

Integrated Manufacturing Systems

The International Journal of Manufacturing
Technology Management

Manufacturing automation

Guest Editors: Dr Y.H. Chen and Dr Ian Gibson



Integrated Manufacturing Systems

The International Journal of Manufacturing Technology Management

Volume 14, Number 7, 2003

Manufacturing automation

Guest Editors: Dr Y.H. Chen and Dr Ian Gibson

Contents

-
- | | |
|--|---|
| 550 Access this journal online | 586 Developing the methods of modeling heterogeneous components
<i>Ke-Zhang Chen and Xin-An Feng</i> |
| 551 Abstracts & keywords | 593 Intelligent process-planning system or optimal CNC programming – a step towards complete automation of CNC programming
<i>Millan K. Yeung</i> |
| 553 Guest editorial | 599 Agent-based architecture for manufacturing system control
<i>C.K. Fan and T.N. Wong</i> |
| 554 Most efficient tool feed direction in three-axis CNC machining
<i>Zezhong C. Chen, Zuomin Dong and Geoffrey W. Vickers</i> | 610 Machining feature extraction for casting parts
<i>B.F. Wang, Y.F. Zhang and J.Y.H. Fuh</i> |
| 567 Research on the process model of product development with uncertainty based on activity overlapping
<i>Renbin Xiao and Shangwen Si</i> | 624 Awards for Excellence |
| 575 GA-driven part e-manufacturing scheduling via an online e-service platform
<i>Yingfeng Zhang, Pingyu Jiang and Guanghui Zhou</i> | |

Access this journal electronically

The current and past volumes of this journal are available at

www.emeraldinsight.com/ft

You can access over 100 additional Emerald journals, each with a comprehensive searchable archive of articles (many dating back to 1989), a detailed classification system and links to referenced material.

See page following contents for full details of what your access includes.



Access this journal online

www.emeraldinsight.com/ims.htm



As a subscriber to this journal, you can benefit from instant, electronic access to the content of this title. Your access includes a variety of features that increase the value of your journal subscription.

Automatic permission to make up to 25 copies of individual articles

This facility can be used for teaching purposes, training courses, meetings or seminars. This only applies to articles of which Emerald owns copyright. For further details visit www.emeraldinsight.com/copyright

How to access this journal electronically

To benefit from electronic access to this journal you first need to register on the Internet. Registration is simple and full instructions are available online at www.emeraldinsight.com/register Once completed, your institution will have instant access to the journal content from www.emeraldinsight.com or via the journal homepage www.emeraldinsight.com/ims.htm

Our liberal institution-wide licence allows everyone within your institution to access your journal electronically, thus making your subscription more cost-effective. Our Web site has been designed to provide you with a comprehensive, simple system that needs only minimum administration. Access is available via IP authentication or username and password.

Key features of Emerald electronic journals

Online Publishing and Archiving

You can gain access to past volumes as well as new material from this journal on the Internet via Emerald Fulltext. You can browse or search the database for relevant articles.

Key Reading

This feature provides abstracts of articles chosen by the journal editor on the basis of their subject-matter. These are selected to provide readers with current awareness of interesting articles from other publications within the same field.

Reference Linking

Direct links are provided from the journal article references to abstracts of the most influential articles cited (where possible this is to the full text of the article).

E-mail an Article

This facility allows you to e-mail relevant and interesting articles in PDF format to another PC for later use, reference or printing purposes.

Additional Complementary Services Available

Your access includes a variety of features that add to the functionality and value of your journal subscription:

E-mail Services

Emerald's range of free e-mail alerting services is designed to deliver personal notification of news and features in a number of different interest areas.

Emerald WIRE (World Independent Reviews)

Emerald WIRE is a fully searchable, subject-specific database brought to you by Emerald Management Reviews. It provides article reviews from the world's top management journals. The database is updated monthly and gives users details of how to obtain the full text of original articles.

Research Register

Located at www.emeraldinsight.com/researchregister, the Emerald Research Register is an Internet research forum where you and your readers can identify information on research activity world-wide. This feature is invaluable if you are seeking information or wish to disseminate information about your own research.

Support Resources

Comprehensive librarian and user toolkits have been created to help you get the most from your journal subscription. For further information about what is available visit www.emeraldinsight.com/usertoolkit

Choice of Access

Electronic access to this journal is available via a number of channels, enabling end users and libraries to reach the content through their preferred delivery system. The Emerald Fulltext Web site – www.emeraldinsight.com/ft – is the recommended means of electronic access, as it provides fully searchable and high value-added access to the complete content of the journal.

Subscribers can also access and search the article content of this journal through the following journal delivery services:

EBSCOhost Electronic Journals Service (EJS)

ejournals.ebsco.com

Huber E-Journals

e-journals.hanshuber.com/english/index.htm

Minerva

www.minerva.at

OCLC Firstsearch Electronic Collections Online

www.oclc.org/firstsearch/

SilverLinker

www.silverplatter.com

Swets Blackwell's SwetsWise

www.swetswise.com

Emerald Customer Support Services

For customer service and technical help, contact:

E-mail support@emeraldinsight.com

Tel +44 (0) 1274 785278

Fax +44 (0) 1274 785204



Abstracts & keywords

Most efficient tool feed direction in three-axis CNC machining

Ze Zhong C. Chen, Zuomin Dong and Geoffrey W. Vickers

Keywords Computer numerical control, Programming and algorithm theory, Control systems, Mechanical engineering

The objective of CNC machining is to produce mechanical parts with designed quality most efficiently. To generate CNC tool paths for machining a sculptured part using a three-axis CNC machine, surface geometry, cutter shape and size, as well as tool path interval and direction need to be considered. In this work, the relation between the direction of a tool motion and cutting efficiency is studied. A new measure of cutting efficiency in three-axis CNC milling – the length of effective cutting edge (ECE) is introduced. The ECE length is mathematically proven to reach its maximum when the tool cuts a sculptured surface along its steepest tangent direction at the cutter contact point. The steepest tangent direction is thus proven to be the most efficient tool feed direction in three-axis sculptured part machining. The study identifies tool feed direction as a new control parameter in CNC tool path planning, and forms the foundation for further research on three-axis tool path generation of sculptured parts.

Research on the process model of product development with uncertainty based on activity overlapping

Renbin Xiao and Shangwen Si

Keywords Product development, Time to market, Production planning and control

The essence of competition among modern enterprises is time; hence the time for product development must be greatly reduced. This paper presents a new process model of product development with uncertainty based on activity overlapping, borrowing ideas from the uncertainty model and activity overlapping model proposed by Loch and Terwiesch, and Krishnan respectively, to realize reduction on the time. Besides the fundamental framework of the proposed model, some derivations on the formula computing the total execution time of upstream and downstream activities are made. The effectiveness of the proposed model is verified through some further discussions and initial computational results.

GA-driven part e-manufacturing scheduling via an online e-service platform

Yingfeng Zhang, Pingyu Jiang and Guanghui Zhou

Keywords Manufacturing systems, Production scheduling, Bills of materials, Programming and algorithm theory, Computer aided manufacturing

This paper is mainly concerned with studying the e-manufacturing scheduling issue based on the bills of materials (BOM) flows of products. On the basis of e-manufacturing philosophy, a genetic algorithm (GA)-based scheduling model for physical e-manufacturing cells is put forward. By means of building the mapping relationship and dynamic task association among physical e-manufacturing cells, logic e-manufacturing cells, e-manufacturing systems and e-manufacturing environment, high-level scheduling issues can be transformed as an inductive procedure concerning the scheduling results to corresponding physical e-manufacturing cells. As verification, finally, a case study is given to demonstrate the method mentioned above.

Developing the methods of modeling heterogeneous components

Ke-Zhang Chen and Xin-An Feng

Keywords Component manufacturing, Modelling, Materials management, Computer aided design

In order to represent, analyze, optimize, and manufacture a component made of multi-heterogeneous materials for high-tech applications, a computer model of the heterogeneous component needs to be built first. Heterogeneous materials include composite, functionally graded materials, and heterogeneous materials with a periodic microstructure. Current modeling techniques focus only on capturing the geometric information and cannot satisfy the requirements from modeling the components made of multi-heterogeneous materials. This paper develops a modeling method, which can be implemented by employing the functions of current CAD graphic software and can obtain the model including both the material information (about its microstructures and constituent composition) and the geometry information without the problems arising from too many data.

Intelligent process-planning system or optimal CNC programming – a step towards complete automation of CNC programming

Millan K. Yeung

Keywords Process planning, Artificial intelligence, Knowledge management, Computer numerical control, Computer aided design

One of the bottle-necks of computer numerical control (CNC) machining is the CNC programming. It relies on the experience and skills of the CNC programmer for the generation of the CNC program. The intelligent process-planning system described in this paper generates a process plan automatically for CNC programming. It utilizes artificial intelligent technologies such as knowledge base, blackboard system and machine learning to extract machineable features and proposes and selects optimal tools for the machining of the given part. Its flexibility and simplicity provide a convenient way to include new techniques and knowledge. The incorporation of this system with other CAD/CAM tools could effectively automate the CNC programming process.

Agent-based architecture for manufacturing system control

C.K. Fan and T.N. Wong

Keywords Flexible manufacturing systems, Object-oriented programming, Modelling, Machine oriented languages

A flexible manufacturing system (FMS) is a complex manufacturing system and it demands a robust control software for its scheduling, planning and control functions. This paper describes the development of an agent-based infrastructure for the control of a cellular FMS. The FMS in this project is a flexible assembly cell (FAC), comprising two assembly robots and a conveyor system. The aim is to establish a multi-agent control system with good expandability and to be able to cope with dynamic changes in the FAC. The proposed agent-based FAC control system comprises a collection of agents implemented in a distributed control network. The approach of the agent design is based on the object-oriented modelling technique. According to the proposed control architecture, a standard agent template has been designed for the establishment of individual agents in the agent-based system.

Machining feature extraction for casting parts

B.F. Wang, Y.F. Zhang and J.Y.H. Fuh

Keywords Manufacturing systems, Programming and algorithm theory, Boolean functions, Modelling

An approach to extract machining features for casting parts is presented. It is capable of recognizing interacting machining features. There are five phases in the recognition process: Boolean difference of the final part model and the raw part; identification of machining faces (M-faces) from the final part model and the raw part model; decomposition of the removed simple volumes into delta simple volumes based on M-faces; gluing the delta simple volumes into sets of feasible simple volumes based on M-faces; testing. This strategy is process-oriented and feature-independent. It recognizes all features that can be produced by common machining operations in a uniform way and produces alternative sets of machining features.

Guest editorial

About the Guest Editors

Dr Y.H. Chen is currently an Associate Professor in the Department of Mechanical Engineering, The University of Hong Kong (HKU). Before joining HKU, Dr Chen worked in industrial organizations such as Motorola and Matsushita as research and development engineer since he obtained his PhD in 1991 from The University of Liverpool, UK. Dr Chen has filed one patent, co-authored a book, and authored or co-authored over 50 refereed papers, either published in international journals or delivered at conferences, in the areas of machine vision, computer-aided design and manufacturing. His current research interests include engineering design, reverse engineering, haptic modelling and layered manufacturing. Dr Chen is a Chartered Engineer and a member of IMechE.

Dr Ian Gibson is an Associate Professor in the Department of Mechanical Engineering, The University of Hong Kong (HKU). He is a specialist in advanced manufacturing technology and is particularly interested in rapid prototyping (RP) and product development. He obtained his Bachelor's and PhD at the University of Hull in the UK and was a lecturer in manufacturing technology at Nottingham University. He is co-editor of the *Rapid Prototyping Journal* and responsible for a number of initiatives in RP research and development, including the book *Software Solutions for Rapid Prototyping* and the formation of the Global Alliance of RP Associations. He is a chartered engineer and member of IEE, and a committee member of the Manufacturing and Industrial Division of the HK Institution of Engineers.

In December 2002, we had the pleasure and the privilege of hosting the International Conference on Manufacturing Automation (ICMA 2002) at the The University of Hong Kong. The Conference was full paper submission, employing a referee process that resulted in just over 60 top-class papers. A number of these papers were in subject areas relevant to *Integrated Manufacturing Systems* and we selected seven of them for this special journal issue. These papers were resubmitted in a longer form and then subjected to further review before being accepted for publication. As a result, the selection and review process was perhaps even more rigorous than the standard review for the journal. I hope you agree with our view that the papers are indeed of a high standard and both interesting and relevant to *IMS* journal readers.

The ICMA has a distinct "software technology" feel to it and there is no surprise that some of the best papers were heavily software-focused. However, the technological feel is also very strong, as can be seen from the fact that three papers are machining-based. Product development was a very specific theme in this ICMA and the paper by Xiao and Si illustrates this theme very well. Also the paper by Zhang *et al.* describes work in another theme area of e-manufacturing. Readers looking for something a little different in terms of research will be interested in the paper by Chen and Feng on heterogeneous solid modelling, a topic that is sure to grow as the technology for making heterogeneous components matures. We consider that all the papers certainly in this

selection are worthy of journal publication and they illustrate both the breadth and the depth of the ICMA.

Nowadays there is a tendency to hold very specialized conferences in very specific areas like materials, CAD, process planning, RP, etc. One of the main objectives of the ICMA was to bring together researchers in fields that are similar, but with a slightly greater diversity than these other, more specialized conferences. We were also very fortunate to obtain funding from the K.C. Wong Education Foundation, which enabled a number of mainland Chinese delegates to attend. It is still quite difficult to find out about the research being carried out in China and we feel that the ICMA provided a useful means for Chinese researchers to present their work to a wider audience. Delegates from China had to submit papers on a competitive basis for support from the K.C. Wong Foundation and so this Conference allowed researchers from outside China to really get to know the good research that is being carried out there.

Hosting conferences is a tough yet rewarding experience. The ICMA proved to be highly enjoyable for both the organizers and (through the feedback we gained) the delegates. Our co-organisers (Huazhong University of Science and Technology and The National University of Singapore) both contributed to a very successful event and we look forward to the next ICMA in Wuhan in 2004.

Y.H. Chen and Ian Gibson
Hong Kong University

Most efficient tool feed direction in three-axis CNC machining

Zezhong C. Chen

Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Quebec, Canada

Zuomin Dong

Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada

Geoffrey W. Vickers

Department of Mechanical Engineering, University of Victoria, Victoria, BC, Canada

Keywords

Computer numerical control, Programming and algorithm theory, Control systems, Mechanical engineering

Abstract

The objective of CNC machining is to produce mechanical parts with designed quality most efficiently. To generate CNC tool paths for machining a sculptured part using a three-axis CNC machine, surface geometry, cutter shape and size, as well as tool path interval and direction need to be considered. In this work, the relation between the direction of a tool motion and cutting efficiency is studied. A new measure of cutting efficiency in three-axis CNC milling – the length of effective cutting edge (ECE) is introduced. The ECE length is mathematically proven to reach its maximum when the tool cuts a sculptured surface along its steepest tangent direction at the cutter contact point. The steepest tangent direction is thus proven to be the most efficient tool feed direction in three-axis sculptured part machining. The study identifies tool feed direction as a new control parameter in CNC tool path planning, and forms the foundation for further research on three-axis tool path generation of sculptured parts.

1. Introduction

Owing to the wide application of sculptured parts in the aeronautical, automotive, and die/injection moulding industries, the manufacturing of these parts becomes a significant business. The complex shape of these parts imposes many technical challenges to the generation of effective tool paths for CNC machining to produce the parts with adequate surface quality and maximum cutting efficiency. Over the years, extensive researches have been carried out on CNC tool path generation for three-axis sculptured part machining (Bobrow, 1985; Huang and Oliver, 1994; Suresh and Yang, 1994). Adequate surface quality is ensured when the extremities of all cusps on a machined surface lie within its designed surface tolerance zone. On the other hand, redundant machining is to be prevented to save machining time.

Earlier researches on generating tool paths are simply based on the geometry of the sculptured surfaces. The iso-parametric tool path generation method creates tool paths using equally spaced iso-parametric curves in the surface parametric space (Broomhead and Edkins, 1986; Loney and Ozsoy, 1987). This straightforward approach is easy to implement, but is unable to control tool path intervals across the surface. As an improvement, the plane-guided tool path scheme generates tool paths in the 3D Cartesian space, using the intersection curves between the sculptured surface and a number of parallel planes (Bobrow, 1985).

More recent researches on three-axis tool path planning consider surface and cutter geometry and tool path interval to control both surface quality and cutting efficiency. Some methods adjust the tool path interval according to the local shape of the surface to reduce redundant machining. The

representative iso-cusped tool path generation method adjusts the tool path intervals to maintain equal-height cusps on the finished surface (Huang and Oliver, 1994; Sarma and Dutta, 1997; Suresh and Yang, 1994). Because the heights of all cusps across the surface are equal to the given surface tolerance, redundant machining is reduced considerably, and the total machining time is thus saved. The steepest-directed tree method forms tool paths based on surface features following a hill-climbing approach (Maeng *et al.*, 1996). The method first casts a mesh on to the sculptured surface and identifies primitive surface features at each mesh node. Candidate tool paths are formed by connecting surface nodes from the valley to the ridge along upward directions. The final tool paths are generated by synthesizing all candidate tool paths, milling one simple surface feature each time. The approach leads to improved machining efficiency.

These developments have considerably advanced the three-axis CNC tool path generation techniques. However, these tool path generation methods are based on simple tool path generation principles, which may be quite efficient in machining some regions of the sculptured surface, but may not necessarily lead to maximum cutting efficiency for the entire surface. For instance, the iso-cusped method generates tool paths one after the other by offsetting from the previous tool path and making the cusps between two adjacent tool paths equal to the given tolerance. However, some of the tool paths arranged purely following this method often conflict with other machining efficiency considerations, such as the efficient direction of a tool motion. These tool paths can gradually lose cutting efficiency, removing less and less material. On the other hand, tool paths planned with the steepest-directed tree method often present



Integrated Manufacturing Systems
14/7 [2003] 554-566

© MCB UP Limited
[ISSN 0957-6061]
[DOI 10.1108/09576060310491919]

The Emerald Research Register for this journal is available at
<http://www.emeraldinsight.com/researchregister>



The current issue and full text archive of this journal is available at
<http://www.emeraldinsight.com/0957-6061.htm>

either partial overlaps in some areas or a large number of disconnected tool path segments, depending on the shape of the surface. The overlapped tool paths produce unnecessarily low cusps and fine surface, leading to longer machining time and higher machining cost. To generate globally effective tool paths, a full understanding of the relationship between tool path geometry and its cutting efficiency becomes crucial.

In this work, the relation between tool feed direction and cutting efficiency is closely examined and mathematically modeled. The most efficient tool feed direction in three-axis sculptured surface machining using flat and torus end-mills is identified analytically. A new cutting efficiency measure for three-axis CNC milling, the length of the effective cutting edge (ECE), is introduced. The general formulae of the ECE length for flat, torus and ball end-mills are derived. The ECE length of a tool motion has been proven mathematically to reach its maximum when the tool cuts along the steepest tangent direction on the sculptured surface at the cutter contact (CC) point. In this case, the cutting efficiency of the motion reaches the maximum and the tool feed direction is most efficient. The proof provides the foundation for a new tool path generation method that determines tool feed direction for high cutting efficiency. Various tool feed directions for machining a hemi-cylindrical example part and their corresponding ECE lengths are used to demonstrate the method. A comparison of the theoretical and experimental ECE lengths is made to verify the most efficient tool feed direction.

2. Cutting efficiency measure

To find the relationship between the tool feed direction and cutting efficiency of a tool motion, the mechanism of three-axis CNC machining is examined, four coordinate systems are introduced, and effective cutting edge is defined.

2.1 Generic machining model

A sculptured surface that is machined using a torus end-mill on a three-axis vertical CNC milling machine is shown in Figure 1. The working table of the machine moves along the X - and Y -axes; the cutter travels along the Z -axis. Tolerance surface is defined as an offset of the sculptured surface by the given surface tolerance. Tool-path cutting efficiency is determined by the cutting efficiency of each tool motion in the path. In an efficient tool motion, the cutter removes a large amount of excess material between the tolerance surface and the design surface. To

quantify the cutting efficiency of a tool motion and to identify the most efficient direction of a tool motion, a generic machining model of the cutter and surface is introduced.

In the generic machining model illustrated in Figure 2, a torus end-mill whose cutting surface is in frying-pan shape contacts a sculptured surface at a cutter contact (CC) point. When the cutter feeds from a lower CC point to the next upper CC point along a tool feed direction, the envelope of the torus cutting surface is formed in this tool motion. The formation of this envelope can be also considered as a result of a motion of a planar cutting edge. This planar cutting edge is defined as the cutter projection profile, if the cutter is projected along the cutter feed direction on to an orthogonal plane. If the projection plane is on the plane of the paper, the planar cutting edge, the free-formed surface and the tolerance surface are shown in Figure 3. The Figure shows that all material that is above the planar cutting edge will be removed in machining. The portion of the cutting edge that is between the design and tolerance surfaces removes the last layer of excess material, determining the machined surface finish. It is thus called effective cutting edge (ECE). The effective cutting edge and its length are illustrated in Figure 3. From this diagram, it is easy to understand that the longer the ECE length, the larger the amount of the removed excess material. Since the ECE length is directly associated with the amount of material removed in a tool motion, it can serve as a cutting efficiency measure for the tool motion.

2.2 Expressions of cutter and cutting surface normal in cutter motion frame

To derive the mathematical formulae of the planar cutting edge and the ECE length, four coordinate systems (or frames) are introduced. These include cutter geometry frame (CGF), cutter location frame (CLF), steepest direction frame (SDF), and cutter motion frame (CMF). Based on the location and orientation relations between each pair of the successive frames, the expressions of the cutter and the cutting surface normal in the CGF are transformed sequentially (Faux and Pratt, 1979; Mortenson, 1987) through the CLF, the SDF, into the CMF, in which the cutter projection is easily obtained.

Cutter geometry frame

Suppose a CC point P_0 on a sculptured surface, S , is being machined on a three-axis vertical CNC machine with a torus end-mill; the general machining condition is described in Figure 4. In the part design coordinate system, the cutter axis is along the direction $[0, 0, 1]^T$.

