III. SYSTEM ARCHITECTURE

The BaseCAM is written in C++ language and UG/Open API, and is embedded in the UG. Base building process with the system is shown in Fig. 2. It consists of two modules, design module and manufacturing module, running on the computers of designers and manufacturing engineers respectively. Its primary functions can be described as follows:

1) Design module: recognizes machining features in a base; generates the XML-based and feature-based description file for the solid model of a base; enables a designer to check the recognized location holes using the graphic means.

2) Manufacturing module: parses the XML files and reconstructs the base geometric model for manufacturing engineers to check; generates NC code files for location holes and slots automatically; optimizes the tool path for hole machining and displays the path graphically.

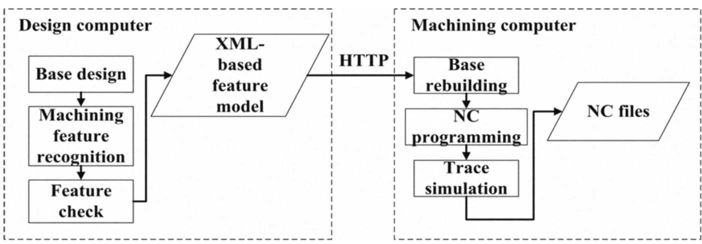


Fig. 2 Base building process with the BaseCAM system.

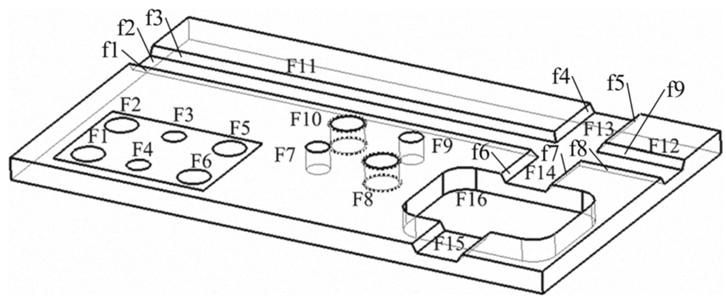
IV. MACHINING FEATURE RECOGNITION

How to effectively integrate CAD and CAPP is the key to improve the manufacturing efficiency of bases. Feature recognition has been a critical component for the integration of CAD and CAPP[2]. Some literatures have attempted to build a generic platform to finish the whole cycle from the recognition of machining features to the generation of NC codes automatically[3-5], however it still has a long way to go before completely put into practice really because of the complexity of objects, especially those with freeform shapes. Shin and Olling et al. [6] developed a specialized CAPP/CAM integration system for the automatic tool paths generation for a stamping die pattern and was used very successfully. Machining feature recognition is essential to automatic tool path generation. Various types of algorithms for feature recognition have been proposed in the literatures, but there are four attractive distinct approaches: graph pattern matching, convex hull decomposition, cell-based decomposition and hint-based reasoning[7]. In this research, the graph-based approach is used, and interacting features are recognized by back-tracing their original design features.

A. Feature Type Classification

There are four main types of machining features in a base: slot, thread hole, pin hole and through hole, as shown in Fig. 3.

It is noticed that F1~F6 are not form features and only represented with circle curves. They are the projections of the location holes of a bracket on the base, which are created using the WAVE Geometry Linker function in UG. It is such that design efficiency can be improved greatly.



F1,F2,F5,F6,F8,F10-thread hole feature F3,F4,F7,F9-pin hole feature
F11,F12,F13,F14,F15-slot feature F16-through hole feature
Fig. 3 Machining feature types in a base.

B. Graph-based Machining Feature Recognition

The graph-based method is first introduced by Joshi and Chang[8]. In this approach, first the B-rep of a part is transformed into an AAG(Attributed Adjacency Graph), whose nodes represent part faces and arcs represent part edges, both of which have an attribute of convexity or concavity. Then the graph is decomposed into sub-graphs by removing faces whose incident edges are all convex. Finally, the sub-graphs are analyzed to match the appropriate predefined feature templates. Fig. 4 shows the decomposed AAG for the F11~F14 features in Fig. 3, in which the nodes, f1~f9, represent the faces constituting features and the arcs have 0 and 1 attributes which denote the concavity and convexity of an edge respectively. The definitions for edge and face attributes given in ref. [9] are as follows:

Convex/concave faces. A face is said to be concave if the basis primitive(i.e. cone, sphere, torus) is hollow. A planar face is said to be neutral.