Zezhong C. Chen,
 Zuomin Dong and
 Geoffrey W. Vickers
*Most efficient tool feed
 direction in three-axis CNC
 machining*

Integrated Manufacturing
 Systems
 14/7 [2003] 554-566

Figure 1
 Illustration of tool paths on machined surface

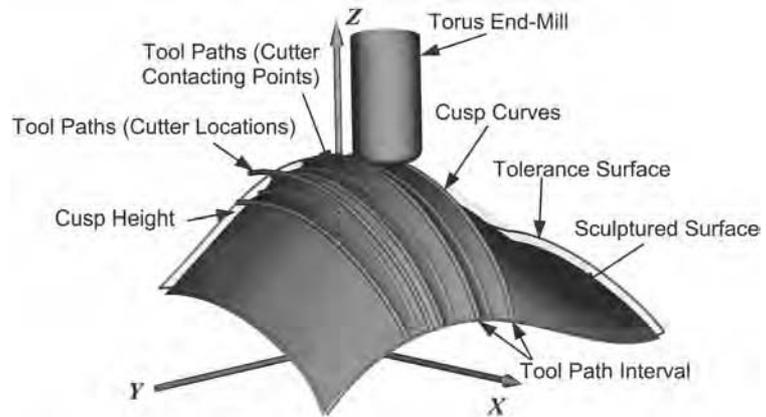


Figure 2
 A generic machining model of sculptured parts for ECE

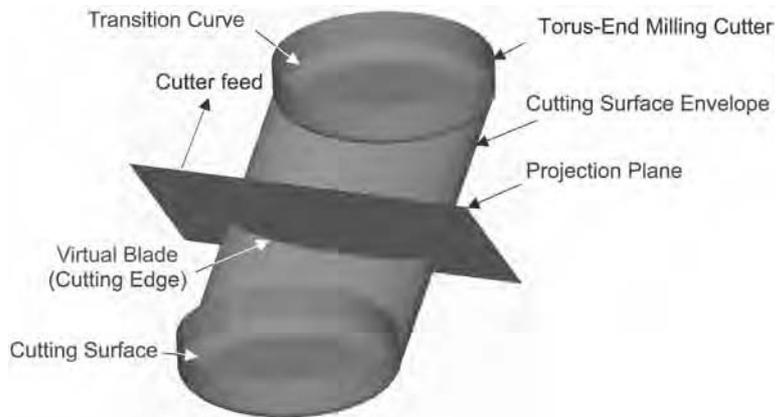


Figure 3
 ECE, sculptured surface and tolerance surface

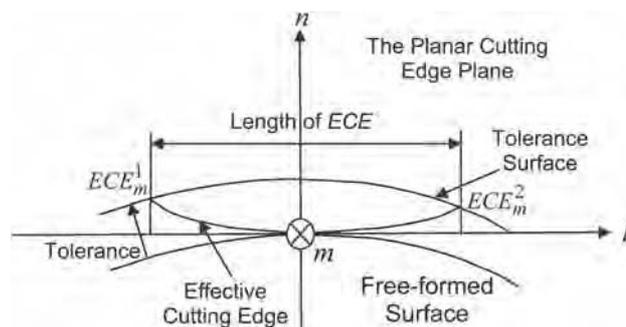
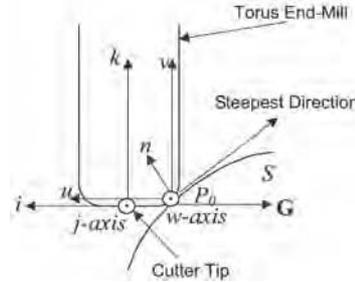


Figure 4
Torus end-mill in cutter geometry frame and
cutter location frame



The cutter geometry frame (i - j - k) is introduced with its origin located at the cutter tip (center); its k -axis is aligned with the cutter axis; and its i -axis is on the plane defined by the k -axis and the surface normal n at the CC point and points off the surface. The j -axis is perpendicular to the i - and k -axes. The expression of the torus cutting surface of the tool in the CGF is:

$$\begin{bmatrix} i \\ j \\ k \end{bmatrix} = \begin{bmatrix} (R_1 + R_2 \sin\phi) \cos\theta \\ (R_1 + R_2 \sin\phi) \sin\theta \\ R_2(1 - \cos\phi) \end{bmatrix}, \quad (1)$$

where $R_1 > 0$, $R_2 > 0$, and $R_1 \gg R_2$; $\phi \in [0, (\pi/2)]$, and $\theta \in [0, 2\pi]$. The geometric parameters of the torus end-mill are given in Figure 5.

In the same coordinate frame, the cylindrical cutting surface of the tool is represented as:

$$\begin{bmatrix} i \\ j \\ k \end{bmatrix} = \begin{bmatrix} (R_1 + R_2) \cos\theta \\ (R_1 + R_2) \sin\theta \\ R_2 + h \end{bmatrix}. \quad (2)$$

At the CC point P_0 , parameter θ is equal to π . The location of the point is specified with the other parameter ϕ as ϕ_0 . The coordinates of the CC point in the CGF are then represented as:

$$[-(R_1 + R_2 \sin\phi_0), 0, R_2(1 - \cos\phi_0)]^T.$$

Cutter location frame

The cutter location frame (u - w - v) is formed by translating the CGF from the cutter tip to

the CC point, P_0 . The i -, j -, and k -axes in the CGF become the u -, w -, and v -axes in the CLF, respectively, as shown in Figure 4. The vectors of v - and u -axes can be found in the part design coordinate system. The v -axis is vertical, $\mathbf{v} = [0, 0, 1]^T$; the u -axis is opposite to the surface gradient \mathbf{G} at P_0 , which is $[\partial z/\partial x, \partial z/\partial y, 0]^T$, discussed in Marsden and Weinstein (1985). The u -axis is then represented as $\mathbf{u} = [-\partial z/\partial x, -\partial z/\partial y, 0]^T$ in the part design coordinate system. The w -axis is defined as the cross-product of \mathbf{v} and \mathbf{u} , which is:

$$\mathbf{w} = \mathbf{v} \times \mathbf{u} = \left[\frac{\partial z}{\partial y}, -\frac{\partial z}{\partial x}, 0 \right]^T. \quad (3)$$

The derived expressions of the torus and the cylindrical cutting surfaces in the CGF can now be transformed in the CLF, as given in Equations (4) and (5):

$$\begin{bmatrix} u \\ w \\ v \end{bmatrix} = \begin{bmatrix} (R_1 + R_2 \sin\phi) \cos\theta + (R_1 + R_2 \sin\phi_0) \\ (R_1 + R_2 \sin\phi) \sin\theta \\ R_2 (\cos\phi_0 - \cos\phi) \end{bmatrix}. \quad (4)$$

$$\begin{bmatrix} u \\ w \\ v \end{bmatrix} = \begin{bmatrix} (R_1 + R_2) \cos\theta + (R_1 + R_2 \sin\phi_0) \\ (R_1 + R_2) \sin\theta \\ R_2 \cos\phi_0 + h \end{bmatrix}. \quad (5)$$

Steepest direction frame

The part surface normal \mathbf{n} is known as $[-\partial z/\partial x, -\partial z/\partial y, 1]^T$, and the steepest tangent direction, \mathbf{SD} , of the surface at the CC point P_0 is $[\partial z/\partial x, \partial z/\partial y, (\partial z/\partial x)^2 + (\partial z/\partial y)^2]^T$ (refer to Chen *et al.* (2001). Since the dot products between any pair of the vectors, \mathbf{n} , \mathbf{SD} , and \mathbf{w} , are null, these vectors are perpendicular with one another. They are used to build another coordinate frame, the steepest direction frame (n - w - SD), as shown in Figure 6. In this frame, n -, w -, and SD -axes are represented by the \mathbf{n} , \mathbf{w} , and \mathbf{SD} vectors, respectively. At the CC point $P_0(\pi, \phi_0)$ the angle between vectors \mathbf{n} and \mathbf{v} is ϕ_0 . The CLF coincides with the SDF when it is rotated along the w -axis by an angle of $(\pi/2 - \phi_0)$. The rotation will align the u - and v -axes with the n - and SD -axes, respectively. Any points on the surface of the torus cutter can be transformed from the CLF into the SDF through the following rotation about w -axis (Mortensen, 1987):

$$\begin{bmatrix} n \\ w \\ SD \end{bmatrix} = \begin{bmatrix} \sin\phi_0 & 0 & \cos\phi_0 \\ 0 & 1 & 0 \\ -\cos\phi_0 & 0 & \sin\phi_0 \end{bmatrix} \cdot \begin{bmatrix} u \\ w \\ v \end{bmatrix}. \quad (6)$$

Cutter motion frame

During machining, the cutter feeds in a direction tangent to the machined surface. The angle between this cutter feed direction, \mathbf{m} , and the steepest tangent direction, \mathbf{SD} is

Figure 5
Geometry of torus end-mill

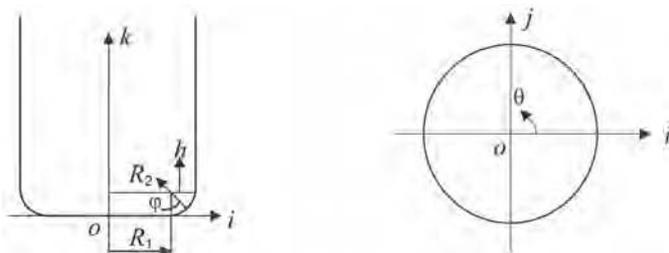
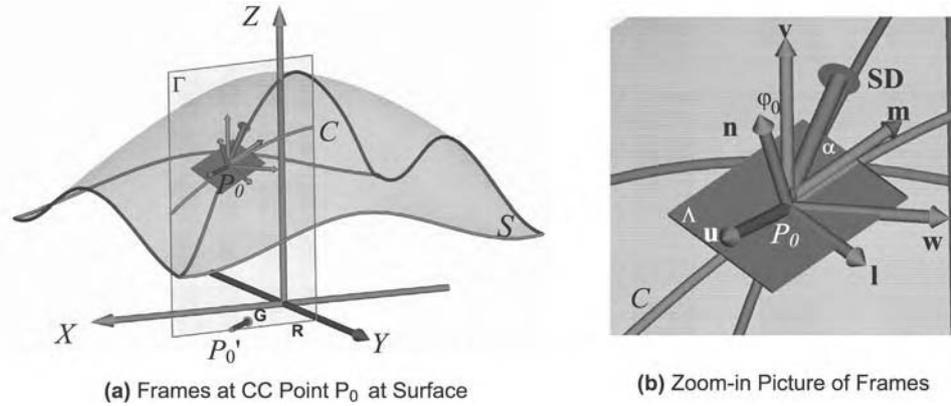


Figure 6

Illustrations of cutter location, steepest direction, and cutter motion frames



α , as shown in Figure 6. The cutter feed direction, \mathbf{m} , the part surface normal at the CC point, \mathbf{n} , and another vector, \mathbf{l} , defined by the cross-product of \mathbf{m} and \mathbf{n} , form another coordinate frame, the cutter motion frame (CMF). The orientation of this frame changes as the cutter feed direction, \mathbf{m} . The difference between the two frames, SDF and CMF, is an angle, α , about the n -axis. With a geometric transformation, the expressions of the introduced torus and cylindrical cutting surfaces of the cutter can be represented in the new CMF, as given in Equations (7) and (8).

The torus cutting surface is expressed in the CMF as:

$$\begin{bmatrix} n \\ l \\ m \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} n \\ w \\ SD \end{bmatrix} = \begin{bmatrix} a \cdot R_1 \cdot \cos\theta + a \cdot R_2 \cdot \cos\theta \cdot \sin\phi - b \cdot R_2 \cdot \cos\phi + a \cdot R_1 + R_2 \\ R_1 \cdot \sin\theta \cdot \cos\alpha + R_2 \cdot \sin\theta \cdot \sin\phi \cdot \cos\alpha + b \cdot R_1 \cdot \cos\theta \cdot \sin\alpha + b \cdot R_2 \cdot \cos\theta \cdot \sin\phi \cdot \sin\alpha + a \cdot R_2 \cdot \cos\phi \cdot \sin\alpha + b \cdot R_1 \cdot \sin\alpha \\ R_1 \cdot \sin\theta \cdot \sin\alpha + R_2 \cdot \sin\theta \cdot \sin\phi \cdot \sin\alpha - b \cdot R_1 \cdot \cos\theta \cdot \cos\alpha - b \cdot R_2 \cdot \cos\theta \cdot \sin\phi \cdot \cos\alpha - b \cdot R_1 \cos\alpha \end{bmatrix} \quad (7)$$

The cylindrical cutting surface is expressed in the CMF as:

$$\begin{bmatrix} n \\ l \\ m \end{bmatrix} = \begin{bmatrix} a \cdot (R_1 + R_2) \cdot \cos\theta + a \cdot R_1 + R_2 + b \cdot h \\ (R_1 + R_2) \cdot \sin\theta \cdot \cos\alpha + b \cdot (R_1 + R_2) \cdot \cos\theta \cdot \sin\alpha + b \cdot R_1 \cdot \sin\alpha - a \cdot h \cdot \sin\alpha \\ (R_1 + R_2) \cdot \sin\theta \cdot \sin\alpha - b \cdot (R_1 + R_2) \cdot \cos\theta \cdot \cos\alpha - b \cdot R_1 \cdot \cos\alpha + a \cdot h \cdot \cos\alpha \end{bmatrix} \quad (8)$$

where $\alpha = \sin\phi_0$ and $b = \cos\phi_0$.

In the CMF, plane ($n-l$) is the cutter projection plane that is perpendicular to the tool feed direction, m . The geometric representation of the cutter in the CMF can be projected on to the $n-l$ plane by replacing the value of m in Equations (7) and (8) with zero. The projection of the torus cutting surface becomes:

$$\begin{bmatrix} n \\ l \\ m \end{bmatrix} = \begin{bmatrix} a \cdot R_1 \cdot \cos\theta + a \cdot R_2 \cdot \cos\theta \cdot \sin\phi - b \cdot R_2 \cdot \cos\phi + a \cdot R_1 + R_2 \\ R_1 \cdot \sin\theta \cdot \cos\alpha + R_2 \cdot \sin\theta \cdot \sin\phi \cdot \cos\alpha + b \cdot R_1 \cdot \cos\theta \cdot \sin\alpha + b \cdot R_2 \cdot \cos\theta \cdot \sin\phi \cdot \sin\alpha + a \cdot R_2 \cdot \cos\phi \cdot \sin\alpha + b \cdot R_1 \cdot \sin\alpha \\ 0 \end{bmatrix} \quad (9)$$

The projection of the cylindrical cutting surface becomes:

$$\begin{bmatrix} n \\ l \\ m \end{bmatrix} = \begin{bmatrix} a \cdot (R_1 + R_2) \cdot \cos\theta + a \cdot R_1 + R_2 + b \cdot h \\ (R_1 + R_2) \cdot \sin\theta \cdot \cos\alpha + b \cdot (R_1 + R_2) \cdot \cos\theta \cdot \sin\alpha + b \cdot R_1 \cdot \sin\alpha - a \cdot h \cdot \sin\alpha \\ 0 \end{bmatrix} \quad (10)$$

Similarly, the mathematical representations of the surface normal of the torus and cylindrical cutting surfaces of the cutter can be transformed from the CGF to the CMF, using the same transformation. The unit normal of the torus cutting surface, $\mathbf{n}_{\text{torus}}$, in the CGF is:

$$\mathbf{n}_{\text{torus}} = \begin{bmatrix} n_i \\ n_j \\ n_k \end{bmatrix} = \begin{bmatrix} \cos\theta \cdot \sin\phi \\ \sin\theta \cdot \sin\phi \\ -\cos\phi \end{bmatrix} \quad (11)$$

and the unit normal of the cylindrical cutting surface, $\mathbf{n}_{\text{cylinder}}$, in the CGF is:

$$\mathbf{n}_{\text{cylinder}} = \begin{bmatrix} n_i \\ n_j \\ n_k \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix}. \quad (12)$$

After the coordinate transformations, the expression of the unit normal of torus cutting surface in the CMF becomes:

$$\mathbf{n}_{\text{torus}} = \begin{bmatrix} n_n \\ n_l \\ n_m \end{bmatrix} = \begin{bmatrix} a \cdot \cos\theta \cdot \sin\phi - b \cdot \cos\phi \\ \sin\theta \cdot \sin\phi \cdot \cos\alpha + b \cdot \cos\theta \cdot \sin\phi \cdot \sin\alpha \\ + a \cdot \cos\phi \cdot \sin\alpha \\ \sin\theta \cdot \sin\phi \cdot \sin\alpha - b \cdot \cos\theta \cdot \sin\phi \\ \cdot \cos\alpha - a \cdot \cos\phi \cdot \cos\alpha \end{bmatrix}. \quad (13)$$

Similarly, the unit normal of the cylindrical cutting surface in the CMF becomes:

$$\mathbf{n}_{\text{cylinder}} = \begin{bmatrix} n_n \\ n_l \\ n_m \end{bmatrix} = \begin{bmatrix} a \cdot \cos\theta \\ \sin\theta \cdot \cos\alpha + b \cdot \cos\theta \cdot \sin\alpha \\ \sin\theta \cdot \sin\alpha - b \cdot \cos\theta \cdot \cos\alpha \end{bmatrix}. \quad (14)$$

2.3 Effective cutting edge

Planar cutting edge calculation

Establishment of the four frames allows the cutter to be projected along the cutter feed direction onto the $n-l$ plane. The mathematical representations of the cutter projection are given in Equations (9) and (10). The periphery of the cutting surface projection is the planar cutting edge, as shown in Figure 7(a). Each point on the planar cutting edge corresponds to a point on

the cutting surface, at which the cutting-surface normal is perpendicular to the tool feed direction, i.e.:

$$\mathbf{m}^T \cdot \mathbf{n} = 0, \quad (15)$$

where \mathbf{m} is the tool feed direction, and \mathbf{n} is the cutting-surface normal. The planar cutting edge can be obtained by solving Equations (9) and (15), or Equations (10) and (15).

Effective cutting edge calculation

If the local shape around the CC point on the part surface is uniform, the tolerance surface intersects the planar cutting edge at two points, and the n component of its coordination is approximately equal to the surface tolerance, as shown in Figure 7(b) and earlier in Figure 3. Since the ECE is the portion of the planar cutting edge within the part surface and tolerance surfaces, the ECE can be obtained using Equations (9) or (10), (15) and (16):

$$a \cdot R_1 \cdot \cos\theta + a \cdot R_2 \cdot \cos\theta \cdot \sin\phi - b \cdot R_2 \cdot \cos\phi + a \cdot R_1 + R_2 \leq t$$

or

$$a \cdot (R_1 + R_2) \cdot \cos\theta + a \cdot R_1 + R_2 + b \cdot h \leq t, \quad (16)$$

where, t represents the surface tolerance.

Suppose the two ends of the ECE are $ECE_m^1(\theta_1, \phi_1)$ and $ECE_m^2(\theta_2, \phi_2)$, and their coordinates in the CMF are:

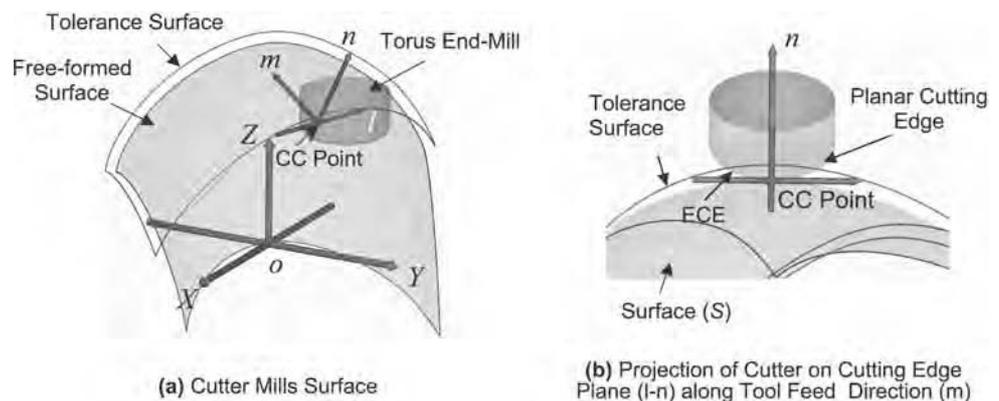
$$ECE_m^1[n_1 \ l_1 \ m_1]^T \text{ and } ECE_m^2[n_2 \ l_2 \ m_2]^T$$

(see Figure 3). The ECE length, L_{ECE} , is measured by the l -axis coordinate difference between the two points, and calculated by:

$$L_{ECE} = |l_1 - l_2|. \quad (17)$$

Figure 7

Cutter mills surface and projection of cutter on cutting edge plane ($l-n$) along tool feed direction (m)



3. Relationship between tool feed direction and cutting efficiency

When the tool feed direction, \mathbf{m} , changes in three-axis machining, the cutting surface projection on the $n-l$ plane in the CMF will change accordingly, as will the ECE. Thus, the length of ECE is a function of the tool feed direction, which is represented by the angle α between \mathbf{m} and \mathbf{SD} (see Figure 6). The length reaches maximum when the cutter feeds along a certain direction. Under these circumstances, the cutter removes the largest amount of material in the tool motion, so the tool feed direction is the most efficient. It is important to identify the most efficient tool feed direction in three-axis tool path planning.

4. Identification of most efficient tool feed direction

In this section, the generic ECE length formulae for the common cutters (ball, flat, and torus end-mill) are derived. The most efficient tool feed direction of a tool motion is identified for flat and torus end-mills. The ECE length for a ball end-mill is independent of the tool feed direction.

4.1 Consistent-efficiency tool feed direction for ball end-mills

A ball end-mill contains a spherical cutting surface, and is different from a torus end-mill by $R_1 = 0$. The ball end-mill expression in the CMF is simplified as:

$$\begin{bmatrix} n \\ l \\ m \end{bmatrix} = \begin{bmatrix} a \cdot R_2 \cdot \cos\theta \cdot \sin\phi - b \cdot R_2 \cdot \cos\phi + R_2 \\ R_2 \cdot \sin\theta \cdot \sin\phi \cdot \cos\alpha + b \cdot R_2 \cdot \cos\theta \cdot \sin\phi \\ \cdot \sin\alpha + a \cdot R_2 \cdot \cos\phi \cdot \sin\alpha \\ R_2 \cdot \sin\theta \cdot \sin\phi \cdot \sin\alpha - b \cdot R_2 \cdot \cos\theta \cdot \sin\phi \\ \cdot \cos\alpha - a \cdot R_2 \cdot \cos\phi \cdot \cos\alpha \end{bmatrix} \quad (18)$$

Suppose an end point coordinate of the ECE ($ECE_m^1(\theta_1, \phi_1)$) in the CMF is $ECE_m^1[n_1 \ l_1 \ m_1]^T$; the tool feed direction, \mathbf{m} , in the CMF is $[0, 0, 1]^T$. To find the coordinates of this end point, first simplify the Equations (15) and (16). The specific form of Equation (15) is:

$$\begin{aligned} \sin\theta_1 \cdot \sin\phi_1 \cdot \sin\alpha - b \cdot \cos\theta_1 \cdot \sin\phi_1 \\ \cdot \cos\alpha - a \cdot \cos\phi_1 \cdot \cos\alpha = 0. \end{aligned} \quad (19)$$

Since n_1 of the coordinate of point $ECE_m^1(\theta_1, \phi_1)$ is set to the surface tolerance, t , Equation (16) becomes:

$$a \cdot R_2 \cdot \cos\theta_1 \cdot \sin\phi_1 - b \cdot R_2 \cdot \cos\phi_1 + R_2 = t. \quad (20)$$

It is not difficult to find that l_1 (the l coordinate of point $ECE_m^1(\theta_1, \phi_1)$) is invariable with respect to angle α , because:

$$l_1^2 = 2 \cdot R_2 \cdot t - t^2. \quad (21)$$

As a result, the coordinate of point ECE_m^1 in the CMF is $[t \ -\sqrt{2R_2t - t^2} \ 0]^T$. Since the other end of the ECE (ECE_m^2) is symmetrical with respect to the n -axis on the plane ($n-l$), its coordinate is $[t \ -\sqrt{2R_2t - t^2} \ 0]^T$.

Hence, the ECE length for a ball end-mill can be calculated by:

$$L(\alpha) = 2\sqrt{2R_2t - t^2}. \quad (22)$$

Equation (22) indicates that the ECE length for a ball end-mill is independent of the tool feed direction and remains a constant. This expression proves that the cutting efficiency of a ball end-mill does not change with the tool feed direction.

4.2 Flat-end mills

A torus end-mill becomes a flat end-mill when the fillet radius R_2 drops to zero, which means that the torus-shaped cutting surface on the cutter degenerates to a circular cutting plane. A flat end-mill consists of a cylindrical cutting surface and a circular cutting plane. The general representation of a flat-end mill in the CMF can be obtained by substituting $R_2 = 0$ into Equations (7) and (8). The unit normal of the cylindrical cutting surface in the CMF is deduced to Equation (14). For example, the circular cutting plane in the CMF is:

$$\begin{bmatrix} n \\ l \\ m \end{bmatrix} = \begin{bmatrix} a \cdot R_1 \cdot \cos\theta + a \cdot R_1 \\ R_1 \cdot \sin\theta \cdot \cos\alpha + b \cdot R_1 \cdot \cos\theta \cdot \sin\alpha + b \cdot R_1 \cdot \sin\alpha \\ R_1 \cdot \sin\theta \cdot \sin\alpha - b \cdot R_1 \cdot \cos\theta \cdot \cos\alpha - b \cdot R_1 \cdot \cos\alpha \end{bmatrix}. \quad (23)$$

Suppose a flat end-mill is used to cut the part surface at point P_0 , and the tangent plane of the part surface at this point is plane Λ . With the four coordinate frames defined in Section 2, the cutter is projected on to plane $n-w$ in the CMF. The projection periphery defines the planar cutting edge. The two end points of the ECE are the intersections between the cutter and a plane Γ that is an offset of plane Λ by t , as shown in Figure 8(a). The plane Γ intersects the circular cutting plane as a line, and the cylindrical cutting surface as a curve. Meanwhile, Γ intersects the planes $n-SD$ and $n-w$ of the SDF at SD'

and w' , and the planes $n - m$ and $n - l$ of the CMF at m' and l' , respectively. The angle between SD - and m -axes is the same as the angle between SD' - and m' -axes, α .

Figure 8(b) is the view of the intersection from the positive n -axis. The intersection line is $Q_L - Q_R$ on the w' -axis, and the intersection curve is a curve $Q_L - P'_1 - Q_R$. When m is aligned with SD , the projection direction will be along the SD' -axis. The projection of the intersection forms line $Q_L - Q_R$. Points, Q_L and Q_R , are the end points of the ECE, ECE_{SD}^1 and ECE_{SD}^2 .

If m diverts from SD , the projection direction is along m' -axis, as shown in Figure 8(b), when α varies from zero to α_0 . The ends of the intersection projection will be at Q_L and Q_R , labelled as ECE_m^1 and ECE_m^2 , since the projections of Q_L and Q_R always lie at the ends of the intersection projection. If α is greater than α_0 , the projection of the intersection curve will be longer than the projection of the intersection line, and the projection of the intersection curve will define the ends of the ECE, not points Q_L and Q_R . The length of the projection shrinks as α increases. To find the maximum length of the ECE, the intersection only needs to be considered for α values between zero and α_0 .

The circular cutting plane in the CMF is given by Equation (23), and Γ in the CMF is specified by $n = t$, and points Q_L and Q_R can be found by solving Equation (24):

$$\begin{cases} a \cdot R_1 \cdot \cos\theta_1 + a \cdot R_1 = t \\ a \cdot R_1 \cdot \cos\theta_2 + a \cdot R_1 = t \end{cases} \quad (24)$$

$$\cos\theta_1 = \cos\theta_2$$

and

$$\theta_2 = \arccos\left(\frac{t}{a \cdot R_1} - 1\right)$$

can be derived from Equation (24). Since points ECE_m^1 and ECE_m^2 are symmetrical with the n -axis, the relation, $\sin\theta_2 = -\sin\theta_1$, holds. The maximum ECE length can be calculated through the following optimization formulation:

$$L_{\alpha \text{ maximum}} = |R_1 \cdot (\sin\theta_1 - \sin\theta_2) \cdot \cos\alpha + b \cdot R_1 \cdot (\cos\theta_1 - \cos\theta_2) \cdot \sin\alpha| = |2R_1 \sin\theta_2 \cos\alpha|. \quad (25)$$

This formulation shows that, when the variable α is equal to zero, that is, when m coincides with SD , the ECE length reaches its maximum. The value of the maximum ECE length, $(L_{\alpha})_{\max}$, is:

$$(L_{\alpha})_{\max} = \frac{2}{a} \sqrt{t^2 - 2 \cdot a \cdot R_1 \cdot t}. \quad (26)$$

Equation (25) indicates that the ECE length is a function of the angle α , and Equation (26) concludes that, when the tool feeds along the steepest direction at the CC point, the ECE length reaches its maximum. When the length of the ECE reaches its maximum, the largest amount of stock material above the cutting edge will be removed. The cutting efficiency reaches the maximum in the tool motion.

4.3 Torus end-mills

Identification of the maximum ECE length for a torus-end mill is more complex. The cutting surface, the offset plane, and their intersection are illustrated in Figure 9. The cutter touches the tangent plane, Λ , of the surface (not shown) at a CC point P_0 . Any Λ -offset plane intersecting the cutting surface generates a closed and convex curve, as

Figure 8
 Flat end-mill machining model and projection of intersection curve

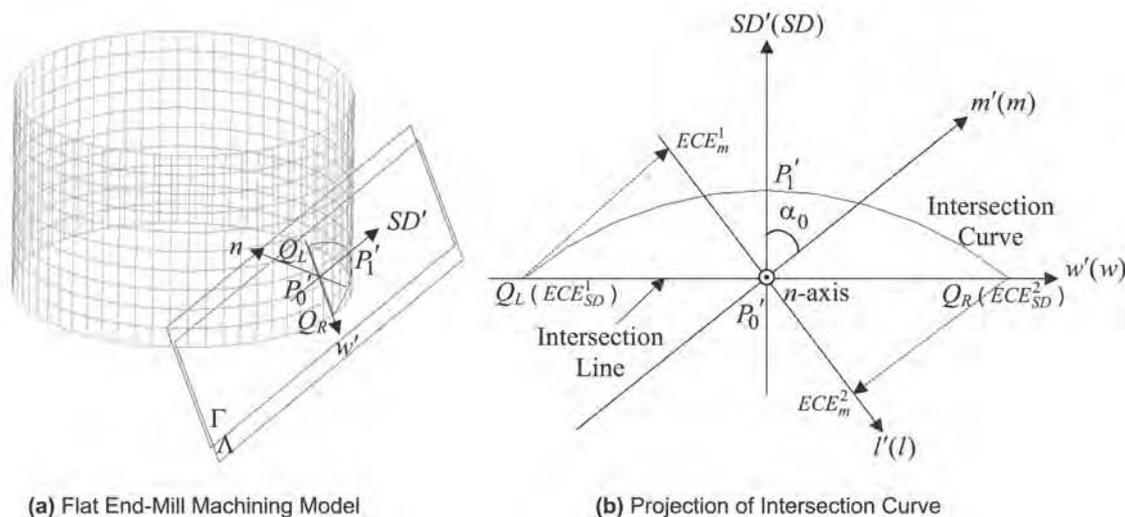
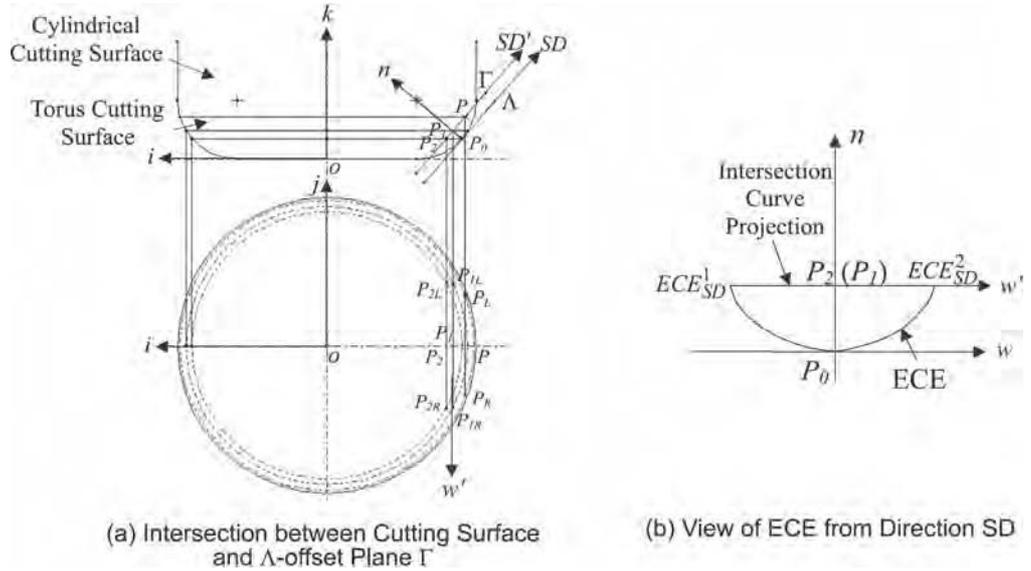


Figure 9

Intersection between cutting surface and Λ -offset plane Γ and view of ECE from direction SD



shown in Figure 10. When the intersection curve is projected along \mathbf{m} on to its orthogonal plane $(n-l)$, two points on the intersection curve, whose tangents are parallel with \mathbf{m} , map the ends of the intersection curve projection. These extremes of the projection define on the ECE. Figure 9(b) presents a special example of the ECE when \mathbf{m} is aligned with \mathbf{SD} .

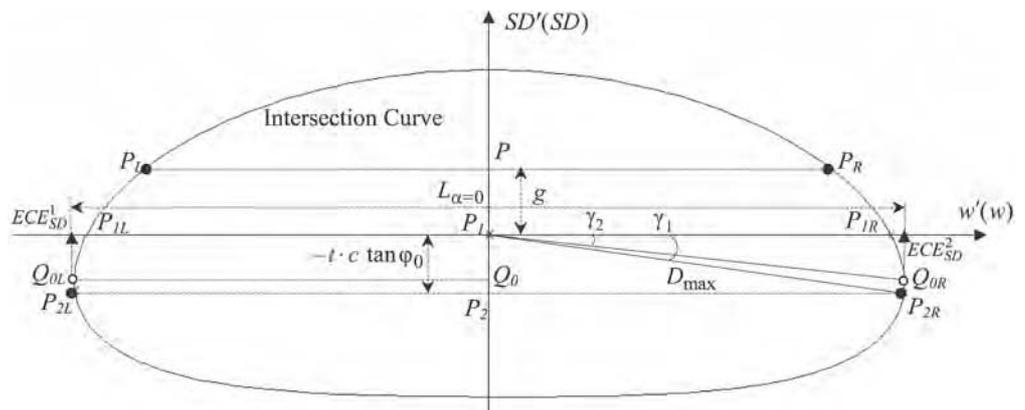
The ECE can be derived with the help of intersection curves. A series of planes, parallel to Λ , intersect the cutting surface and form a group of closed curves. If these curves are projected on to the plane $(n-l)$ along \mathbf{m} , each curve corresponds to two points on the ECE. In particular, the curve on a Λ -offset plane Γ by the value of the surface tolerance, t , specifies the two ends of the ECE. The difference of the l -component of these two end points defines the ECE length.

To find the maximum ECE length and the cutter feed direction, two steps are needed. First the ECE length ($L_{\alpha=0}$), when \mathbf{m} is aligned with \mathbf{SD} , is to be derived. Second, this ECE length is to be proved to be longer than any other ECE length (L_{α}) when \mathbf{m} diverts from \mathbf{SD} .

ECE length if tool feed direction is along steepest tangent direction, $L_{\alpha=0}$
 Suppose plane Γ intersects the n -axis at point P_1 , as shown in Figure 9(a). It also intersects the planes $(n-SD)$ and $(n-w)$ of the SDF at the lines SD' and w' , respectively. Line SD' is parallel to the SD -axis, and line w' is parallel to the w -axis. Similarly, Γ intersects the planes $(n-m)$ and $(n-l)$ of the CMF at the lines m' and l' , respectively. Line m' is parallel to the m -axis, and line l' is parallel to the l -axis. Point P_1 is the origin for both

Figure 10

View along n -axis on intersection curve on Γ



$SD' - w'$ and $m' - l'$. Figure 10 illustrates the intersection curve in $SD' - w'$, and its projection along SD' on to w' -axis. Two points on the curve, Q_{0L} and Q_{0R} , map the ECE ends, ECE_{SD}^1 and ECE_{SD}^2 .

Equation (6) is the representation of the cutter surface in the SDF. The plane Γ can be represented as $n = t$, and line SD is along $[0 \ 0 \ 1]^T$ in the SDF. After substituting these knowns into Equations (15) and (16), the two points on the curve, Q_{0L} and Q_{0R} , can be obtained using the following equations:

$$\begin{cases} -b \cdot \cos\theta \cdot \sin\phi - a \cdot \cos\phi = 0 \\ a \cdot R_1 \cdot \cos\theta + a \cdot R_2 \cdot \cos\theta \cdot \sin\phi - b \\ \cdot R_2 \cdot \cos\phi + a \cdot R_1 + R_2 = t \end{cases} \quad (27)$$

However, finding a closed-form solution to these equations is stunningly complex. One can employ numerical methods to solve these equations. Geometric analysis is used to prove that solutions exist for Equation (27).

The torus cutting surface of the cutter in the CGF has been defined in Equation (1), and the coordinate of the CC point $P_0(\pi, \phi_0)$ has been calculated previously in Section 2.2. In Figure 9, point P_1 is located at:

$$[-R_1 - (R_2 - t) \sin\phi_0, 0, R_2 - (R_2 - t) \cdot \cos\phi_0]^T$$

in the CGF, and the SD' -axis is represented in this frame as well:

$$\begin{bmatrix} i_{SD'} \\ j_{SD'} \\ k_{SD'} \end{bmatrix} = \begin{bmatrix} -R_1 - (R_2 - t) \cdot \sin\phi_0 - g \cdot \cos\phi_0 \\ 0 \\ R_2 - (R_2 - t) \cdot \cos\phi_0 + g \cdot \sin\phi_0 \end{bmatrix}, \quad (28)$$

where the line parameter, g , represents the distance between any point P on the axis and the point P_1 (see Figure 9). Through point P and perpendicular to the plane $(n - SD)$, a line, $P_L - P - P_R$, intersects the curve at two points, P_L and P_R (see Figures 9 or 10). These two points are symmetrical with SD' . When the value of g changes, points P_L and P_R move along the curve. In the SDF, the coordinates of P_L and P_R can be derived. For instance, the w value of P_R 's coordinates in the SDF is in the form of:

$$\begin{aligned} (w_{P_R})^2 = & 2R_1 \cdot \left\{ \sqrt{R_2^2 - [(R_2 - t) \cdot \cos\phi_0 - g \cdot \sin\phi_0]^2} \right. \\ & \left. - (R_2 - t) \cdot \sin\phi_0 - g \cdot \cos\phi_0 \right\} + R_2^2 - (R_2 - t)^2 - g^2. \end{aligned} \quad (29)$$

The distance ($D_{P_1 P_R}$) between any point P_R on the curve and the origin P_1 can be calculated:

$$\begin{aligned} (D_{P_1 P_R})^2 = & 2R_1 \cdot \left\{ \sqrt{R_2^2 - [(R_2 - t) \cdot \cos\phi_0 - g \cdot \sin\phi_0]^2} \right. \\ & \left. - (R_2 - t) \cdot \sin\phi_0 - g \cdot \cos\phi_0 \right\} + R_2^2 - (R_2 - t)^2. \end{aligned} \quad (30)$$

The distance ($D_{P_1 P_R}$) is a function of g . The maximum distance ($(D_{P_1 P_R})_{\max}$) can be found by maximizing this distance with respect to the variable, g , and the value of g_{\max} is:

$$g_{\max} = -t \cdot \tan\phi_0. \quad (31)$$

The calculated g_{\max} refers to an across point P_2 between SD' and a horizontal line passing through point, P_0 , as shown in Figure 9(a). Assume a line normal to the plane $(n - SD)$ intersects the cutting surface at two symmetrical points, P_{2L} and P_{2R} (see Figure 10). The distance (D_{\max}) between P_{2L} or P_{2R} and P_1 is the longest among all the distances between any other point on the curve and point P_1 . The coordinate of point P_{2R} in the SDF $(n - w - SD)$ is:

$$\left[t, \frac{1}{\sin\phi_0} \sqrt{2 \cdot R_1 \cdot t \cdot \sin\phi_0 + 2 \cdot R_2 \cdot t \cdot \sin^2\phi_0 - t^2}, -t \cdot \tan\phi_0 \right],$$

and the coordinate of point P_{2L} is similar to P_{2R} except for a negative w component.

In Figure 10, the angle between the line $P_1 P_{2R}$ and the line w' is γ_1 . The surface normal of the cutter at P_{2R} is calculated. Its SD component is:

$$b \cdot \sin\phi_0 - a \cdot \cos\phi_0 - \frac{t}{R_1 + R_2 \cdot \sin\phi_0}.$$

This implies that point P_{2R} is not the point on the curve, which maps the end of the ECE, since the surface normal at this point does not meet the perpendicularity condition, $\mathbf{m}^T \mathbf{n} = 0$. There must exist a point Q_{0R} between P_{1R} and P_{2R} and a point Q_{0L} between P_{1L} and P_{2L} , which map the ends of the ECE, ECE_{SD}^1 and ECE_{SD}^2 .

Suppose the angle between $P_1 Q_{0R}$ or $P_1 Q_{0L}$ and the line w' is γ_2 . When \mathbf{m} is in line with \mathbf{SD} , the projections of the points Q_{0L} and Q_{0R} on the w' -axis defines the ends of the ECE. Hence, the ECE length $L_{\alpha=0}$ becomes:

$$L_{\alpha=0} = Q_{0L} P_1 \cdot \cos\gamma_2 + Q_{0R} P_1 \cdot \cos\gamma_2. \quad (32)$$

Maximum ECE length ($L_{\alpha=0}$)

When the cutter feed direction, \mathbf{m} , is not aligned with the steepest tangent direction, \mathbf{SD} , and the angle between them, α , is not zero, the intersection curve should be projected along \mathbf{m} on to the plane $n - l$ of the CMF, rather than along \mathbf{SD} on to the plane $n - w$ of the SDF. The angle between \mathbf{m} and \mathbf{SD} is the same as the angle between m' and SD' , as is the angle between w' and l' , as shown in Figure 11. The two ends of the ECE can be identified by projecting the intersection curve along m' on to l' -axis (see Figure 11).

Assume two points, Q_L and Q_R , on the intersection curve are projected as the two ends of ECE, ECE_m^1 and ECE_m^2 , the angle between the lines $P_1 Q_R$ and $P_1 Q_{0R}$ is β_1 , and

5. Case study

To verify that the steepest tangent direction is the most efficient tool feed direction in three-axis sculptured part machining, a hemi-cylindrical part with radius of 50mm and length of 100mm, as shown in Figure 12, is machined using a VM-5 Victor four-axis machining center. In the machining, this part represented with a parametric form $S(\omega, \nu)$ is set up horizontally. The part material is Ren-Shape corrugated fibreboard. The cutter is a 6.35mm flat end-mill, and the specified surface tolerance is 0.2mm. In order to compare the ECE lengths along different tool feed directions, the cutter starts from six points, A to F, and feeds along different directions. These points share the same parameter ω of 80° . The different tool feed directions, measured by the angle α between the tool feed direction and the steepest tangent direction of the surface, are listed in Table I.

A close look of the surface at points A to F is shown in Figure 13. The ECE length at each point is measured manually, and the corresponding theoretical ECE length is calculated using Equation (27). The maximum discrepancy of both corresponding ECE lengths is less than 6.5 per cent. The discrepancy source includes hand measurement, cutter size, tool feed direction, etc. The case study demonstrates that the ECE length reaches the maximum when the tool feeds along the steepest tangent

direction, as indicated at point A, and decreases considerably when the cutter feeds away from this direction. The steepest tangent direction is proven to be the most efficient tool feed direction of a tool motion.

6. Conclusions

In this work, the relationship between tool feed direction and cutting efficiency is studied. The most efficient tool feed direction for three-axis sculptured surface machining using common tools (flat, torus and ball end-mills) is identified. A new cutting efficiency measure for three-axis CNC milling, the length of the effective cutting edge, is introduced. The formulae of the ECE lengths for commonly-used cutters are derived. While the ECE lengths for flat and torus end-mills are mathematically proven to reach their maximum when the tool moves along the steepest tangent direction on the sculptured surface at the cutter contact point, the ECE length for a ball end-mill is independent of the tool feed direction of the tool motion. A testing example of hemi-cylindrical parts is machined, and the ECE lengths in different tool feed directions are measured. After the comparison of theoretical and experimental ECE lengths, the most efficient tool feed direction is verified.