Convex/concave edges. If an edge is shared by faces *a* and *b* and α is the solid angle between *a* and *b*, then the edge is:

- Concave if $180^\circ < \alpha < 360^\circ$.
- Convex if $0^\circ < \alpha < 180^\circ$.
- Smooth convex if faces *a* and *b* are tangential to each other, and if both are convex or one is convex and the other is neutral.
- Smooth concave if faces *a* and *b* are tangential to each other, and if both are concave or one is concave and the other is neutral.
- Smooth neutral if faces *a* and *b* are tangential to each other and if one is concave and the other convex.

Fig. 5 shows an example of edge attributes.

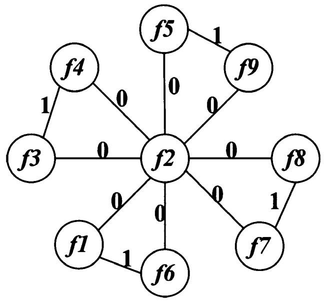


Fig. 4 Decomposed AAG for the F11~F14 features of Fig. 3.

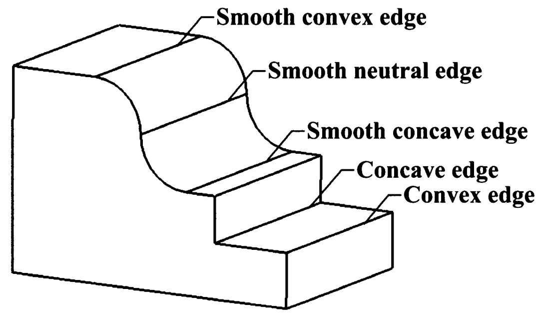


Fig. 5 Edge convexity and concavity attributes.

C. Recognition of Slot and Through Hole Features

The overview of the recognition of a slot or through hole feature can be represented by the following algorithm:

1. Calculate the attributes of all faces and edges of a base
2. Inner loop feature recognition
   2.1 Search each face for an inner loop with only convex edges, as indicates the existence of a through hole feature
   2.2 For each edge of the loop, find its incident surfaces and assign them to the feature
3. Non-inner loop feature recognition
   3.1 Construct the AAG of a base in the form of adjacency list
   3.2 Break the AAG into its concave sub-graphs
   3.3 For each sub-graph
      3.3.1 Search the sub-graph for a face that is concave or has a concave edge, which implies the existence of a feature
      3.3.2 Collect all faces in the feature
         3.3.2.1 Assign the face which is found in step 3.3.1 to the feature
         3.3.2.2 Find all concave edges of the face
         3.3.2.3 Assign all faces sharing the concave edges to the feature
         3.3.2.4 Get a face from the feature which is not traversed and go to step 3.3.2.2. If all of faces in the feature have been traversed, a specified feature has been found
   3.4 Traverse the feature to find whether there is a base face that is one which touches all of other faces in the feature; if no, a through hole is found; if yes, search side faces to find whether opposing sets of faces can be formed, if yes, a slot is found, if no, a pocket is found.
4. If features have the same design feature tag and are of the same machining feature type, they will be grouped into one machining feature. This step is for the identification of interacting features, which will be explained in the following section.

D. Interacting Feature Recognition

Graph-based methods are quite successful in recognition of isolated features, but reveal many difficulties when patterns of the part are altered due to feature intersections. Vandenbrande and Requicha [10] proposed the hint-based approach to deal with the problem of interacting features. Another representative approach was proposed by Gao and Shah[11], in which a hybrid of graph-based and hint-based approaches is used. However it is still difficult for the hint-based approaches to generating desirable interpretations[7].