This study provides a better understanding on the relationship between tool path

Figure 12
Horizontal cylindrical part and tool feed directions

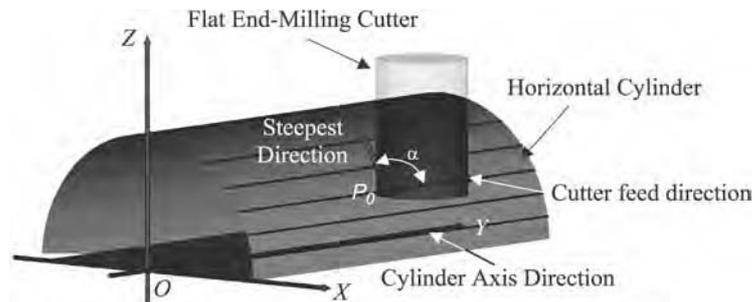


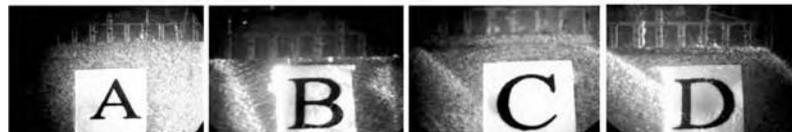
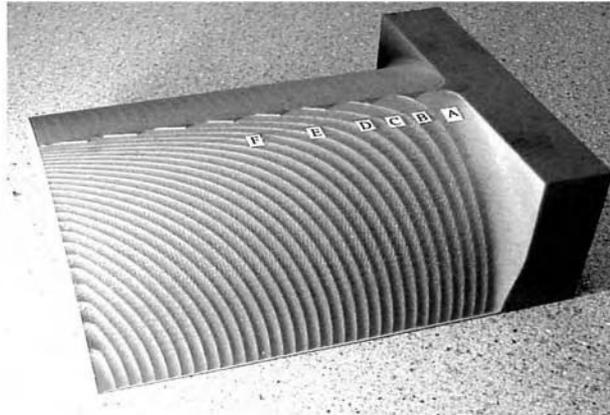
Table I
ECE length along different tool feed directions for common cutters

Points	Angle α	Measured ECE length (mm)	Theoretical ECE length (mm)	Discrepancy (%)
A	0	7.518	7.294	3.1
B	25.6	6.883	6.611	4.1
C	46.1	5.283	5.057	4.5
D	57	4.165	3.978	4.7
E	68.5	3.026	2.860	5.8
F	74.1	2.515	2.363	6.4

Ze Zhong C. Chen,
Zuomin Dong and
Geoffrey W. Vickers
*Most efficient tool feed
direction in three-axis CNC
machining*

Integrated Manufacturing
Systems
14/7 [2003] 554-566

Figure 13
Machined hemi-cylindrical part



geometry and cutting efficiency. The research introduces a new generic principle for tool path generation in the three-axis sculptured part machining, and this principle serves as a foundation for generating globally effective tool paths for complex sculptured surfaces.

References

- Bobrow, J.E. (1985), "NC machine tool path generation from CSG part representations", *Computer-Aided Design*, Vol. 17 No. 2, pp. 69-76.
- Broomhead, P. and Edkins, M. (1986), "Generation NC data at the machine tool for the manufacture of free-form surfaces", *International Journal of Production Research*, Vol. 24 No. 1, pp. 1-14.
- Chen, Z.C., Dong, Z. and Vickers, G.W. (2001), "Steepest-directed tool path in 3-axis CNC machining – the most efficient machining scheme and its mathematical proof", *Proceedings of the 2001 Design Technical Conference and Computers and Information in Engineering Conference*, ASME, DETC2001/CIE21301.
- Faux, I.D. and Pratt, M.J. (1979), *Computational Geometry for Design and Manufacture*, Ellis Horwood, Chichester.
- Huang, Y. and Oliver, J.H. (1994), "Non-constant parameter NC tool path generation on sculptured surfaces", *The International Journal of Advanced Manufacturing Technology*, Vol. 9, pp. 281-90.
- Loney, G.C. and Ozsoy, T.M. (1987), "NC machining of free form surfaces", *Computer-Aided Design*, Vol. 19 No. 2, pp. 85-90.
- Maeng, H., Ly, M. and Vickers, G.W. (1996), "Feature-based machining of curved surfaces using the steepest directed tree approach", *Journal of Manufacturing Systems*, Vol. 15 No. 6, pp. 1-13.
- Marsden, J. and Weinstein, A. (1985), *Calculus*, Springer-Verlag, New York, NY.
- Mortenson, M.E. (1987), *Geometric Modeling*, John Wiley & Sons Inc., New York, NY.
- Sarma, R. and Dutta, D. (1997), "The geometry and generation of NC tool paths", *Journal of Mechanical Design*, Vol. 119, pp. 253-8.
- Suresh, K. and Yang, D.C.H. (1994), "Constant scallop height machining of free-form surfaces", *Journal of Engineering for Industry*, Vol. 116, pp. 253-9.

Financial support from the Natural Science and Engineering Research Council (NSERC) of Canada and the University of Victoria Scholarship are gratefully acknowledged.

Research on the process model of product development with uncertainty based on activity overlapping

Renbin Xiao

CAD Center, Huazhong University of Science and Technology, Wuhan, P.R. China

Shangwen Si

CAD Center, Huazhong University of Science and Technology, Wuhan, P.R. China

Keywords

Product development,
Time to market,
Production planning and control

Abstract

The essence of competition among modern enterprises is time; hence the time for product development must be greatly reduced. This paper presents a new process model of product development with uncertainty based on activity overlapping, borrowing ideas from the uncertainty model and activity overlapping model proposed by Loch and Terwiesch, and Krishnan respectively, to realize reduction on the time. Besides the fundamental framework of the proposed model, some derivations on the formula computing the total execution time of upstream and downstream activities are made. The effectiveness of the proposed model is verified through some further discussions and initial computational results.

Introduction

Product development is presently faced with most challenging environments due to fierce competition in the market. At the initial stage of industrialization, competitiveness mainly lies with the price of products. Only if the products were cheap and usable, would they be of competitive advantage in the market. Since the price of products is determined by the cost, this type of competition is called cost-based competition. With the improvement in people's livelihood, quality as well as service turned up trumps, which led to the competition being quality-based. Since the 1980s, uncertainty of business circumstances has increased and the enterprise competition tends to be more drastic, when competitiveness depends greatly on the time as well as variety of products. Only those who respond to the market changes rapidly would occupy the larger market share. Thus the pattern of competition turned out to be time-based.

To succeed in the time-based competition, it is necessary to shorten the time of product development greatly; hence, some new concepts, theories and technologies relevant to the issue emerged, like concurrent engineering and time compression. Regarding the product development process, this paper proposed a new time compression method via a modelling approach, which is based on the idea of concurrent engineering, conforms with the current market with its uncertainty and the demand for quick response to market changes for modern enterprises; accordingly, the research conducted in this paper can be of effective support for the management of complicated processes of product development.

Analysis of activities in product development

Description of information relations in product development

Many models have appeared describing the process of product development, where there are a lot of species based on the design structure matrix (DSM) (Smith and Morrow, 1999). The concept of DSM originates from the systemic design (Steward, 1981), and it can also be used to analyze the process of product development that has a wider overlay. In DSM, each row and its corresponding column are identified with one of the tasks or activities. Along each row, the marks indicate other activities that are necessary to fulfil the activity represented by this row, and a certain column indicates the other activities that receive information from the activities represented by this column (Ulrich and Eppinger, 2000). Diagonal elements do not convey any meaning at this point, since an activity cannot depend on itself. Thus, DSM can be utilized to describe the information relations in product development activities.

From the manner of information connection, there are three connection forms among activities in product development described by DSM, viz. serial, parallel, and couple (Eppinger, 1994), as shown in Figure 1.

Analysis of the connection forms among activities

The three connection forms among activities in Figure 1 can be analyzed as follows:

- 1 There exists no information connection among the parallel or independent activities, which is an ideal situation. Either the situation depicts the very simple activities in the objective world, or divides the complicated connection among the real activities so as to produce



Integrated Manufacturing
Systems
14/7 [2003] 567-574

© MCB UP Limited
[ISSN 0957-6061]
[DOI 10.1108/09576060310491928]

The Emerald Research Register for this journal is available at
<http://www.emeraldinsight.com/researchregister>



The current issue and full text archive of this journal is available at
<http://www.emeraldinsight.com/0957-6061.htm>

the man-made “concurrent engineering”. In fact, the essence of concurrent engineering is to seek the parallel processing of activities with the dependent relations of information. Therefore, such a form is in fact a distorted reflection of the relation among complicated activities, whereas it just displays itself as “parallel” formally. It is of no practical meaning to the complete process of complicated product development.

- 2 The connection form that is mutually coupled reflects the complicated relations among activities in the objective world, which has been researched quite a lot by overseas scholars (e.g. Smith and Eppinger (1997a, b)). Owing to the information coupling among the activities, viz. activity A requires the information of B and activity B requires the information of A (see Figure 1c), the circular relation forms. For problem solving, the relations among activities are required to be divided artificially according to a certain principle and the optimized iterative sequence to be found, and then the problem of convergence of the iterative process should be studied in a further step. In view of the extensive research on this issue, no further discussion will be given in this paper.
- 3 There exist simple one-way connections among the serially dependent activities. This kind of connection can be easily disposed of; however, it will result in a time delay for product development. In order to shorten the total time for the execution of activities, the preliminary information of the activity being executed can be utilized to implement its partial overlapping with the forthcoming activities. The former is usually called upstream activity, and the latter as downstream activity. For a relatively pure connection form of serial dependence, the result of time compression of product development can be obtained when adopting proper strategies and through overlapping of upstream and downstream activities, which truly embodies the idea

of the concurrent engineering, and the research in this paper belongs to this class. In the following part, the motivation of research is given based on reviewing the main research work on activity overlapping.

Review and analysis of the research on activity overlapping

Krishnan *et al.* (1995, 1997) and Krishnan (1996) completed the fundamental framework of research on activities overlapping, which is the basis of almost all of the later relevant works. The function of uncertainty on the product development process has been generally studied in the quite influential work of Loch and Terwiesch (1998). In the following part, the activity-overlapping model of Krishnan and the uncertainty analysis of Loch and Terwiesch (1998) based on Krishnan’s work are first introduced. Then the actual meaning of the model derived from merging of the above two approaches is explained, which makes it natural to elicit the work of modeling in the next section.

The model of activity overlapping

Two important concepts are introduced by Krishnan when discussing the information relations among the upstream and downstream activities, viz. the evolution of information in the upstream activity and the degree of sensitivity of downstream activity. The former refers to speed of exchange information from the upstream to the downstream approaching the final form, and the latter refers to the magnitude of reaction yielded when exchanged information changes. Viewed qualitatively, the information evolution speed of upstream activities differs greatly and the sensitivity degree of those downstream is also not the same. Through combination of the two facts, four typical patterns form, where one kind of pattern has a low sensitivity degree of downstream activity and the slow speed of information evolution of the upstream one. The preliminary information of upstream can be utilized early to trigger downstream activities. However, the information evolution of upstream activity is slow, leading to the fact that downstream activities must be iterated to be convergent to the final results. This pattern is emphasized by Krishnan and named the iterative overlapping problem (IOP).

The model of IOP is shown in Figure 2, where t_{As} is the start time of the upstream activity A, usually assigned with 0; and t_{Af} is the end time of A. d_i represents the i th ($i=0, 1, \dots, n$) execution time of the downstream activity. When A and B serially execute this

Figure 1

Three connection forms among activities

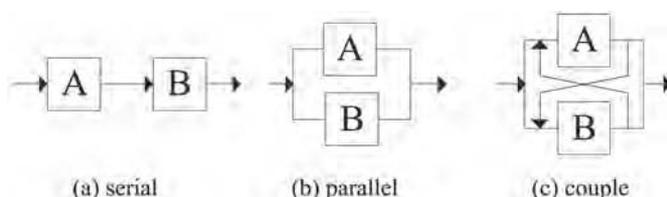
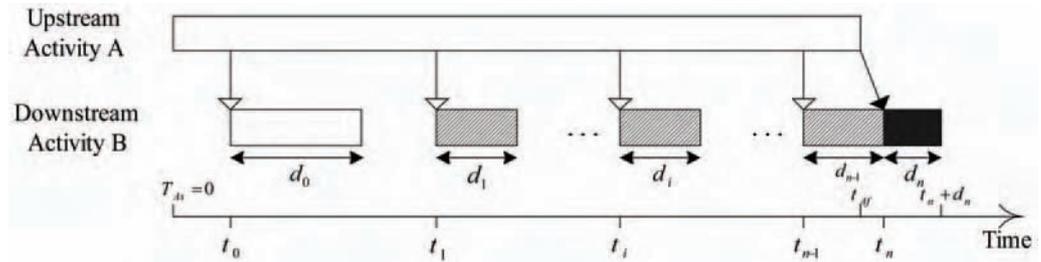


Figure 2
 The model of IOP



extreme case whose overlapping degree is 0, none of the iterations where $i \geq 1$ exists except where $i = 0$; when the degree of overlapping is not 0, some iterations exist where $i \geq 1$ besides where $i = 0$; therefore, whatever the case, $d_0 > 0$, which is called planned iteration. Then, n is the iterative times after the planned iteration.

In the IOP model, t_i is the start time of the i th iteration, $T_{\text{sum}} = t_n + d_n$ is the total execution time of upstream and downstream activities. The number of information exchanges is $n + 1$, which are finished instantaneously, and the following expressions are true:

$$t_i \geq t_{i-1} + d_{i-1}, \quad (1)$$

$$t_0 \geq t_{As}, \quad (2)$$

$$t_{n-1} < t_{Af}, \quad (3)$$

$$t_n \geq t_{Af}. \quad (4)$$

The consideration of uncertainty

Based on Krishnan's framework, Loch and Terwiesch argued that the information of the upstream activity is variable; hence the arrival of the exchanged information of the upstream activity A can be supposed to be a non-homogeneous Poisson stochastic process, the rate of which is $\mu(t_A)$. Loch and Terwiesch (1998) proposed the relevant mathematical expression in a further step. On the other hand, after the arrival of the exchanged information of the upstream activity, the downstream activity will respond on the exchanged information through certain rework. Since there exists time consumption of information commutation between the upstream and downstream activities, Loch and Tierwiesch thought that it is unsuitable to trigger information commutation due to the downstream reaction on the arrival of the commuting information of the upstream activity at any moment, whereas it is better to adopt an appropriate batching policy. However, Loch and Terwiesch (1998) only

considered the one shot-batching instance, which is not consistent with the real facts.

Analysis

From the introduction of "The model of activity overlapping" and "The consideration of uncertainty", we know both the merits and drawbacks of Krishnan and Loch and Tierwiesch's works respectively. The former considers the repetitiveness and instantaneousness of communication that consequently trigger the iteration of the downstream activity, while it disregards information uncertainty. The latter considers the uncertainty of information arrival from upstream activity, which satisfies the requirement of complicated product development presently with uncertainties, and is therefore much closer to reality. However, it only considers the information commutation of upstream and downstream activity one time, which is accumulative due to the batch-processing operation; hence the iteration of downstream activity is only one time. Viewed from this point, the multi-iterative model proposed by Krishnan is, no doubt, more reasonable. It is obvious that the combination of the two complementary models has more advantages and is of more practical significance. This is the motivation for creating models corresponding to the product development process with uncertainties based on activity overlapping.

The uncertainty model of product development process based on activity overlapping

The model presented by this paper is based on the models proposed by Krishnan and by Loch and Terwiesch, wherein Krishnan's model has been introduced in detail in "The model of activity overlapping". Therefore, the brief introduction of Loch and Terwiesch's uncertainty model based on activity overlapping will be given first in this section, and then we propose our model and its deductive process.

The uncertainty model based on activity overlapping

Information arrival model of revised upstream activities

According to the above discussion, the arrival of the exchanged information of the upstream activity A can be supposed to be a non-homogeneous Poisson stochastic process whose intensity is $\mu(t_A)$. Loch and Terwiesch (1998) introduced a model of the information arrival of such stochastic process, adopting the linear function:

$$\mu(t_A) = \mu \left[1 + e \left(2 \frac{t_A}{T_A} - 1 \right) \right], \quad (5)$$

where T_A is the executing time of the upstream activity and t_A is limited within $[0, T_A]$, e is a parameter reflecting the degree of the evolution of the upstream activity, where $e < 0$ corresponds to the fast evolution and $e > 0$ corresponds to the slow evolution; hence e is named as the evolution degree which is limited within $[-1, 1]$.

In Equation (5), μ is the average Poisson intensity. Based on some exponential results, Loch and Terwiesch thought that μ is determined by the following formula:

$$\mu = \mu_0 \exp \{-B\alpha\}, \quad (6)$$

where α is the total amount of information exchange before start-up, parameter B represents the coordination capability of the development group, μ_0 is the inherent technical uncertainty of the project.

The rework model of the downstream activity

The rework time of the downstream activity depends on the degree of the downstream activity reacting on the exchanged information and the evolutionary status of the downstream activity. Therefore, Loch and Terwiesch define the influencing function $f(t)$ that denotes the rework time of the downstream activity. The more downstream has progressed in its work, the more cumulative work must be modified; hence $f(t)$ is a non-decreasing function. In practice, $f(t)$ might be concave or convex. Adopting the linear function, Loch and Terwiesch (1998) gave the expression $f(t)$ as follows in order to make it simple:

$$f(t) = kt, \quad (7)$$

where k is the sensitive degree of the downstream activity that is usually limited within $[0.01, 0.09]$.

The model proposed in this paper

Fundamental framework

In the model proposed in this paper, the times for information exchange between upstream and downstream activities, viz. $n + 1$ times

iteration in all, are consistent with that of Krishnan's model and, since each iteration is accumulative, which is in accord with that of Loch and Terwiesch's model, the iteration time of the downstream activity is $n + 1$ too. A fragment of overlapping between upstream and downstream activities is given in Figure 3, which covers iteration for two times, where Figure 3(a) denotes the physical structure reflecting real states and Figure 3(b) gives the logical structure equivalent to the former one.

It can be obtained from Figure 3(a) that the i th iterative time of the downstream includes the following three parts:

- 1 The execution time d_i of iteration during the delivery period of upstream information, wherein d_0 corresponds to the planning iteration.
- 2 The cost of time c_i when exchanging information between upstream and downstream activities.
- 3 The rework time r_i after information exchange. The model proposed in this paper is more comprehensive than Krishnan's model that considers only d_i , and Loch and Terwiesch's model that only takes c_i and r_i into account.

To utilize Krishnan's model conveniently, activity A in Figure 2 should be in the state of information processing all the time. Then the relevant T_A cannot include c_i , viz. c_i exists physically instead of logically, which is not the same as for the downstream activity B. Suppose T_B to be the executive time of activity B, T_B will include c_i , viz. c_i exists not only physically but also logically. The reason can be accessed from Figure 2 where activity B admits the existence of some idle time sections. The whole of the following discussion is based on the logical structure (in Figure 3b). First, let t_{Bs} , t_{Bf} be the initial and terminative time of activity B respectively, T_0 be the overlapping time of activity A and B, λ be the overlapping rate, then we can obtain the following formulae:

$$T_B = t_{Bf} - t_{Bs} = t_{Bf} - t_0, \quad (8)$$

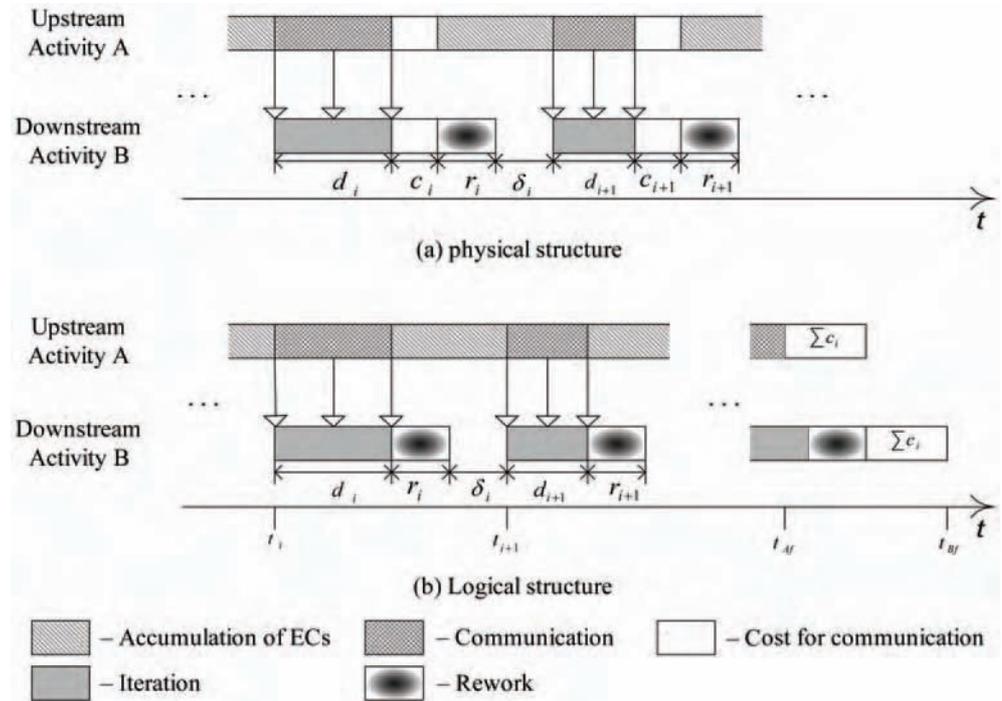
$$\lambda = \frac{T_0}{T_A}, \quad (9)$$

$$t_0 = t_{Af} - T_0 = T_A - \lambda T_A = (1 - \lambda)T_A. \quad (10)$$

The amendment to Loch and Terwiesch's model

Two points are modified according to the practical situation in the model proposed in this paper, although the model of the upstream information arrival introduced in "Information arrival model of revised upstream activities" has been taken into consideration.

Figure 3
 Twice iterations in activity overlapping



Amendment to Equation (5)
 Since $T_A = t_{Af} - t_{As} = t_{Af}$, Equation (5),
 alternatively, can be written as:

$$\mu(t_A) = \mu \left[1 + e \left(2 \frac{t_A}{t_{Af}} - 1 \right) \right], \quad (11)$$

where $0 \leq t_A \leq t_{Af}$.
 Suppose t_F to be the termination time of
 the upstream information, t_F is nearly
 correlative to the value of e . If upstream
 evolves slowly, the final form of
 exchanged information cannot be obtained
 once the upstream activity is terminated,
 where $t_F = t_{Af}$ at that time. If upstream
 evolves fast, the exchanged information can
 get to the final form ahead where $t_F < t_{Af}$.
 There is no information arrival from t_F to
 t_{Af} for the latter instance, which is not
 rational because $\mu(t_A)$ is not equal to 0
 according to Equations (5) or (11). Then
 there is the revision that substitutes t_{Af}
 with t_A ; therefore the model of the upstream
 information arrival that we adopt
 practically is the following:

$$\mu(t_A) = \mu \left[1 + e \left(2 \frac{t_A}{t_F} - 1 \right) \right]. \quad (12)$$

Amendment to Equation (6)
 Our model has the repetitive iterations,
 whereas Loch and Terwiesch's model
 considers only one iteration of the

downstream activity. Therefore, Equation (6)
 should be amended as follows:

$$\mu = \mu_0 \exp \{-B\alpha_i\}, \quad (13)$$

where α_i is the total amount of the exchanged
 information when the i th iteration proceeds.
 $\alpha_0 = \alpha$, while α_i can be obtained from the
 following formula:

$$\alpha_i = \alpha_{i-1} + \int_{t_{i-1}}^{t_{i-1}+d_{i-1}} \mu(t) dt, \quad (14)$$

where $\int_{t_{i-1}}^{t_{i-1}+d_{i-1}} \mu(t) dt$ is the amount of the
 revised information arrived at the $(i-1)$ th
 iteration from the upstream activity.

The derivation of T_{sum}
 T_{sum} refers to the total project execution time
 including upstream and downstream
 activities.

Considering the $(i+1)$ th iteration of the
 overlapping of A and B that correspond to
 the i th iteration; since i is calculated starting
 from 0, according to Cho and Eppinger (2001):

$$d_i = d_0 \phi_B^i, \quad (15)$$

where ϕ_B^i is the decreasing ratio of the
 execution time at the i th iteration of the
 downstream activity and it satisfies the
 following formula:

$$\lim_{i \rightarrow \infty} \phi_B^i = 0.5. \quad (16)$$

According to Loch and Terwiesch (1998):

$$c_i = \int_{t_i}^{t_i+d_i} \sqrt{\frac{k\mu(t)\tau}{2}} dt, \quad (17)$$

where τ is the coefficient of communication cost.

The rework at the i th iteration is $\int_{t_i}^{t_i+d_i} k\mu(t)(t-t_i)dt$; disregarding communication cost and the additional rework at the i th iteration, this produces:

$$\int_{t_i}^{t_i+d_i} \left(\sqrt{\frac{k\mu(t)\tau}{2}} - \frac{k}{2} \right) dt,$$

caused by time cost when exchanging information; thus the sum of the above two is:

$$r_i = \int_{t_i}^{t_i+d_i} \left(k\mu(t)(t-t_i) + \sqrt{\frac{k\mu(t)\tau}{2}} - \frac{k}{2} \right) dt. \quad (18)$$

If δ_i is the time interval between the i th and $(i+1)$ th iteration of the downstream activity, then:

$$t_{i+1} = t_i + d_i + r_i + \delta_i. \quad (19)$$

When the upstream activity is approaching the end, there exist two cases (see Figure 4).

In Figure 4(a), the last execution of downstream iteration, which started before t_F , is finished before t_F , whose rework of the iteration is ended at t_F or later, when there still exists certain accumulative information of the upstream activity corresponding to the interval of $t_F - t_{n-1} - d_{n-1}$ in Figure 4(a), which has not been passed to the downstream

activity. Therefore, a downstream iteration should be triggered, where there are only iterative execution and no rework. Thus t_{Bf} , the end time for downstream activity, can be given as:

$$t_{Bf} = t_n + d_n + \sum_{i=0}^{n-1} c_i. \quad (20)$$

In Figure 4(b), the last downstream iterative execution starting before t_F is accomplished at t_F or after that. In such a situation, all the upstream information has been transferred. So the new iteration is not required to trigger and t_{Bf} can be calculated by the following formula:

$$t_{Bf} = t_n + d_n + r_n + \sum_{i=0}^n c_i, \quad (21)$$

where:

$$r_n = \int_{t_n}^{t_F} \left(k\mu(t)t + \sqrt{\frac{k\mu(t)\tau}{2}} - \frac{k}{2} \right) dt,$$

$$c_n = \int_{t_n}^{t_F} \sqrt{\frac{k\mu(t)\tau}{2}} dt.$$

So far, the calculation formula of T_{sum} can be given as follows.

For the first case:

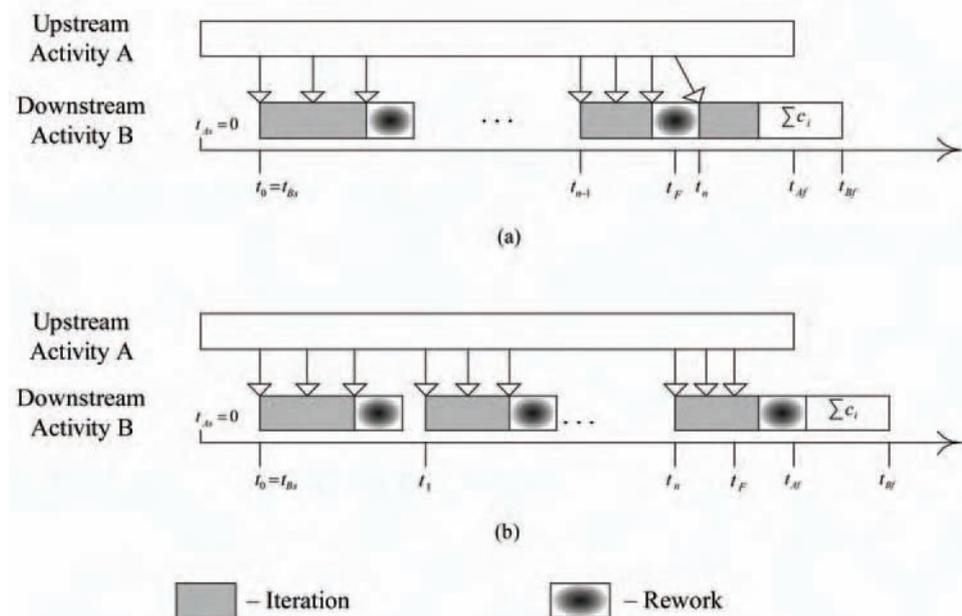
$$T_{sum} = \max \left\{ T_A + \sum_{n=0}^{n-1} c_i, t_{Bf} \right\}. \quad (22)$$

For the second case:

$$T_{sum} = \max \left\{ T_A + \sum_{i=0}^n c_i, t_{Bf} \right\}. \quad (23)$$

Figure 4

Two cases approaching the terminal of upstream activity



Discussion

The expression of t_F

According to the above discussion, t_F is closely relevant to the evolution degree e . Obviously, $t_F = t_{Af}$ when the evolution is the slowest that $e = 1$ and $t_F < t_{Af}$ when the evolution is the fastest that $e < 0$. Here r_A , an information terminative coefficient of the upstream A , is introduced as follows, which denotes the advance ratio of the upstream information termination when $e = 0$:

$$r_A = \frac{t_{Af} - t_F}{t_{Af}} \quad (24)$$

From Equation (24), we can obtain that:

$$t_{Af} - t_F = r_A t_{Af} \quad (25)$$

$t_{Af} - t_f = 0$ when $e = 1$. Moreover, smaller e indicates faster evolution; thus smaller t_F and then larger $t_{Af} - t_F$. So $t_{Af} - t_F$ is the decreasing function of e .

Based on the above points, a simple expression of $t_{Af} - t_F$ can be constructed as follows:

$$t_{Af} - t_f = (1 - e)r_A t_{Af} \quad (26)$$

From the Equation (26), t_F can be yielded as:

$$t_F = [1 - (1 - e)r_A]t_{Af} \quad (27)$$

The preliminary calculation results and analysis

The essential purpose of model construction in the uncertainty model of product development process based on activity overlapping is to calculate T_{sum} , so as to quantify the effect of time compression. But there are too many influence parameters in the proposed model, each of which has many values. Owing to the length limitation of the present paper, the comprehensive calculation, analysis and discussion will be given in another paper. Here we just list the calculation results with one kind of parameter combination in order to verify the effectiveness of the proposed model.

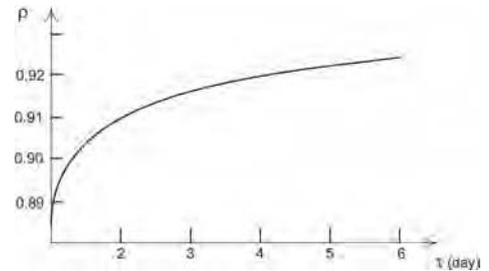
To make T_{sum} non-dimensional, the coefficient of time compression is introduced and defined as follows:

$$\rho = \frac{T_{sum}}{T_A + d_0} \quad (28)$$

According to the proposed model in this paper, the relation between time consumption of the information exchange and ρ is shown in Figure 5 and the relation between the times of the advance information commutation and ρ is shown in Figure 6, which results from the calculation. The parameters adopted in the calculation are: $T_A = 20$ weeks, $d_0 = 5$ weeks, $e = 1$, $k = 0.01$,

Figure 5

The relation between time consumption of the information exchange and ρ



$\lambda = 0.94$, $n = 4$, $r_A = 0.01$, $\mu_0 = 1$, $B = 0.1$ and $\alpha = 0$ in Figure 5, $\tau = 3$ days in Figure 6.

In Figure 5, the total time to execute the upstream and downstream activities may increase with the augmentation of the communication costs. It can be derived from Figure 6 that the total executing time may decrease correspondingly with the increase of time for the advance information commutation before the project starts. Such results accord with our intuition.

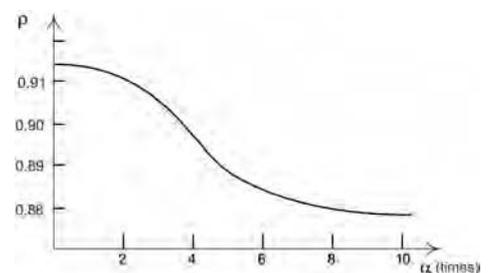
Both the results in Figures 5 and 6 satisfy $\rho < 1$, which shows that the effect of time compression is achieved by utilizing the model in this paper.

Conclusion

In view of the complicated and non-deterministic environments of current markets and the requirement of rapid response of modern enterprise to market changes, a kind of process model of product development with uncertainty based on activity overlapping is proposed in this paper, adopting the idea of concurrent engineering and taking the work of Krishnan and of Loch and Terwiesch as references. The proposed model integrates the merits of both Krishnan's and Loch and Terwiesch's, and the influence of many factors is generally considered, which makes it closer to the

Figure 6

The relation between the times of the advance information commutation and ρ



actual process of the complex product development. The effectiveness of the proposed model is verified through preliminary calculation and analysis. The work of this paper is of great importance in mastering in principle the inherent regularities of a complex product development process and for effective compression of development time.

References

- Cho, S.H. and Eppinger, S.D. (2001), "Product development process modeling using advanced simulation", *Proceedings of 2001 ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference*, DETC2001/DTM-21691.
- Eppinger, S.D. (1994), "A model-based method for organizing tasks in product development", *Research in Engineering Design*, Vol. 6 No. 1, pp. 1-13.
- Krishnan, V. (1996), "Managing the simultaneous execution of coupled phases in circumvent product development", *IEEE Transactions on Engineering Management*, Vol. EM-43 No. 2, pp. 210-17.
- Krishnan, V., Eppinger, S.D. and Whitney D.E. (1995), "Accelerating product development by the exchange of preliminary product information", *ASME Journal of Mechanical Design*, Vol. 117 No. 4, pp. 491-8.
- Krishnan, V., Eppinger, S.D. and Whitney, D.E. (1997), "A model-based framework to overlap product development activities", *Management Science*, Vol. 43 No. 4, pp. 437-551.
- Loch, C.H. and Terwiesch, C. (1998), "Communication and uncertainty in concurrent engineering", *Management Science*, Vol. 44 No. 8, pp. 1032-48.
- Smith, R.P. and Eppinger, S.D. (1997a), "A predictive model of sequential iteration in engineering design", *Management Science*, Vol. 43 No. 8, pp. 1104-20.
- Smith, R.P. and Eppinger S.D. (1997b), "Identifying controlling features of engineering design iteration", *Management Science*, Vol. 43 No. 3, pp. 276-93.
- Smith, R.P. and Morrow, J.A. (1999), "Product development process modeling", *Design Studies*, Vol. 20 No. 3, pp. 237-61.
- Steward, D.V. (1981), "The design structure system: a method for managing the design of complex systems", *IEEE Transactions on Engineering Management*, Vol. EM-28 No. 3, pp. 71-4.
- Ulrich, K.T. and Eppinger, S.D. (2000), *Product Design and Development*, McGraw-Hill, Boston, MA.

GA-driven part e-manufacturing scheduling via an online e-service platform

Yingfeng Zhang

CAD/CAM Institute, School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, P.R. China

Pingyu Jiang

CAD/CAM Institute, School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, P.R. China

Guanghui Zhou

CAD/CAM Institute, School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, P.R. China

Keywords

Manufacturing systems, Production scheduling, Bills of materials, Programming and algorithm theory, Computer aided manufacturing

Abstract

This paper is mainly concerned with studying the e-manufacturing scheduling issue based on the bills of materials (BOM) flows of products. On the basis of e-manufacturing philosophy, a genetic algorithm (GA)-based scheduling model for physical e-manufacturing cells is put forward. By means of building the mapping relationship and dynamic task association among physical e-manufacturing cells, logic e-manufacturing cells, e-manufacturing systems and e-manufacturing environment, high-level scheduling issues can be transformed as an inductive procedure concerning the scheduling results to corresponding physical e-manufacturing cells. As verification, finally, a case study is given to demonstrate the method mentioned above.

1. Introduction

Currently, the rapid progress of information and network techniques is impelling the globalization of manufacturing activities. In order to win competition in global markets, enterprises have to innovate their manufacturing modes. E-manufacturing mode, as a novel manufacturing mode, has attracted profound interest in academic and industrial areas in recent years. Here, e-manufacturing can be seen as a kind of digital, networked and e-commerce-driven manufacturing technology that is formed by introducing e-technologies to current manufacturing environment with legacy and new-developed resources.

As one of the key enabling technologies for e-manufacturing, the scheduling mechanism and correspondent algorithms for e-manufacturing systems play a key role in making the systems work well. Compared with the legacy job-shop scheduling mechanism/algorithms, they have several new attributes that can be described in detail as follows:

- *Online attribute.* It is because equipment that is used to form the e-manufacturing systems has been reconstructed by means of adopting "e-technologies" in advance. Such reconstructions deal with building up the reconfigurable, new-type equipment whose front-sides are installed as one or a group of Web service mechanisms, organizing them together through the Internet, and then providing e-services in terms of using them. Therefore, the scheduling activities for e-manufacturing cells and e-manufacturing equipment corresponding to an e-manufacturing system must be performed in online mode.

- *Collaboration.* Because an e-manufacturing system is formed dynamically, and e-manufacturing cells used to form the e-manufacturing system have dynamical, agile and distributive attributes too, the scheduling information among the e-manufacturing system, e-manufacturing cells and e-manufacturing equipment must be produced in real time in the form of collaboration. Such collaboration pertains to the people who are participating in the corresponding manufacturing activities.
- *Dynamical attribute.* As we know, since the e-manufacturing cells are dynamically created based on specific manufacturing tasks, the scheduling mechanism for these e-manufacturing cells also has dynamical attributes.

According to the above new features of the scheduling mechanism/algorithms for e-manufacturing systems, this paper first adopts genetic algorithm (GA) to solve the problem of scheduling the e-manufacturing cells based on specific manufacturing tasks, and then the scheduling outcomes for the e-manufacturing system that consists of several e-manufacturing cells or for the whole e-manufacturing environment that is composed of different e-manufacturing systems can be exported on the basis of the scheduling results of the e-manufacturing cells.

The rest of this paper is organized as follows. Section 2 presents a brief review of the literature related to scheduling manufacturing systems/cells/job-shops. Section 3 provides some basic definitions used in this paper. In Section 4, a reference model for scheduling the part machining in e-manufacturing environment is put forward. An actual scheduling model for part



Integrated Manufacturing Systems
14/7 [2003] 575-585

© MCB UP Limited
[ISSN 0957-6061]
[DOI 10.1108/09576060310491937]

The Emerald Research Register for this journal is available at
<http://www.emeraldinsight.com/researchregister>



The current issue and full text archive of this journal is available at
<http://www.emeraldinsight.com/0957-6061.htm>

machining in e-manufacturing cells is described in Section 5 and its GA implementation is given in Section 6. A case study is demonstrated in Section 7, which is followed by concluding remarks in Section 8.

2. Literature review

The job-shop-scheduling problem has been known as a strongly NP-complete problem for many years. Recently, great efforts have been applied and some solutions have emerged to solve this problem. For example, heuristic arithmetic (He *et al.*, 1996), various neural network algorithms (Yang and Wang, 2002), genetic algorithm (GA) and so on. Davis was the first to study such a problem using GA in 1985 (Davis, 1985). Since then, a number of successful applications using GA for job-shop-scheduling problems have appeared (Yamada and Nakano, 1992; Watanabe *et al.*, 1995; Maturana *et al.*, 1997; Lawrence and Sewell, 1997; Lee and Dagli, 1997; Chenga *et al.*, 1999; Cardon *et al.*, 2000). Sakawa and Mori (1999) considered the imprecise or fuzzy nature of the data in real-world problems, and proposed an efficient genetic algorithm for job-shop-scheduling problems with fuzzy processing time and fuzzy due-date. Yao *et al.* (2001) presented a comparison approach of genetic algorithms based on the multi-reproductions, and the results of calculation indicated that this approach had a fast speed of convergence. Yun (2002) demonstrated a GA with fuzzy logic controller for preemptive and non-preemptive job-shop-scheduling problems in 2002 (Yun, 2002). Naturally, in these job-shop-scheduling problems, the research objects are relatively limited at a job-shop.

However, in the current manufacturing environment, the manufacturing activities take place in many distributed job-shops, which make the scheduling model more complex. In this environment, the scheduling operations must be provided with online attribute, collaboration and dynamical attribute. Nobuaki and Masaaki (1997) put forward a generic framework for an online scheduling and control system in batch process management. They described the needs and advantages of a genetic framework for online scheduling and proposed the framework for an online scheduling and control system (FOSS), which provided a basic structure and mechanisms of FOSS in detail as well as its implementation issues in a practical environment. Jürgen *et al.* (1998) presented an approach to global scheduling that is concerned with heuristic scheduling

strategies and fuzzy concepts. In their research, a multi-site scheduling system that can resolve scheduling activities in different plants was designed.

Under these circumstances, in this paper, we bring forward a set of scheduling models for part machining in e-manufacturing environment, e-manufacturing systems and e-manufacturing cells, and design a physical e-manufacturing cell-scheduling algorithm that can provide the task-scheduling service for distributing users by taking advantage of the Internet and Web-based process-planning database.

3. Definitions of basic concepts

Before describing the scheduling models for e-manufacturing environment/systems/cells, we have to define the following concepts used in this paper:

- *E-manufacturing environment (e-ME)* is defined as a set of current existing e-manufacturing systems.
- *E-manufacturing system (e-MS)* is used for performing manufacturing tasks related to a kind of special product or a product family. It is formed based on clients' dynamical manufacturing tasks. According to the BOM (bill of material) of this specific product or product family, a set of logic e-manufacturing cells that can finish the corresponding manufacturing tasks may be chosen to construct an e-manufacturing system in terms of adopting some selecting mechanism, e.g. bidding model.
- *Physical e-manufacturing cell (e-PMC)* mainly refers to the combination of concrete equipment that has the concrete manufacturing capabilities. It is a basic managing and executing unit during the manufacturing process in an e-manufacturing environment and has the concrete or fixed geographical site.
- *Logic e-manufacturing cell (e-LMC)* is a basic element for establishing an e-manufacturing system. Through selecting the one, several or all items of equipment in a physical e-manufacturing cell, which correspond to some of those special manufacturing tasks, a logic e-manufacturing cell can be formed dynamically, which bridges the gap between e-manufacturing systems and physical e-manufacturing cells. In other words, a logic e-manufacturing cell can be taken for a kind of existent mode of a physical e-manufacturing cell in an e-manufacturing system.

- *E-equipment* is connected with network-centric equipment such as new-style CNC equipments or the legacy equipment that has been reconstructed by adopting “e-” technologies, e.g. using network-centric interfaces to encapsulate the legacy equipments.
- *Manufacturing tasks* are defined as a series of part-based machining activities that may consist of some different sets of processes. Here, assembling tasks are included only for making the problem solving simple.
- *Process* is defined as a series of machining operations or steps that are performed only in an e-equipment. In general, it is generated with the process-planning system.

Based on the above concepts, Figure 1 illustrates a logic framework for an e-ME. From this logic framework, we can definitely find out the relationship among e-ME, e-MSs, e-PMCs and e-LMCs. An e-ME consists of a number of e-PMCs that are owned by different manufacturers. Based on different kinds of part-based manufacturing tasks submitted by clients, the correspondent e-MSs are established dynamically. In a specific e-MS, the specific manufacturing tasks are divided into a series of sub-manufacturing tasks. Through adopting a bidding method, some of or all the items of

equipment that belong to e-PMCs and have the capabilities requested for finishing the corresponding sub-manufacturing tasks are selected out, so as to form a series of e-LMCs dynamically. Through scheduling and controlling e-PMCs, the correspondent e-LMCs are scheduled and controlled and, further, the corresponding manufacturing tasks are performed efficiently and effectively.

4. Scheduling model for part fabrication in e-ME

As we know, the precondition and groundwork of research on systemic methodologies for e-manufacturing scheduling are to deeply analyze its task-based scheduling model. The methodology and framework for constructing and running e-MSs via an online e-service platform can be found in the literature (Jiang *et al.*, 2002). Here, from the viewpoint of task-based scheduling, we only focus on the key points of e-manufacturing scheduling.