In this research, design features are introduced to the machining feature recognition. As shown in Fig. 3, F11~F14 will be recognized as one slot feature because they share the same base face, F15 another independent slot feature according to the algorithm in Section C. However actually, F11 and F12 can be machined one time; F13, F14 and F15 should belong to one machining feature as well. This is the case because feature interaction results in the missing of some arcs between face nodes in AAG. From the design point of view, F11 and F12 is formed using the extruded operation in CAD, hence they share the same design feature tag, so do F13, F14 and F15. The pseudo-code for the interacting feature recognition is shown as follows:

MFs: machining features which have been recognized
FLsts: feature lists, each of which saves the faces constituting a machining feature, each face of which shares the same design feature tag and the same machining feature type.

```
For each feature m_f in MFs
  f_type = m_f.type
  For each face f in m_f
    f_tag = the design feature tag the face belongs to
    For each list l in FLsts
      If f_tag = l.design_feature_tag and f_type = l.machining_feature_type
        Insert f to l
        Break
      End If
    End For
    If f does not belong to any list in FLsts
      New a list l
      Insert f to l
      Add l to FLsts
    End If
  End For
End For
```

E. Recognition of Location Hole Features

Holes are the most important machining features in a base and used for the location of jigs. Hole features have two different types: thread holes and pin holes, where pin holes might have different diameters. So the key of the hole feature recognition is to not only distinguish thread holes from pin holes but also identify pin holes with different diameters. As mentioned in Section A, holes from brackets are represented with circles so as to improve design efficiency, so regular graph-based approach can not be used to recognize such features. The WAVE Geometry Linker function in UG is used for recognizing such location hole features. Its procedure, as shown in Fig.6, can be described as follows:

1) Assign attributes to thread holes and pin holes on a bracket respectively;

2) Assemble the bracket on a base, and use WAVE Geometry Linker operation to map the geometric elements, lines and circles, on the bottom of the bracket onto the surface of the base, which assures that the geometric elements from the mappings of the bracket have the same linker feature tag;

3) The system automatically searches each circle, then finds its corresponding hole on the bracket according to the linker feature tag, finally assigns a proper feature type to the circle in terms of the predefined hole attribute.

Additionally, a pin hole not from brackets must be created with hole operation in UG. For a thread hole not from brackets, first a designer must use the hole operation to create a hole, then add a thread feature to it, so that the system might distinguish pin holes from thread holes according to thread features.

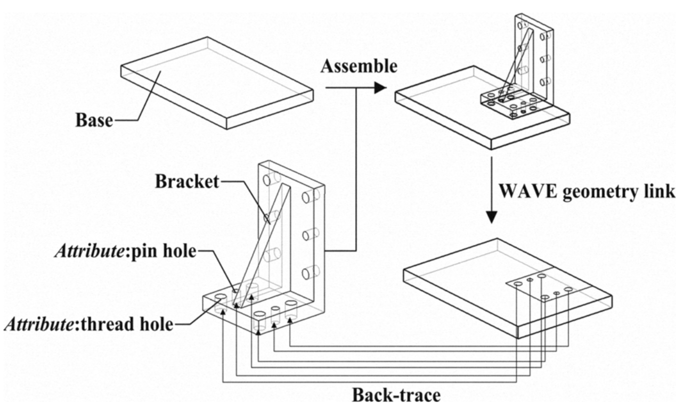


Fig. 6 Location hole feature recognition process.

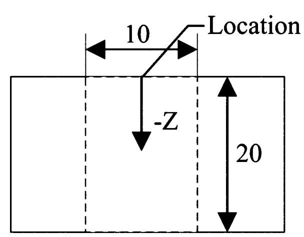
V. FEATURE-BASED DATA EXCHANGE

After the recognition of machining features, how to rapidly transmit these feature information from designers to manufacturing engineers is another key problem. Currently the data exchanges between different CAX still mainly depend on file service such as STEP, IGES, STL and DXF. However they result in too much volume of data transmission and only carry low-level geometric information instead of machining semantic, where although STEP supports feature definition, it is not mature and sometimes leads to the loss of constraints and feature information during the data exchange[12]. Some literatures have studied solid model description and exchange methods to increase transmission speed[13-15].