Figure 2 shows a task-based scheduling framework for e-MSs in an e-ME. From this Figure, we can see that the core enterprise in an enterprise federation can decompose the manufacturing tasks submitted by clients into three categories, that is, outsourcing, in-house, and out-house machining tasks

Figure 1
A logic framework for an e-ME

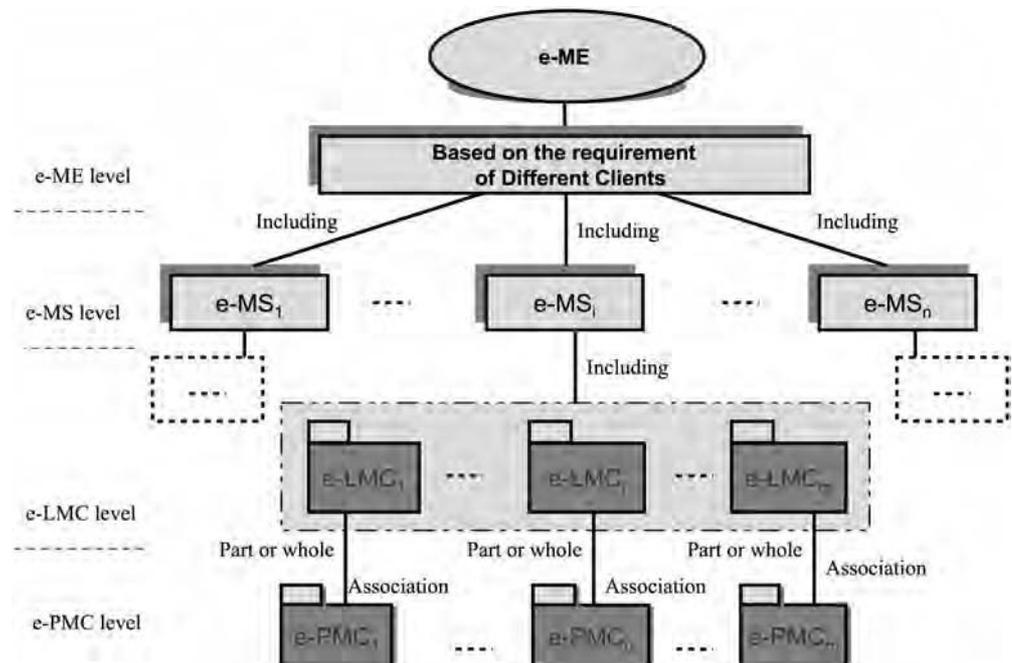
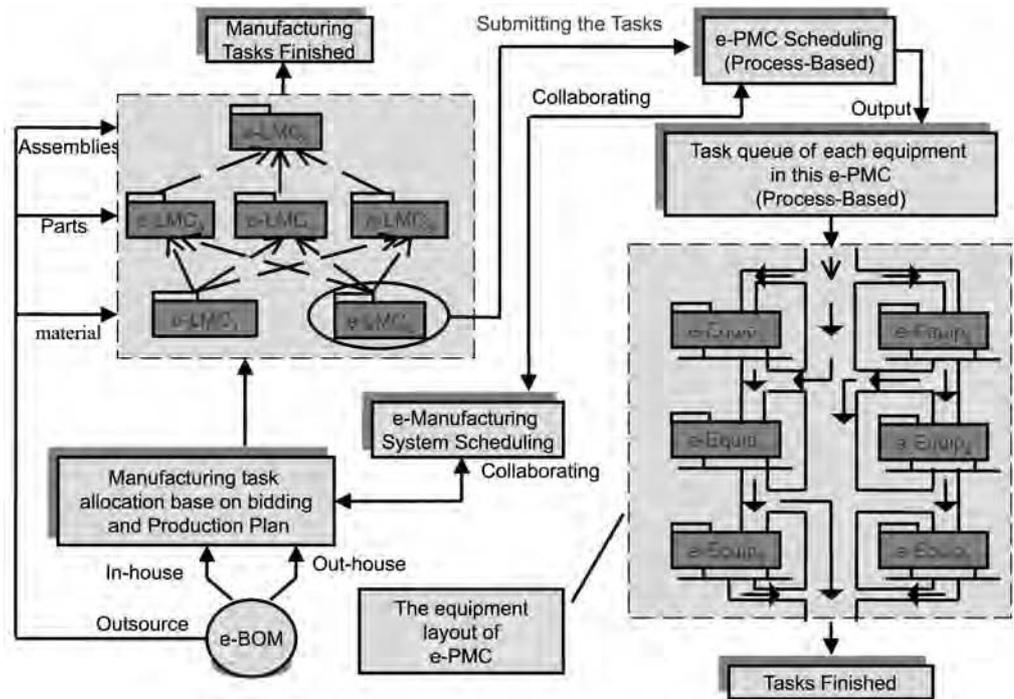


Figure 2
A task-based scheduling framework for e-manufacturing systems



based on the product BOM (bill of materials). Here, outsourcing parts correspond to suppliers; out-house machining parts are dispatched to the geographical distributive e-LMCs of other relative enterprises in the federation by the core enterprise based on the bidding mechanism; in-house parts are machined in the e-LMCs of the core enterprise. So, according to the above decomposition, we know that e-PMCs corresponding to e-LMCs are the actual workplaces to perform the corresponding manufacturing tasks. How to realize the task allocation in e-PMCs under the condition of satisfying the correspondent constraints is becoming the key point for e-manufacturing scheduling. For the issues on scheduling e-LMCs, e-MSs, even e-ME as well, the corresponding outcomes can be induced from the scheduling results for e-PMCs by means of using the relationship shown in Figure 1.

5. Scheduling model for part fabrication in e-PMCs

In an e-manufacturing system, the different combinations of e-PMCs can be mapped as the job-shops in real society. The core of scheduling the e-PMCs is to take advantage of the manufacturing resources of the e-PMCs to sequence a series of manufacturing tasks, and further arrive at the objective of the minimal values of some objective function

such as minimal cost and minimal machining time. Here, as defined above, the manufacturing tasks that are posted into the e-PMCs only consist of parts. Because a process is a basic element for formatting the manufacturing tasks of parts, and the operations/steps of a process are generally finished on the same equipment, the scheduling object for e-manufacturing procedure can be focused on the process-oriented scheduling of parts.

Before describing the scheduling model for part machining in e-PMCs, the following notations (N1-N5) need to be introduced, first:

- N1. Let P be the set of machining tasks posted into an e-PMC; we can define $P = \{P_1, P_2, P_j, \dots, P_n\}$.
- N2. Suppose that an e-PMC consists of m e-equipments, this e-PMC can be seen as the set of these e-equipments, which is described as follows: $e\text{-PMC} = \{DEV_1, DEV_2, DEV_k, \dots, DEV_m\}$.
- N3. Let N_i be the number of processes of machining task P_i , and JP_i be the set of processes of P_i workpiece, the JP_i can be described as $JP_i = \{J_1, J_2, J_j, \dots, J_{N_i}\}, I = 1, 2, \dots, n$.
- N4. Let (P_i, J_i, DEV_k) be the j process of task P_i machined on the DEV_k equipment, and let the starting machining time of this process be $S(P_i, J_i, DEV_k)$ and machining time be $T(P_i, J_i, DEV_k)$.
- N5. Let D_i be the delivery time of task P_i .

Taking the minimum manufacturing cycle as the capability index of scheduling, the mathematical model for e-PMCs can be built up as follows:

$$\min E = \max S(P_i, J_j, DEV_k) + T(P_i, J_j, DEV_k)$$

$$i \in [1, n], j \in [1, N_i], k \in [1, m]. \quad (1)$$

$$\begin{aligned} S(P_i, J_{j+1}, DEV_a) - S(P_i, J_j, DEV_b) \\ \geq T(P_i, J_j, DEV_b) \end{aligned}$$

$$i \in [1, n], k \in [1, N_i - 1], a, b \in [1, m]. \quad (2)$$

$$\begin{aligned} S(P_x, J_a, DEV_k) - S(P_y, J_b, DEV_k) \\ \geq T(P_y, J_b, DEV_k), \end{aligned}$$

or:

$$\begin{aligned} S(P_y, J_b, DEV_k) - S(P_x, J_a, DEV_k) \\ \geq T(P_x, J_a, DEV_k) \end{aligned}$$

$$x, y \in [1, n], a \in [1, N_x], a \in [1, N_y], k \in [1, m]. \quad (3)$$

$$S(P_i, J_j, DEV_k) + T(P_i, J_j, DEV_k) \leq D_i$$

$$i \in [1, n], j = N_i, k \in [1, m]. \quad (4)$$

In the above equations, Equation (1) is the objective function, whose value is to take the maximal value referring to the finishing time of the last process of all manufacturing tasks. According to different scheduling results, this value is also changeable. If this maximal value is minimized, an optimized scheduling result concerning the minimal manufacturing cycle can be obtained. Equations (2) to (4) express the constraints. Here, Equation (2) describes the sequence constraint of processes. Under this constraint, different processes belonging to the same task cannot be machined at the same time. Equation (3) expresses the resource constraints, which mean that each equipment can only machine one manufacturing task at the same time. Equation (4) describes the constraint of delivery time.

6. GA implementation for scheduling model

As we have known, task-based scheduling has been recognized as a difficult NP-problem. But using the genetic algorithm (GA) to solve this kind of task-based scheduling NP-problem just displays the stronger advantage such as quick stypic rate, optimal resolution. Based on the feature of GA, we design and develop a scheduling

algorithm and model for scheduling an e-PMC.

6.1 Principle of GA-based scheduling model for e-PMC

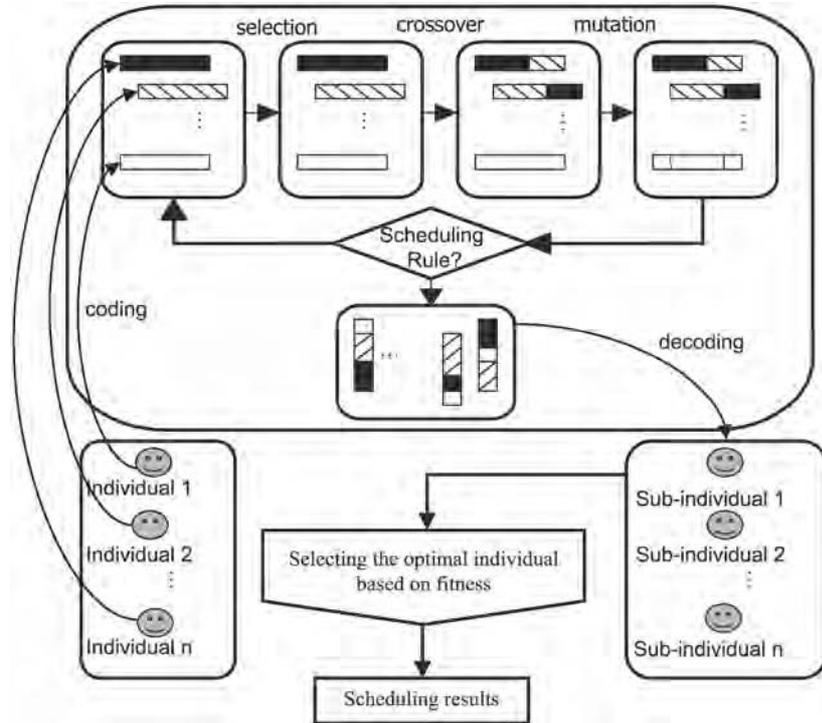
A GA is a kind of optimal algorithm, which imitates the evolution of biology and incorporates biological concepts into analytical studies of systems. The basic philosophy and work flow of GA are described in detail as follows:

- 1 We select and create an initial population from the latent result set of a problem. Here, each population is made up of a series of individual strings, also called chromosomes, and each individual represents a possible solution to a given problem.
- 2 Then, based on the principle of "survival of the fittest" and the fitness value of each individual assigned by the fitness function, extremely appropriate chromosomes are assigned more opportunities to reproduce and the offspring share features taken from their parents.
- 3 The GA executes three operators: evolution, selection, and recombination. Here, the evolution is used to decide whether or not a chromosome is better based on a fitness function. The selection is a process in which individual strings are copied according to their fitness function values. The recombination always consists of two sub-operators: cross-over and mutation. Through carrying out these three operators and their sub-operators, a new population will be produced.
- 4 Finally, repeatedly applying (2) and (3) until some termination criterion is satisfied, the fitness of the final population will converge and arrive at an optimal and near-optimal solution.

Based on the mechanism and work flow of GA, Figure 3 illustrates the scheduling mechanism and presents a scheduling model for e-PMC. Its scheduling logic can be divided into the following three steps:

- 1 Mapping the exterior representing model to the inner gene model and deciding the number of the population;
- 2 Carrying out the cross-over and mutation operations to produce the new population based on the designed genetic operators and deciding the times of evolution operations according to the scheduling rules such as production period; and
- 3 Selecting out the optimal individual as a near-optimal solution and decoding it as the scheduling result of e-PMC according to the final fitness value.

Figure 3
The scheduling logic for e-PMC based on GA



6.2 Design of GA-based scheduling model for e-PMC

On the basis of the philosophy of GA and mathematical model, the GA-based scheduling model for e-PMC can be divided into three steps.

6.2.1 Copy operations

In fact, creating the evolution objects plays an important role in solving any practical problem with GA. It becomes much more important how we transform the practical scheduling problems of e-PMC into "gene", "chromosome" and "population" that are the basic elements of GA.

(1) *The genes of e-PMC scheduling.* In order to create the genes of this scheduling model, the first thing we have to deal with is to code for the corresponding objects. In this paper, we take manufacturing tasks as our research objects. Referring to the manufacturing tasks, we adopt a series of characters to code for them. Here, these characters mainly consist of "0", ..., "9", "-", ",", ". Through assorting these characters to form a character string, a manufacturing task can be coded well. In the e-PMC, the coding format for each process of a manufacturing task can be expressed as follows:

$$("i", "j", "m", "t", "ID-Pro-Part"),$$

where "i" represents the task ID in the e-PMC "j" represents the process ID of the

manufacturing task, "m" represents the machining capability description of the equipment related to the process, "t" represents the machining time of the process, "ID-Pro-Part" represents the relationship between the manufacturing task and the correspondent e-manufacturing system. For example, (1, 2, 2, 6, 01-01-01) expresses that manufacturing task "01" belongs to project "01" of e-manufacturing system "01", process "2" belongs to manufacturing task "01" is machined on an e-equipment whose machining capability is assigned "2", the machining time of process "2" is assigned "6", and the ID of this manufacturing task is "1". With this expression, all manufacturing tasks can be described clearly. We can also declare that the coding for a manufacturing task is the combination of coding its processes.

In the same way, we can code for the equipment related to the manufacturing tasks. In this paper, we design the following coding expression for every e-equipment in an e-PMC:

$$("j", "c", "s"),$$

where "j" represents the coding of the e-equipment in an e-PMC, "c" represents the machining capability of the e-equipment, and "s" represents the current status of the e-equipment.

At the same time, we have to code for the finishing time of manufacturing tasks. In this paper, the coding for finishing time is described as (“*i*”, “*d*”). Here, “*i*” represents the manufacturing task, and “*d*” stands for finishing time.

Based on the coding method mentioned above, we can throw out the definition of genes of scheduling model for e-PMCs. Here, genes are represented as a set of manufacturing task coding, “process” coding of manufacturing tasks, manufacturing equipment coding, and finishing time coding of every manufacturing task.

(2) *The chromosomes.* Similar to biological bodies, e-PMCs are carriers of the genes related to the e-PMCs. To code for the chromosomes of e-PMC scheduling, we must keep to the following principles:

- The chromosome-coding scheme has to be simple, so as to improve the rate of evolution operations.
- The chromosome-coding scheme has to be convenient for the design and operations of the genetic operators.
- The chromosome-coding scheme has to keep its solution validity, even if it is transformed because of selection, cross-over and mutation, etc.

Based on the principles and the coding schemes concerning processes, manufacturing tasks and equipment mentioned above, for example, a coding string for a chromosome can be described as “*i-j-i-k-i-j-k-j-k*”, and its length relies on the total number of all the processes concerning all manufacturing tasks. Here, “*i*”, “*j*” and “*k*” represent manufacturing task IDs and will be replaced with their correspondent manufacturing task coding in the e-PMCs. The times of their appearing in the chromosome depend on the number of processes that are connected respectively with their own manufacturing task. The position sequence of their appearing in the chromosome represents the sequence of different processes of a manufacturing task to be machined. After filling in the correspondent ID values, the above chromosome coding string is like “1-2-1-3-1-2-3-2-3”, which represents that there exist three manufacturing tasks and each manufacturing task has three processes. This chromosome also consists of the machining queue information of the manufacturing tasks. The machining flow hinted at from this chromosome declares such facts as that process “1” of task “1” is first machined, and then process “1” of task “2”, and then process “2” of task “1”, and then process “1” of task

“3”, etc. Based on the method, the final scheduling result can be reached.

(3) *The population of e-PMC scheduling.* The population of e-PMC scheduling is a combination of some individuals of e-PMC scheduling. Because the chromosomes of the individuals include the evolution information, these individuals participating in the evolution procedure can be represented with their chromosomes. Under this condition, an individual of e-PMC scheduling can be described as a valid coding string of a chromosome.

During the implantation of population of e-PMC scheduling, the following requirements must be satisfied:

- The produced individuals of e-PMC scheduling should not be reduplicated.
- The produced individuals of e-PMC scheduling should be random and distributive.

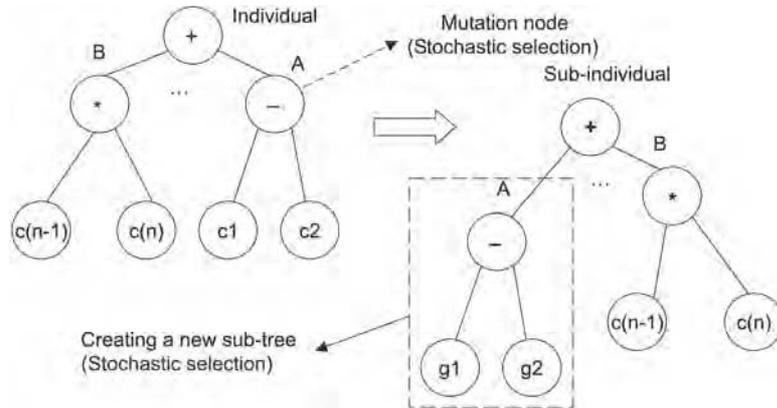
6.2.2 Evolution

The genetic operators of e-PMC scheduling mainly consist of selection, cross-over and mutation, whose detailed functions are described as follows:

(1) *Selection.* Selection operator is used to decide the individuals for cross-over operations in population and the number of sub-individuals produced by selected individuals. So, the fitness value of individuals must be calculated first. Then, according to the sequence of fitness value of individuals, the number of individuals is selected and the concrete individuals used for mutation are confirmed randomly at the mutation probability.

(2) *Cross-over.* Cross-over operator is used to create the new sub-individual based on the excellent individual. In order to expediently describe the process of cross-over operations of e-PMC scheduling, we adopt structure trees with the tabs to describe every individual. Figure 4 shows using the structure trees to describe individuals and the cross-over operations between two individuals. In Figure 4, the root node (individual) is described with character “+”, the inner nodes with characters “*” and “_”, and the leaf nodes correspond to genes described with “c1”, “c2”, ..., “g1”, “g2”, ... So, the cross-over operations can be implemented by exchanging the sub-trees of the two trees (corresponding to the two individuals) to produce two new trees (corresponding to the two new individuals after cross-over). In Figure 4, node “A” related to two individuals is randomly selected as cross-over node and the correspondent cross-over procedure between them is illustrated. Similarly, according to

Figure 4
 Using structure trees to describe individuals and the cross-over operation between them



the different number of selected cross-over nodes, the concepts of single-node cross-over and multi-node cross-over are also considered in this paper.

(3) *Mutation*. In e-PMC scheduling, the design philosophy for mutation operation follows such facts that a new sub-tree is produced through replacing the sub-tree of an individual selected for mutation and then a new individual is created. On the basis of checking the validity of this new individual, we can decide whether or not the individual is an offspring. Figure 5 shows the mechanism of mutation operations.

6.2.3 Deciding the fitness function of e-PMC scheduling

Because the objective function of e-PMC scheduling is to minimize the maximal

values of finishing time of the last process of all manufacturing tasks, we directly adopt the objective function of e-PMC scheduling as the fitness function, which is described as follows:

$$Fitness = \max\{S(p[i], k[N(i)], m[j])$$

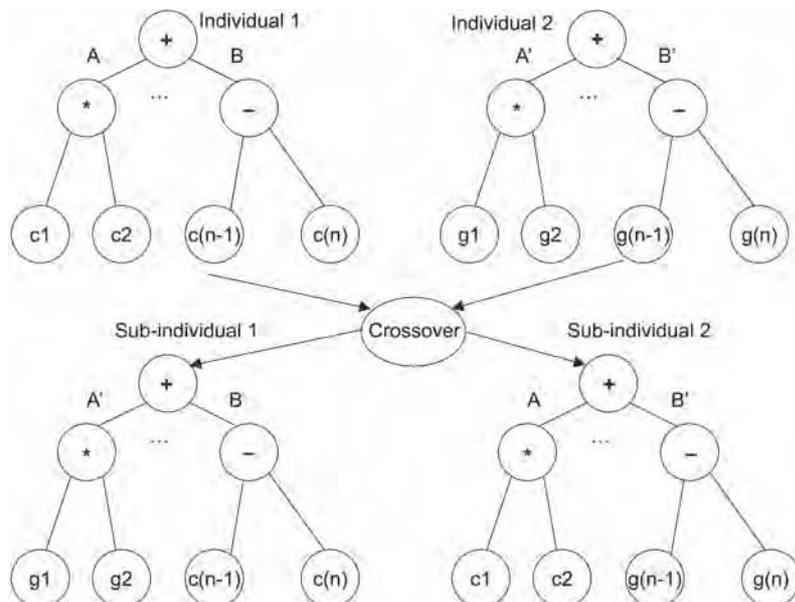
$$+ T(p[i], k[N(i)], m[j])\}$$

$$1 \leq i \leq n, 1 \leq j \leq m.$$

6.3 Software implementation of GA-based scheduling model for e-PMCs

According to the above-mentioned scheduling mechanism and the basic traits of e-manufacturing, we develop a Web-based scheduling module for e-PMCs as an

Figure 5
 The mechanism of mutation operations



attachment of the e-service platform for online part e-manufacturing (Jiang *et al.*, 2002). This module is under the control of the platform and depends on the Java Web solution concerning a three-tier “browser/server/database” infrastructure. Referring to the machining process information of in-house and out-house workpieces derived from BOM flow of products and the resource information of e-equipments in e-PMCs, the Web-based and task-driven scheduling operations are realized. Furthermore, in terms of associating the manufacturing tasks that are posted into e-PMCs with correspondent e-LMCs and referring to the relationship between e-LMCs and an e-MS, the scheduling based on manufacturing tasks for the whole e-manufacturing system can also be exported in a collaborative way. Figure 6 just illustrates the software architecture for task-driven e-manufacturing scheduling mechanism on the Web.

scheduling model developed for e-PMCs. In order to run a scenario, we set up the following initial data:

- The number of population is assigned as “40”.
- The cross-over probability is designated as “0.8”.
- The mutation probability is “0.05”.
- The e-PMC consists of five items of e-equipment in which equipment “5” is a grinding machine and the others are lathes.

The data used in the scheduling scenario come from a Web-based database of a computer-aided process-planning system and are shown in Table I.

After inputting the above initial data into the module, we can execute it to simulate the scheduling procedure of e-PMC. Figure 7 shows a screen snapshot concerning the optimal scheduling solution in the form of the Gantt charts.

7. Case study

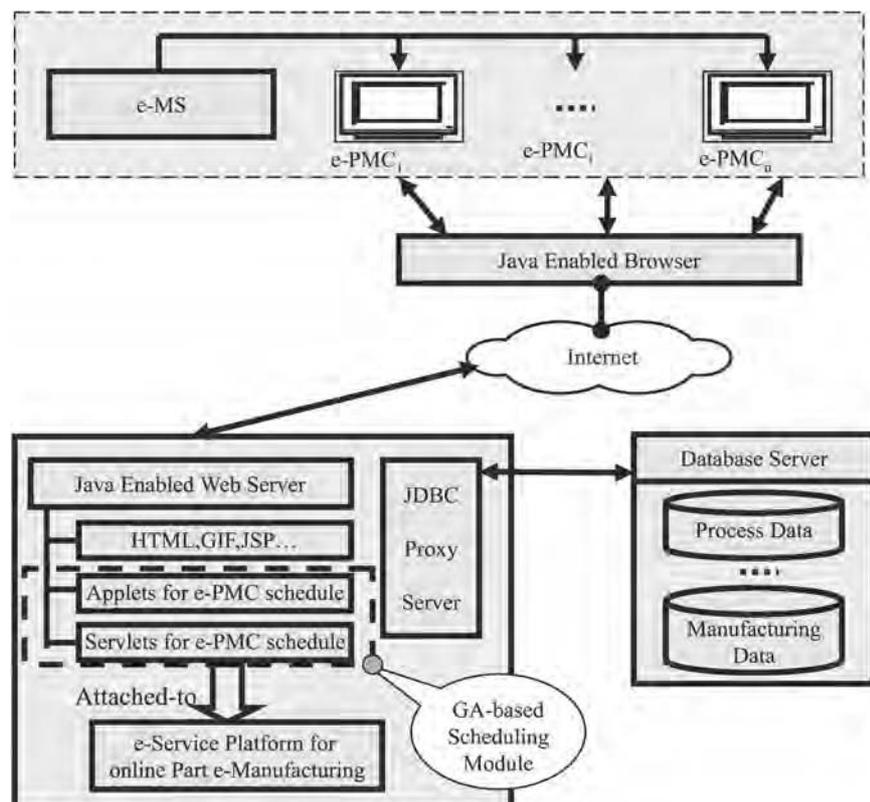
Based on the above ideas, we use the following case to test the GA-based

8. Concluding remarks

E-manufacturing scheduling mechanism plays an important role in enabling and executing e-manufacturing systems. By

Figure 6

The software architecture for task-driven e-manufacturing scheduling system



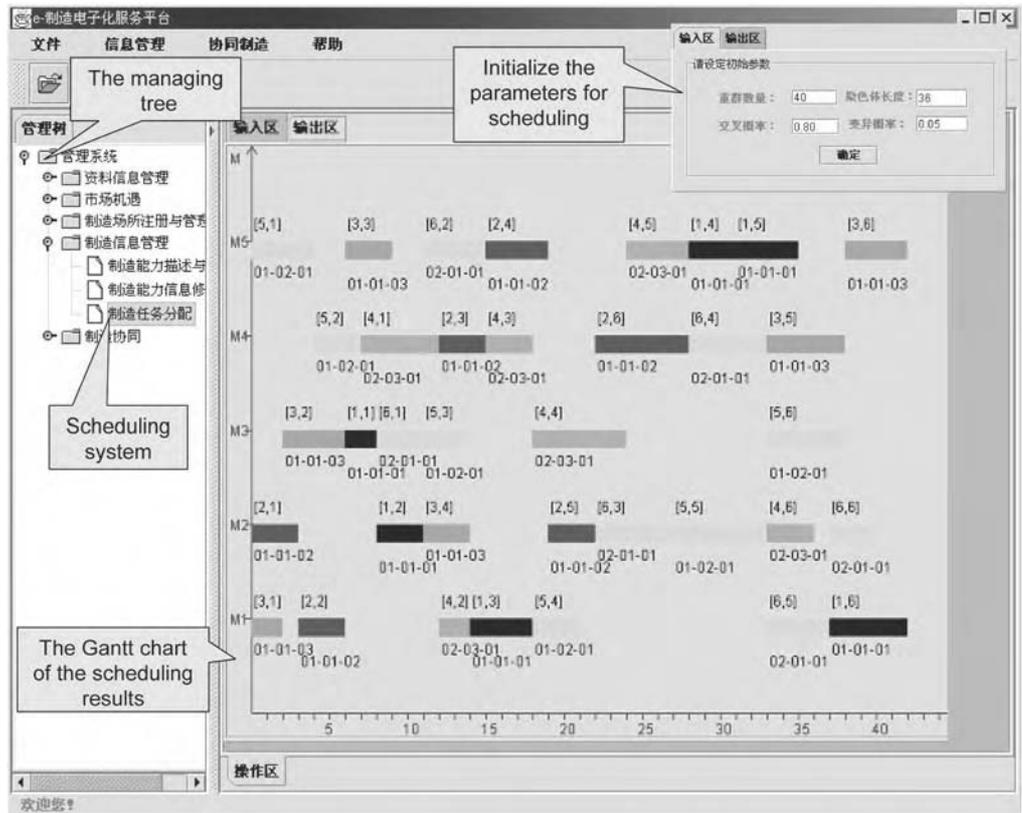
Yingfeng Zhang, Pingyu Jiang and Guanghui Zhou
GA-driven part e-manufacturing scheduling via an online e-service platform
 Integrated Manufacturing Systems
 14/7 [2003] 575-585

Table 1
 The initial data for a 6 × 6 scheduling problem

Task ID	Process ID related to task (<i>i, j, m, t, ID</i>)	Finishing time
1	(1,1,1,2,01-01-01)(1,2,1,3,01-01-01)	46
	(1,3,1,4,01-01-01)(1,4,2,3,01-01-01)	
	(1,5,2,4,01-01-01)(1,6,1,5,01-01-01)	
2	(2,1,1,3,01-01-02)(2,2,1,3,01-01-02)	30
	(2,3,1,3,01-01-02)(2,4,2,4,01-01-02)	
	(2,5,1,3,01-01-02)(2,6,1,6,01-01-02)	
3	(3,1,1,2,01-01-03)(3,2,1,4,01-01-03)	44
	(3,3,2,3,01-01-03)(3,4,1,3,01-01-03)	
	(3,5,1,5,01-01-03)(3,6,2,4,01-01-03)	
4	(4,1,1,5,02-03-01)(4,2,1,2,02-03-01)	38
	(4,3,1,3,02-03-01)(4,4,1,6,02-03-01)	
	(4,5,2,4,02-03-01)(4,6,1,3,02-03-01)	
5	(5,1,2,4,01-02-01)(5,2,1,3,01-02-01)	35
	(5,3,1,3,01-02-01)(5,4,1,3,01-02-01)	
	(5,5,1,6,01-02-01)(5,6,1,5,01-02-01)	
6	(6,1,1,3,02-01-01)(6,2,2,4,02-01-01)	40
	(6,3,1,5,02-01-01)(6,4,1,5,02-01-01)	
	(6,5,1,4,02-01-01)(6,6,1,3,02-01-01)	

Notes: In this Table (*i, j, m, t, ID*) expresses that process “*j*” belongs to task “*i*”, machining time of process “*j*” is “*t*”, “*m*” marks the machining constraint of process “*j*” (for example, “1” corresponds to lathe and “2” to grinding machine tool), and “*ID*” describes the relationship between the manufacturing task and the e-manufacturing system to do this task

Figure 7
 An optimal scheduling result for e-PMC



means of using the genetic algorithm, in this paper, the mathematical model for e-manufacturing scheduling is described. On the basis of defining the concepts of e-ME, e-MS, e-PMC, and e-LMC and analyzing the relationship between them, the scheduling issue for e-MS can be brought down to the scheduling issue for the e-PMCs. Through establishing the mathematical model for scheduling the e-PMCs, the Web-based and GA-driven module is designed, developed and demonstrated based on part machining. The case study has proved that the implementation can be used efficiently to control the e-manufacturing procedure in the level of e-manufacturing environment via an online e-Service platform for part online manufacturing. We also hope that the research philosophy and results obtained can establish a foundation for the e-manufacturing mode.

References

- Cardon, A., Galinho, T. and Vacher, J.P. (2000), "Genetic algorithms using multi-objectives in a multi-agent system", *Robotics and Autonomous Systems*, Vol. 33 No. 2/3, pp. 179-90.
- Chenga, R., Genb, M. and Tsujimura, Y. (1999), "A tutorial survey of job-shop-scheduling problems using genetic algorithms, part II: hybrid genetic search strategies", *Computers & Industrial Engineering*, Vol. 36, pp. 343-64.
- Davis, L. (1985), "Job-shop scheduling with genetic algorithms", *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 136-40.
- He, Z.S., Yang, T. and Tiger, A. (1996), "An exchange heuristic imbedded with simulated annealing for due-dates job-shop scheduling", *European Journal of Operational Research*, Vol. 91 No. 2, pp. 99-117.
- Jiang, P.Y., Zhou, G.H. and Liu, Y. (2002), "ASP-based e-service platform for Web-based online manufacturing", *Integrated Manufacturing Systems*, Vol. 13 No. 5, pp. 306-18.
- Jürgen, S., Suelmann, G. and Appelrath, H.J. (1998), "Multi-site scheduling with fuzzy concepts", *International Journal of Approximate Reasoning*, Vol. 19 No. 1/2, pp. 145-60.
- Lawrence, S.R. and Sewell, E.C. (1997), "Heuristic, optimal, static, and dynamic schedules when processing times are uncertain", *Journal of Operations Management*, Vol. 15, pp. 71-82.
- Lee, H.C. and Dagli, C.H. (1997), "A parallel genetic-neuro scheduler for job-shop-scheduling problems", *International Journal of Production Economics*, Vol. 51, pp. 115-22.
- Maturana, F., Gu, P., Naumann, A. and Norrie, D.H. (1997), "Object-oriented job-shop scheduling using genetic algorithms", *Computers in Industry*, Vol. 32, pp. 281-94.
- Nobuaki, I. and Masaaki, M. (1997), "A generic framework for an online scheduling and control system in batch process management", *Computers & Chemical Engineering*, Vol. 21 No. 11, pp. 1291-310.
- Sakawa, M. and Mori, T. (1999), "An efficient genetic algorithm for job-shop-scheduling problems with fuzzy processing time and fuzzy due date", *Computers & Industrial Engineering*, Vol. 36, pp. 325-41.
- Watanabe, T., Hashimoto, Y., Nishikawa, I. and Tokumaru, H. (1995), "Line balancing using a genetic evolution model", *Control Engineering Practice*, Vol. 3 No. 1, pp. 69-76.
- Yamada, T. and Nakano, R. (1992), "A genetic algorithm applicable to large-scale job problems", *Parallel Problem Solving from Nature*, Vol. 2, pp. 281-90.
- Yang, S.X. and Wang, D.W. (2002), "A new adaptive neural network and heuristics hybrid approach for job-shop scheduling", *Computers & Operations Research*, Vol. 28, pp. 325-41.
- Yao, Z.H., Zhao, G.W. and Han, B. (2001), "The comparison of multi-reproduction groups of genetic algorithms and its application in the optimization schedule", *Control Theory and Applications*, Vol. 18 No. 6, pp. 882-6.
- Yun, S.Y. (2002), "Genetic algorithm with fuzzy logic controller for preemptive and non-preemptive job-shop-scheduling problems", *Computers and Industrial Engineering*, Vol. 43, pp. 623-44.

The research is under the supports of national 863 High-Tech Research and Development Program of China (Project No. 2001AA415230), the Excellent Young Teachers Program of Ministry of Education, China (Project No. 20011886). The authors hereby thank them for their financial support.

Developing the methods of modeling heterogeneous components

Ke-Zhang Chen

Department of Mechanical Engineering, The University of Hong Kong, Hong Kong

Xin-An Feng

School of Mechanical Engineering, Dalian University of Technology, Dalian, China

Keywords

Component manufacturing, Modelling, Materials management, Computer aided design

Abstract

In order to represent, analyze, optimize, and manufacture a component made of multi-heterogeneous materials for high-tech applications, a computer model of the heterogeneous component needs to be built first. Heterogeneous materials include composite, functionally graded materials, and heterogeneous materials with a periodic microstructure. Current modeling techniques focus only on capturing the geometric information and cannot satisfy the requirements from modeling the components made of multi-heterogeneous materials. This paper develops a modeling method, which can be implemented by employing the functions of current CAD graphic software and can obtain the model including both the material information (about its microstructures and constituent composition) and the geometry information without the problems arising from too many data.

1. Introduction

With rapid developments of high technology in various fields, there appear to be more critical requirements for special functions of components/products, such as negative Poisson's ratio, zero thermal expansion coefficient, and extreme bulk modulus. These special requirements cannot be satisfied by using conventional homogeneous materials. The attention has now focused on heterogeneous materials (HM), including composite materials, functionally graded materials, and heterogeneous materials with a periodic microstructure. A heterogeneous component (HC) can be made of single material in a heterogeneous form or multiple heterogeneous materials, according to the requirements for special functions of the components/products (Chen and Feng, 2003). In order to design and manufacture the heterogeneous components, the computer models for representing the heterogeneous components, especially those made of multi-heterogeneous materials, need first to be built, so that further analysis, optimization and layered manufacturing can be implemented based on the models.

Current modeling techniques focus only on capturing the geometric information (Lee, 1999). Some researchers are now focusing on modeling heterogeneous objects by including both the variation in constituent composition and the geometry in the solid model (Kumar and Dutta, 1998; Kumar *et al.*, 1999; Jackson *et al.*, 1999; Siu and Tan, 2002a, b; Morvan and Fadel, 2002). But modeling or representing the microstructure of heterogeneous components is beyond their scope (Kumar and Dutta, 1998). Since the microstructure size is very small, a model consisting of such microstructures would have a huge number of data to be stored. Even with the help of

high-speed modern computers, the processing of such a model is difficult and needs extreme care and thoughts for I/O operations. The modeling should, therefore, be able to apply the functions of current CAD systems without compromising the speed of the operations and the reasonable utilization of computer resources.

This paper develops a modeling method, using axiomatic design (Suh, 1990, 2001), which can be implemented by employing the functions of current CAD graphic software and can obtain the model including both the material information about its microstructures and constituent compositions and the geometry information without the problems arising from too many data.

2. Requirements of modeling heterogeneous components

Before developing a modeling method for heterogeneous components, the requirements for representing a component made of heterogeneous materials should be made clear first. Since heterogeneous materials cover composite materials, functionally graded materials, and heterogeneous materials with a periodic microstructure, the requirement for each one of them is analyzed, respectively, as follows.

A composite material (Berthelot, 1999; Chawla, 1998; Barbero, 1998) consists of one or more discontinuous phases distributed in one continuous phase. The continuous phase is called the matrix and may be resin, ceramic, or metal. The discontinuous phase is called reinforcement or inclusion and may be fibers, particles, or voids. Reinforcement is used to improve certain properties of matrices, such as stiffness, behavior with temperature, and resistance to abrasion. For



Integrated Manufacturing Systems
14/7 [2003] 586-592

© MCB UP Limited
[ISSN 0957-6061]
[DOI 10.1108/09576060310491946]

The Emerald Research Register for this journal is available at
<http://www.emeraldinsight.com/researchregister>



The current issue and full text archive of this journal is available at
<http://www.emeraldinsight.com/0957-6061.htm>

instance, lead inclusions in copper alloys make them easier to be machined. The properties of composite materials result mainly from the properties of both their matrix and reinforcements and the geometry and distribution of their reinforcements. Thus, to describe a component made of a composite material, its model will have to specify the geometry, material, and distribution of reinforcement and the matrix material as well as the geometry of the component.

Functionally graded materials are used to join two different materials without stress concentration at their interface. Gradation in properties from one portion to another portion can be determined by material constituent composition. The volume fraction of one material constituent should be changed from 100 per cent on one side to zero on another side, and that of another material constituent should be changed the other way round. In fact, there are many material composition functions (Bhashyam *et al.*, 2000). The designers can choose certain composition functions from them for their applications. For example, the following parabolic function is selected for material composition function of the metal/ceramic functionally graded material in the cylinder of vehicular engines or pressure vessels:

$$V_A = a_0 + a_1x + a_2x^2 \quad (1)$$

where V_A is the volume fraction of metal and x is the distance from one side. The coefficients of the parabolic function are optimized subject to criteria that the thermal flux across the material is minimized, and the thermal stresses are minimized and restricted below the yield stress of the composite material. After the determination of composition function, physical properties can also be estimated based on property estimation models (Bhashyam *et al.*, 2000). Thus, in order to describe a component made of a functionally graded material, its model will have to specify its material constituents and their constituent composition as well as the geometry of the component.

Heterogeneous material with a periodic microstructure is indicated in Figure 1. Such materials are described by the base cell, which is the smallest repetitive unit of material and comprises a material phase and a void phase. It should be emphasized that, in comparison with the dimensions of the component, these non-homogeneities should be very small in size. The effective properties of the heterogeneous material are determined by the topology of its base cell and the properties of its constituents, and can be predicted by the mathematical theory of

homogenization (Bendsoe, 1995; Hassani and Hinton, 1998a, b). In other words, the effective properties of the heterogeneous material can be changed by designing various topologies of its base cells. With the homogenization method, the topology of the base cell can be designed, according to certain requirements, using topology optimization (Silva *et al.*, 1997; Larsen *et al.*, 1997). Thus, in order to describe a component made of a heterogeneous material with a periodic microstructure, its model will have to specify its material and its microstructure (or base cell) as well as the geometry of the component.

According to the above analysis, it is obvious that the functional requirement (FR) of its model can be decomposed into three sub-FRs, representing: geometry, material constituent composition, and material microstructures (including the geometry and distribution of reinforcement for composite materials) of the component, which are noun phrases corresponding to "what we want to achieve" and can be written as:

FR_1 . Representing geometry of the component.

FR_2 . Representing material constituent compositions of the component.

FR_3 . Representing material microstructures of the component.

In other words, in order to represent a component made of composite, functionally graded materials, and/or heterogeneous materials with a periodic microstructure, its model has to be able to satisfy the above three functional requirements. According to axiomatic design (Suh, 1990, 2001), the model should be decomposed into three sub-models as the design solution (DS) to satisfy the three FRs, respectively. These DSs are stated starting with a verb corresponding to "how we achieve it" and can be written as:

DS_1 . Build its geometric model.

DS_2 . Build its material constituent composition model.

DS_3 . Build its material microstructure model.

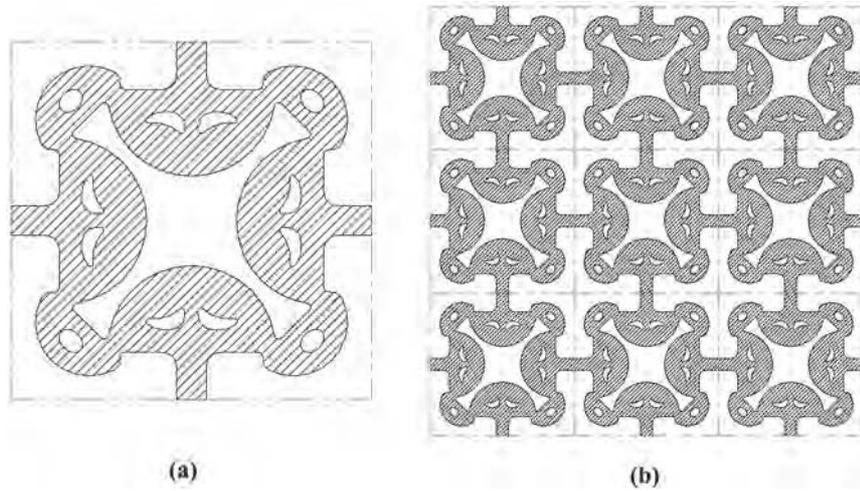
Thus, according to axiomatic design, the design equation for it can be obtained as follows:

$$\begin{Bmatrix} FR_1 \\ FR_2 \\ FR_3 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 \\ X & X & 0 \\ X & X & X \end{bmatrix} \begin{Bmatrix} DS_1 \\ DS_2 \\ DS_3 \end{Bmatrix} \quad (2)$$

From the equation obtained, it can be seen that the design matrix is a triangular matrix, which indicates that the design is a decoupled design and satisfies the independence axiom (Suh, 1990). Therefore, these DSs can be adopted, that is, it is correct to build the three sub-models to represent the

Figure 1

A heterogeneous material with a periodic microstructure in a Cartesian coordinate system



component made of multi-heterogeneous materials.

3. Modeling of heterogeneous components

A 3D model representing the geometry of a component can be made by using current CAD graphic software (Lee, 1999), and is indicated by G . This model can be divided into n regions or parts based on their different types of material constituent compositions. Thus, the material constituent composition set can be indicated by:

$$C = \{C_i, i = 1, 2, \dots, n\}. \quad (3)$$

According to its material microstructures, the geometry model can also be divided into m regions or parts, if there are m different types of microstructures. The material microstructure set can be written as:

$$S = \{S_j, j = 1, 2, \dots, m\}. \quad (4)$$

Thus, the material set (M) of the component can be obtained by solving Cartesian product of C and S as:

$$M = C \times S =$$

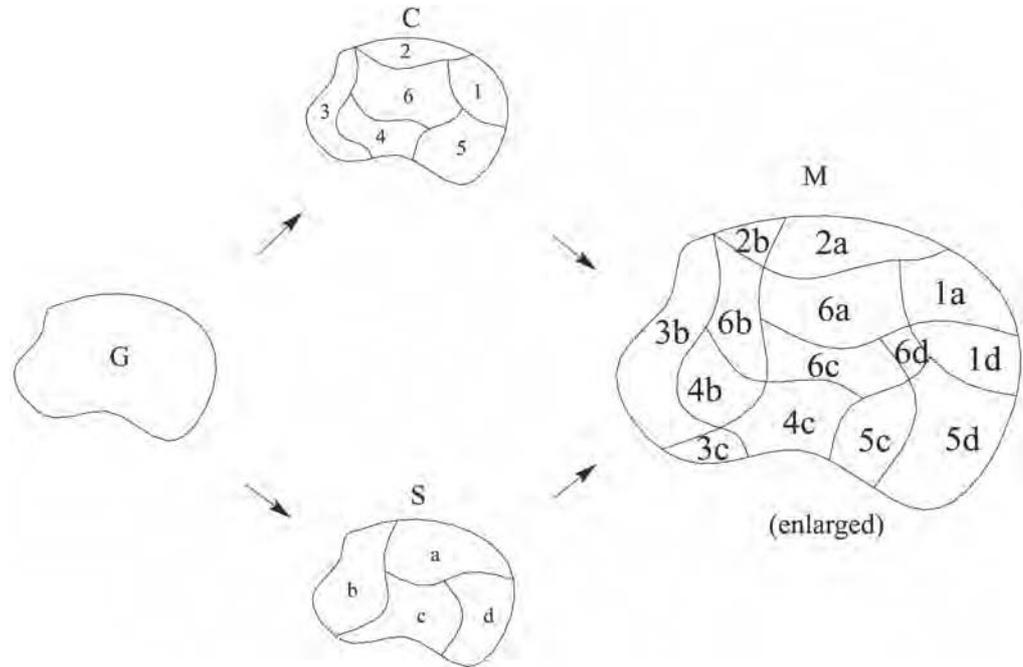
$$\{M_{ij} | i \in (1, 2, 3, \dots, n), j \in (1, 2, \dots, m)\}. \quad (5)$$

For example, there are six different material constituent composition regions ($n=6$) and four different material microstructure regions ($m=4$) in the component G , as shown in Figure 2. Solving its Cartesian product of C and S can obtain 14 material regions, each of which contains the same information of both material constituent composition and microstructure. The first Arabic figure of the symbol of each region is the code name of

material constituent composition region, and the second English letter is the code name of material microstructure region. Region $6a$, for instance, indicates that its material constituent composition in this region is determined by that in Region 6 of material constituent composition set and its material microstructure is specified by that in Region a of material microstructure set.