In this research, the feature-based data exchange is used and the transmission message is represented with XML[16]. All types of features in a base can be created using extruded operation, so they can be defined uniformly as follows:

feature ::=*<feature_ type><parameter>*
parameter:: = *<location><profile><depth><orientation>*
location ::= *<point>*
profile ::= *<arc>*|*<line>*
depth ::= *<positive real number>*
orientation ::= *<point>*
arc ::= *<center><radius><start_angel><end_angle>*
line ::= *<start_point><end_point>*

For example, the XML-based description of an pin hole feature is shown as follows:




```
<feature_type value="pin" />
<parameter>
  <location x="0.0" y="0.0" z="0.0" />
  <profile>
    <arc>
      <center x="0.0" y="0.0" z="0.0" />
      <radius value="5.0" />
      <start_angle value="0.0" />
      <end_angle value="360.0" />
    </arc>
  </profile>
  <depth value="20.0" />
  <orientation x="0.0"  y="0.0" z="-1.0" />
</parameter>
```

Fig. 7 Instances of a pin hole and its XML description.

After the machining computer receives a XML file from a design computer, the file will be parsed, and then the base will be reconstructed through the reconstruction of features, as shown by the following procedure:

1) Generate the profile sketch of each feature;

2) Extrude the profile to form the feature.

VI. NUMERIC CONTROL PROGRAMMING

A. Strategies for Machining Bases

A base is machined with the combination of manual and NC machining means.

- Its profile and the through holes on it are machined with the flame cutting manually, for which no NC codes are needed.
- Slots and location holes are machined on a NC machine. For location holes, after their center coordinates are extracted, the system will create NC codes automatically. For slots, the system extracts their section geometric parameters and their length, and then calls UG/Open API functions to generate the NC codes.

B. Tool Path Optimization for the Machining of Location Holes

When machining location holes, the moving time of the tool occupies a large portion in the whole cycle of machining, so optimizing hole machining trace is an effective approach to improve the machining efficiency. Apparently it is a TSP

(Traveling Salesman Problem), which is a NP-hard problem. In this research, GA(Genetic Algorithm) is used to optimize a hole machining trace.

In the algorithm, the path representation is chosen for the coding of chromosomes; the roulette wheel selection mechanism is used to select individuals with higher fitness; the order crossover operator is used for generating a child from two parent chromosomes; for mutation operation, two genes in a chromosome are randomly selected and exchanged; see [17] for details. The shortest tool path is the objective of optimization, so the fitness function can be defined:

$$Maximize \qquad F = \frac{1}{\sum_{i}^{n}\sum_{j}^{n} d_{ij}u_{ij}} \tag{1}$$

where d_{ij} denotes the distance between hole i and hole j; u_{ij} means whether the tool move from hole i to j directly, if yes, it equals 1, otherwise it equals 0. The generation number is used as a stop measure. Some key parameters, such as crossover ratio, mutation ratio and generation number, can be configured according to requirements in the system. Fig. 8 shows the tool path for pin hole machining after optimization.

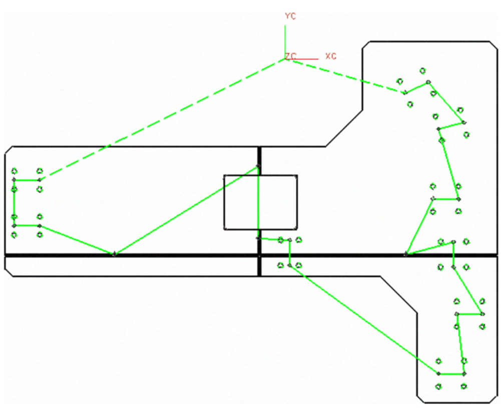


Fig. 8 Tool path simulation for pin hole machining.

VII. CONCLUSION

In this paper, an integrated CAD/CAPP system for bases in auto-body welding lines has been implemented and used in Wooshin(Yantai) Ltd. very successfully. It would take about 20 minutes for a manufacturing engineer to carry out the NC programming for a medium-complex base as shown in Fig. 8 in the past; using the BaseCAM, the job can be finished approximately within 5 minutes. Especially for the bases only with location holes from brackets, the speed of the NC programming almost only depends on that of a computer.

In addition, the feature recognition function exists as an independent DLL(Dynamic Link Library) file embedded in UG, so it might apply to the machining feature recognition of other components. Currently we are developing another CAD/CAPP/CAM integration system for disk arms in hard disk drives based on this function module.

Although the BaseCAM is a highly specialized CAD/CAPP integration system, we hope that its integration approach provide valuable suggestions to other manufacturing domains.

ACKNOWLEDGMENT

The authors wish to thank Wooshin(Yantai) Ltd. for many helpful suggestions. This research was supported by Yantai university foundation for young scholars under Grant JX07Z9.

REFERENCES

[1] X.P. Xiong, C.Z. Li and Q.D. Jin, "The design of auto-body welding Line based on the virtual manufacturing & assembling environment," New Technology & New Process, no. 3, pp. 2-4, March 2004.

[2] M. Mantyla, D. Nau and J. Shah, "Challenges in feature-based manufacturing research," Communications of the ACM 1996, vol.39, no.2, pp. 77-85, February 1996.

[3] H.K. Miao, N.K. Sridharan and J.J. Shah, "CAD-CAM integration using machining feautures," Computer Integrated Manufacturing, vol. 15, no. 4, pp. 296-318, July 2002.

[4] X.H. Zhou, Y.J. Qiu, G.R. Hua, H.F. Wang and X.Y. Ruan, "A feasible approach to the integration of CAD and CAPP," Computer-Aided Design, vol. 39, no. 4, pp. 324-338, April 2007.

[5] Y. YILDIZ, İ. KORKUT and U. ŞEKER, "Development of a feature based CAM system for rotational parts," G.U. Journal of Science , vol. 19, no. 1, pp. 35-40, 2006.

[6] H.Y. Shin, G.J. Olling, Y.C. Chung, B.H. Kim and S.K. Cho, "An integrated CAPP/CAM system for stamping die pattern machining," Computer-Aided Design, vol. 35, no. 2, pp. 203-213, February 2003.

[7] J.H. Han, M. Pratt and W.C. Regli, "Manufacutring feature recognition from solid models: a status report," IEEE Transactions on Robotics and Automation, vol. 16, no. 6, pp. 782-796, December 2000.

[8] S. Joshi and T.C. Chang, "Graph-based heuristics for recognition of machined features from 3D model," Computer-Aided Design, vol. 20, no. 2, pp. 58-66, March 1988.

[9] X. Xu and S. Hinduja, "Recognition of rough machining features for 2.5D components," Computer-Aided Design, vol. 30, no. 7, pp. 503-516, June 1998.

[10] J.H. Vandenbrande and A.A.G. Requicha, "Spatial resoning for automatic recognition of machinable features in solid models," IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 12, pp. 1269-1285, December 1993.

[11] S. Gao and J.J. Shah, "Automatic recognition of interacting machining features based on minimal condition sub-graph," Computer-Aided Design, vol. 30, no. 9, pp. 727-739, August 1998.

[12] M.J. Pratt and B.D. Anderson, "A shape modeling applications programming interface for the STEP standard," Computer-Aided Design, vol. 33, no. 7, pp. 531-543, 2001.

[13] J.Y. Lee, S.B. Han, H. Kim and S.B. Park, "Network-centric feature-based modeling," Proceedings of Pacific Graphics'99, IEEE Computer Society, Seoul, Korea, pp. 280-289, 1999.

[14] D. Wu, S. Dhargave and R. Sarma, "Solid model streaming as a basis for a distributed design environment," Proceedings of DETC 2000 ASME Design Engineering Technical Conferences, Baltimore, Maryland, DETC2000/DAC-14250, 2000.

[15] G.H. Choi, D.W. Mun and S.H. Han, "Exchange of CAD part models based on the macro-parametric approach," International Journal of CAD/CAM, vol. 2, no. 1, pp. 13-21, 2002.

[16] Y. Xi, A heterogeneous collaborative design based on feature data exchange, M.S. thesis, Zhejiang University, Zhejiang, China, 2006, pp. 28-31.

[17] L.P. Wang and L.M. Cao, Genetic Algorithms: Theory, Applications and Software Implementation, Xian: Xi'an Jiaotong University Press, 2002, pp. 123-126.