A model is an approximation of the component or object along one or more dimensions of interest, and can be any entity that exhibits some aspect of the component that is required for the purposes concerned (Ulrich and Eppinger, 2000). Therefore, modeling can be in many different forms, such as a physical model, a wire-wrapped circuit board, a system of equations, frames and slots (i.e. schema) (Jonassen *et al.*, 1993), 3D solid models, or their combinations. Since modeling the components made of multi-heterogeneous materials needs to include not only its 3D geometry but also all the material information about periodic microstructures, constituent composition, and inclusions, it is obvious that contemporary computer solid modeling systems are not sufficient for the purpose and other types of models have to be applied. A model representing a component made of multi-heterogeneous materials can be decomposed into three sub-models, as analyzed in the previous section, so as to include all the required information. The three types of sub-models are: geometric model (i.e. 3D solid model), material constituent composition model, and microstructure model for each material region. For geometric model, the B-rep scheme (Lee, 1999) can be used to represent the shape of the whole component and all the borders between different material regions in the component. We will use a schema to

Figure 2
An example of heterogeneous components



represent the structural knowledge or information for each of the last two sub-models, since the schema is easy to use to establish the linkage among graphic library, database and application software, which is prerequisite for modeling the components made of multi-heterogeneous materials. Each schema consists of several frames. Each frame represents a type of inclusion or periodic microstructure cell and consists of several slots. Each slot contains a type of information to describe the frame in more detail, such as the type of inclusion material, the distribution function of each inclusion, the inserting array for each type of periodic microstructure cell, the type of the local coordinate system of the material region, the location and orientation of the local coordinate system in the global coordinate system, composition function of each material constituent, or the code name of the parametric geometrical model of the periodic microstructure cell.

3.1 Material constituent composition models

Each region in material constituent composition set has a specified material constituent composition. The volume fraction of the k -th material constituent at the position (x, y, z) can be represented as:

$$V_k = f_k(x, y, z). \quad (6)$$

This material composition function along with primary material combinations and intended applications can be obtained from

much of the literature (Bhashyam *et al.*, 2000), and organized into a database for applications. Designers may select suitable material composition functions from it for their applications according to the functional requirements of the component.

Based on schema theory (Jonassen *et al.*, 1993), frame and slots can be used to organize the knowledge for modeling. The model for the i -th material constituent composition region is then designed using axiomatic design as the following schema form:

$C_i = \{\text{Number of material types: } q_i$

Material type: m_1, m_2, \dots, m_{q_i}

Coordinate system type:

(Cartesian, cylindrical, or spherical coordinate system)

Origin of coordinate system: X_i, Y_i, Z_i

Orientation of coordinate system: $\alpha_i, \beta_i, \gamma_i$

Material constituent composition functions:

$$[V_{ki} = f_{ki}(x, y, z), k = 1, 2, \dots, q_i$$

$$| \sum_{k=1}^{q_i} V_k = 1, (x, y, z), \in C_i]$$

}

(7)

The first slot is the number of material types in the material region. The second slot is the material types represented by the code names of material types. All the information about a material type can be retrieved from a material database according to its code name. The third slot is the local coordinate system

type of material region, which may be Cartesian, cylindrical, or spherical coordinate system. The fourth and fifth slots are the origin and the orientation of the local coordinate system, respectively, which are based on global coordinate system. The sixth slot is the material constituent composition function set, which includes the composition function of each material constituent in the region that is the function of the position (x, y, z) . In each position, the sum of their volume fractions should be equal to one.

3.2 Material microstructure models

The material microstructure model introduced in this paper covers both the microstructure of composite materials and the periodic microstructure of heterogeneous materials.

3.2.1 Material microstructure models for composite materials

The composite material consists of matrix and reinforcements (or inclusions). The latter may have various shapes, sizes, and distributions. Their shapes and size are varied randomly, and their distribution in the matrix may be even or uneven. Since the components are considered to be made by layered manufacturing technology in this paper, the inclusions are sprayed on to the layer where the matrix material is being spread. Using the schema theory, the model for the k -th material microstructure region (with composite materials) is designed as the following form:

$S_k = \{$ Insertion: operation 1:

Inclusion material: "code name of material 1"

Coordinate system type: (Cartesian, cylindrical, or spherical coordinate system)

Origin of coordinate system: X_k, Y_k, Z_k

Orientation of coordinate system: $\alpha_k, \beta_k, \gamma_k$

Spraying function: $V_{k1} = f_{k1}(x, y, z)$

Insertion operation 2:

Inclusion material: "code name of material 2"

Coordinate system type: (Cartesian, cylindrical, or spherical coordinate system)

Origin of coordinate system: X_k, Y_k, Z_k

Orientation of coordinate system: $\alpha_k, \beta_k, \gamma_k$

Spraying function: $V_{k2} = f_{k2}(x, y, z)$

...

Insertion operation N :

Inclusion material: "code name of material N "

Coordinate system type: (Cartesian, cylindrical, or spherical coordinate system)

Origin of coordinate system: X_k, Y_k, Z_k

Orientation of coordinate system: $\alpha_k, \beta_k, \gamma_k$

Spraying function: $V_{kN} = f_{kN}(x, y, z)$

} (8)

This model consists of N frames based on the number of spraying operations, each of which describes one type of spraying operation. In each frame, each slot is defined as follows:

The first slot is the material type of inclusion, which is represented by the code name of material type. All the information about the inclusion (e.g. its material type, shape, and average size) can be retrieved from a material database according to its code name. The second slot is the local coordinate system type of material microstructure region, which may be Cartesian, cylindrical, or spherical coordinate system. The third and fourth slots are the origin and the orientation of the local coordinate system, respectively, which are based on global coordinate system. The fifth slot is the spraying volume fraction of inclusion that is a volume ratio between inclusion and matrix material and a function of the spraying position (x, y, z) .

3.2.2 Material microstructure models for heterogeneous materials with a periodic microstructure

A heterogeneous material with a periodic microstructure is described by its base cell, which is the smallest repetitive unit of material and comprises a material phase and a void phase. The base cells are arranged into a rectangular array (Figure 1b), cylindrical array or spherical array. Figure 3(b), for example, shows a cross-section of the base cells shown in Figure 3(a) in a cylindrical array, and also represents the cross-section passing the center in a spherical array.

The model for the j -th material microstructure region (with a heterogeneous material with a periodic microstructure) can be expressed as follows:

$S_j = \{$ Insertion: "code name of base cell"

Coordinate system type: Cartesian, cylindrical or spherical coordinate system

Origin of coordinate system: X_j, Y_j, Z_j

Orientation of coordinate system: $\alpha_j, \beta_j, \gamma_j$

Inserting position function:

$$(x, y, z) = [(x_j(t_1), y_j(t_2), z_j(t_3))$$

$$|(t_1, t_2, t_3) \in (\text{"Integer"} (x, y, z) \in S_j)$$

$$\left. \begin{aligned} D: F_{dj}(x, y, z) \\ \theta: f_{\theta j}(x, y, z) \end{aligned} \right\} \quad (9)$$

In this model, the first slot is the pattern of base cell, which can be retrieved from the parametric microstructure graphics library according to the code name of its pattern. The second slot is the local coordinate system, which may also be Cartesian, cylindrical, or spherical coordination system. The third and fourth slots are the origin and the orientation of the local coordinate system, respectively, which are based on global coordinate system. The fifth slot is the inserting position of base cells, which should be at a point in an array or matrix of local coordinate system and within its material microstructure region. For example, if Cartesian coordinate system is applied, its matrix can be determined, as shown in Figure 4, by:

$$\left. \begin{aligned} x &= \{k_x t_1 + c_x, t_1 = 1, 2, \dots\} = \{x_1, x_2, \dots\} \\ y &= \{k_y t_2 + c_y, t_2 = 1, 2, \dots\} = \{y_1, y_2, \dots\} \\ z &= \{k_z t_3 + c_z, t_3 = 1, 2, \dots\} = \{z_1, z_2, \dots\} \end{aligned} \right\} \quad (10)$$

where k_x, k_y, k_z, c_x, c_y and c_z are constants. The sixth slot is the dimension of base cell, which consists of all the parameters of its 3D parametric model and is determined by a special function set, $D = F_{dj}(x, y, z)$, where D is a vector including all the parameters of base cell, while x, y, z are the coordinates of inserting point of base cell. In Cartesian coordination system, the dimension vectors (D) for all the base cells shown in Figure 1(b) are the same, i.e. $D = \text{"constant"}$. But, if cylindrical or spherical coordination system

Figure 3

A heterogeneous material with a periodic microstructure in cylindrical coordinate system

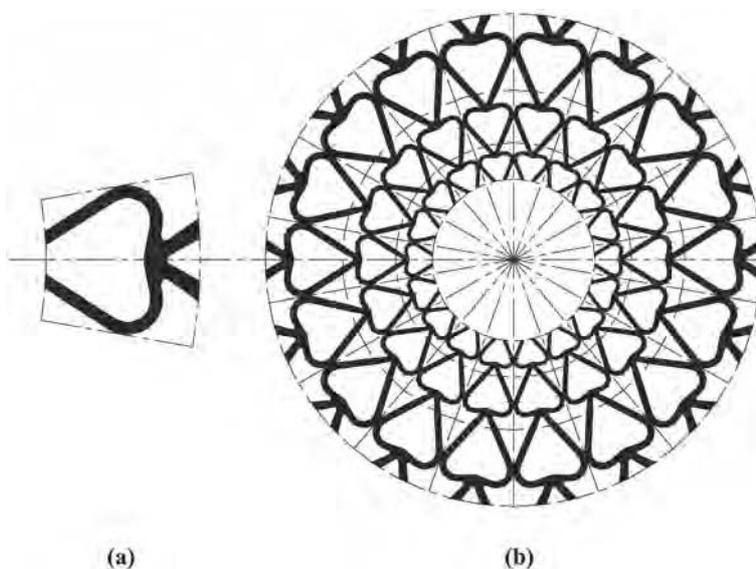
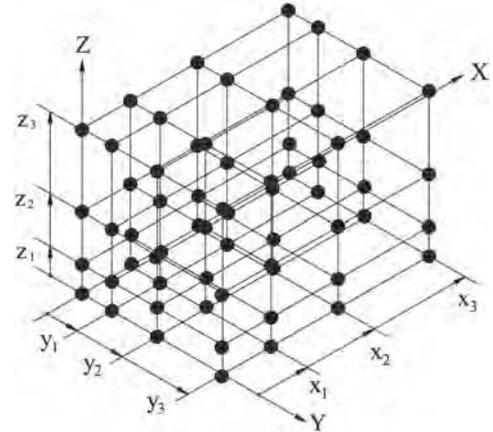


Figure 4

Inserting point positions



is used, as shown in Figure 3, the dimensions vectors of all the base cells in the same circle are the same and those in different circles are the linear functions of their radial position coordinates. The seventh slot is the orientation of base cell, which is also determined by a special function set, $\theta = f_{\theta j}(x, y, z)$, where θ is a vector including three axial angles of base cell, while x, y, z , are the coordinates of inserting point of base cell. In Cartesian coordination system, the orientation vectors (θ) for all the base cells shown in Figure 1(b) are the same, i.e. $\theta = \text{"constant"}$. But, if cylindrical or spherical coordination system is used, as shown in Figure 3, the orientation vectors of base cells are the normal vectors of their inserting points. Thus, the orientation vector of all the base cells in the same radial is the same.

4. Conclusions

This paper develops the methods for modeling heterogeneous components using axiomatic design. The method divides a component into two region sets (C and S) based on its material constituent compositions and material microstructures. Thus, a component can be decomposed, by Cartesian product of C and S , into material regions (M_{ij}), each of which has the same material constituent composition and microstructure. A model representing a component made of multi-heterogeneous materials can then be decomposed into three sub-models. The three types of sub-models are: geometric model (i.e. 3D solid model), material constituent composition model, and microstructure model for each material region. For geometric model, the B-rep scheme can be used to represent the shape of the whole component and all the borders

between different material regions in the component. The other two types of sub-models are built, respectively, using schema theory. This method can employ the functions of current CAD graphic software and obtain the model including the material information about both microstructures and constituent compositions along with geometry information in current 3D solid modeling without problems arising from too many data.

References

- Barbero, E.J. (1998), *Introduction to Composite Materials Design*, Taylor & Francis, Ann Arbor, MI.
- Bendsoe, M.P. (1995), *Optimization of Structure Topology, Shape, and Material*, Springer Verlag, Berlin.
- Berthelot, J.M. (1999), *Composite Materials: Mechanical Behavior and Structural Analysis*, Springer-Verlag, New York, NY.
- Bhashyam, S., Shin, K.H. and Dutta, D. (2000), "An integrated CAD system for design of heterogeneous objects", *Rapid Prototyping Journal*, Vol. 6 No. 2, pp. 119-35.
- Chawla, K.K. (1998), *Composite Materials: Science and Engineering*, Springer-Verlag, New York, NY.
- Chen, K.Z. and Feng, X.A. (2003), "Computer-aided design method for the components made of heterogeneous materials", *Computer-Aided Design*, Vol. 35 No. 5, pp. 453-66.
- Hassani, B. and Hinton, E. (1998a), "A review of homogenization and topology optimization I – homogenization theory for media with periodic structure", *Computers and Structures*, Vol. 69 No. 6, pp. 707-17.
- Hassani, B. and Hinton, E. (1998b), "A review of homogenization and topology optimization II – analytical and numerical solution of homogenization equations", *Computers and Structures*, Vol. 69 No. 6, pp. 719-38.
- Jackson, T.R., Liu, H., Patrikalakis, N.M., Sachs, E.M. and Cima, M.J. (1999), "Modeling and designing functionally graded material components for fabrication with local composition control", *Materials and Design*, Vol. 20 No. 2/3, pp. 63-75.
- Jonassen, D.H., Beissner, K. and Yacci, M. (1993), *Structural Knowledge: Techniques for Representing, Conveying, and Acquiring Structural Knowledge*, Lawrence Erlbaum Associates, Hillsdale, NJ.
- Kumar, V. and Dutta, D. (1998), "An approach to modeling and representation of heterogeneous objects", *Journal of Mechanical Design*, Vol. 120 No. 4, pp. 659-67.
- Kumar, V., Burns, D., Dutta, D. and Hoffmann, C. (1999), "A framework for object modeling", *Computer-Aided Design*, Vol. 31 No. 9, pp. 541-56.
- Larsen, U.D., Sigmund, O. and Bouwstra, S. (1997), "Design and fabrication of compliant micromechanisms and structures with negative Poisson's ratio", *Journal of Micro-electromechanical Systems*, Vol. 6 No. 2, pp. 99-106.
- Lee, K. (1999), *Principle of CAD/CAM/CAE System*, Addison-Wesley Longman, Reading, MA.
- Morvan, S. and Fadel, G.M. (2002), "MMA-rep, a V-representation for multi-material object", *Journal of Software Solutions for Rapid Prototyping*, PEP Press, London.
- Silva, E.C.N., Fonseca, J.S.O. and Kikuchi, N. (1997), "Optimal design of piezo-electric microstructures", *Computational Mechanics*, Vol. 19 No. 5, pp. 397-410.
- Siu, Y.K. and Tan, S.T. (2002a), "Source-based heterogeneous solid modeling", *Computer-Aided Design*, Vol. 34 No. 1, pp. 41-55.
- Siu, Y.K. and Tan, S.T. (2002b), "Modeling the material grading and structures of heterogeneous objects for layered manufacturing", *Computer-Aided Design*, Vol. 34 No. 10, pp. 705-16.
- Suh, N.P. (1990), *The Principle of Design*, Oxford University Press, New York, NY.
- Suh, N.P. (2001), *Axiomatic Design: Advances and Applications*, Oxford University Press, New York, NY.
- Ulrich, K.T. and Eppinger, S.D. (2000), *Product Design and Development*, McGraw-Hill Company, Boston, MA.

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. HKU 7062/00E). The financial contribution is gratefully acknowledged.

Intelligent process-planning system or optimal CNC programming – a step towards complete automation of CNC programming

Millan K. Yeung

Integrated Manufacturing Technologies Institute, National Research Council
Canada, London, Ontario, Canada

Keywords

Process planning,
Artificial intelligence,
Knowledge management,
Computer numerical control,
Computer aided design

Abstract

One of the bottle-necks of computer numerical control (CNC) machining is the CNC programming. It relies on the experience and skills of the CNC programmer for the generation of the CNC program. The intelligent process-planning system described in this paper generates a process plan automatically for CNC programming. It utilizes artificial intelligent technologies such as knowledge base, blackboard system and machine learning to extract machineable features, and proposes and selects optimal tools for the machining of the given part. Its flexibility and simplicity provide a convenient way to include new techniques and knowledge. The incorporation of this system with other CAD/CAM tools could effectively automate the CNC programming process.

© NRC-CNRC [2003]



Integrated Manufacturing
Systems
14/7 [2003] 593-598

MCB UP Limited
[ISSN 0957-6061]

1. Introduction

CNC machining was one of the most important developments for manufacturing technologies in the twentieth century. It provides high efficiency for mass production of consumer products and flexibility for low quantity production of specialized parts and components. However, one of the bottle-necks of CNC machining is the CNC programming. It requires a skillful programmer who should be not only CAD/CAM and computer-literate but also a machining expert. While CNC tool paths can be generated by most CAD/CAM systems efficiently, planning for the machining process is tedious and almost completely relies on the expertise of the programmer. Decisions such as tools used for roughing and for finishing, number of passes and sequence of cuts are completely dependent on the knowledge of the programmer. Recent publications have started to address this shortcoming of CNC programming and many research works have been conducted to find ways to reduce this deficiency but they mostly dealt with specific environments and conditions (Chin, 1988; Mantyla, 1989; Martin 1989; Pande and Prabhu, 1990; Budde and Imbusch, 1992; Norton, 1993; Yeo, 1995; Fuh *et al.*, 1995; Christensen and Mogensen, 1995). This paper proposes an intelligent process-planning system based on a generic methodology and algorithm to automate and optimize the planning process for CNC machining.

The system utilizes the knowledge base, blackboard system and generic learning technologies. The knowledge base captures the experience of the programmer and facts relevant to the machine tools. The combination of the blackboard system, learning structure and algorithm defines the process plan and optimizes it for the

operation. The implemented system would be very flexible. It has the capability to store and utilize new and specialized knowledge and facts or rules. Optimization is achieved within the defined knowledge set. With the capability and flexibility to expand and learn, the system could fully automate and optimize the CNC planning process within its given machining domain.

2. The system

The system divides the CNC planning process into three modules. The first module "Feature extraction" is a knowledge base (Waterman, 1986, p. 392) that captures and stores part features, common machining techniques and logics for the given CNC machine. The second module "Tool competition" is a blackboard system (Waterman, 1986, pp. 388-9) that considers each cutting tool preset on the CNC machine as a "competitor". It competes for the "job" to machine the part that is posted on to the blackboard. The third module "Tool optimization" is a learning system (Tanimoto, 1990) that evaluates and selects the optimal cutting tools from the successful competitors to machine the part. An optimal cutting tool is selected for each pass of the tool path and therefore the tool path is optimized.

2.1 The feature extraction module

The core of this module is a knowledge base that consisted of a series of parallel nodes. Each of these nodes represents a family or class of machinable parts (or sections of a part). Rules, facts, topological machining logics and constraints for the machining of the part are stored in a tree under the node. The leaf of the tree is the feature to be machined. To plan a machining job, the part

The Emerald Research Register for this journal is available at
<http://www.emeraldinsight.com/researchregister>



The current issue and full text archive of this journal is available at
<http://www.emeraldinsight.com/0957-6061.htm>

to be machined is divided into features according to their machining criteria and procedures. Subsequently, the features are extracted in a topological order for the actual machining. The granularity of feature or part division can be adjusted for specific preference and application with appropriate design and implementation. The knowledge base S is a set of tree structures that represents different classes of parts:

$$S = \{x_1, x_2, x_3, \dots, x_n\},$$

where $x_i \cap x_j = \phi$; $i, j = 0, 1, 2, \dots, n$; $i \neq j$ and x_i is the root of a tree such that:

$$x_i = (y_1, y_2, \dots, y_k),$$

where y_i is a sub-tree that represents a sub-assembly of features or a leaf that represents a feature to be machined, and k is the number of sub-assemblies or features.

The output F of this module is a set of m ordered pairs of machinable feature z_i and updated blank configuration b_i for the part to be machined:

$$F = \{(z_1, b_1), (z_2, b_2), (z_3, b_3), \dots, (z_m, b_m)\};$$

and

$$z_i \cap z_j = \phi; i \neq j; b_p \in b_q; b_q \notin b_p;$$

$$p < q; i, j, p, q = 0, 1, 2, m.$$

The operation starts by setting z_0 to null, b_0 to the blank and p_0 to the part. Then:

$$z_0 = \phi;$$

$$b_0 = \text{blank_configuration; size, material, shape, etc.}$$

$$P_0 = \{f_1, f_2, \dots, f_m\}; \text{ the part, represented by features}$$

$$z_{i+1} = g(S, p_i);$$

$$b_{i+1} = b_i + z_i;$$

$$P_{i+1} = P_i - z_i;$$

$$\text{and } f_i \subseteq z_j; i, j = 0, 1, 2, \dots, m,$$

where g is a function (the inference engine) that extracts the machinable feature f from the part P according to the machining topology and knowledge stored in S and outputs it along with the specific rules (if there are any) to z . It can be as simple as:

$$g(S, p_i) = \text{extract_leaf}(x_i \leftarrow p_i \text{ or } f_j \in S);$$

The procedure reiterates until all the features are extracted.

2.2 The tool competition module

The main element of this module is a blackboard or bulletin board structure where the feature and requirements extracted from the feature extraction module are posted. Each of the cutting tools preset on the machine is represented by a "vendor" (a bidder to the job listed on the bulletin-board) or tool-node that will "compete" to machine

the feature based on its capabilities and constraints. Each tool-node is equipped with a knowledge base that stores these capabilities and constraints. The structure of the knowledge base is similar to the one in the feature extraction module. The capabilities are represented by a set of parallel nodes and their constraints are stored in tree structures under the nodes. A tool T is represented by a set of k capabilities C :

$$T \supset C;$$

$$C = \{c_1, c_2, c_3, \dots, c_k\},$$

where $c_i \cap c_j = \phi$; $i, j = 0, 1, 2, \dots, k$; $i \neq j$ and c_i is the root of a tree such that

$$c_i = (r_1, r_2, \dots, r_n),$$

where r_i is a sub-tree that represents an intermediate constraint or a leaf. The leaf could be restrictions, limits of the functional/operational range, quantified finishing qualities, etc.

To machine a feature F being posted in the bulletin board, tool T examines its capabilities and decides whether to compete for the machining work or not. The output Q of this module would be a set of tools that can machine F .

The algorithm is relatively simple for this process:

$$F = (f, b);$$

$$Q = \phi;$$

$$t_i(f, b) = \begin{cases} T_i; & \text{if } c_i \rightarrow f, \text{ i.e. } T_i \text{ is to compete} \\ \phi; & \text{Otherwise; i.e. } T_i \text{ is to compete} \end{cases}$$

$$Q = Q + t_i(f, b);$$

$$\text{for } i = 1, 2, \dots, m,$$

where f is the feature to be machined, b is the current configuration of the blank, t_i is the inference engine of the i th tool that decides if T_i is capable to machine F and m is the number of tools preset for the machine.

2.3 The tool optimization module

The tool optimization module works in conjunction with the tool competition module to select optimal tools for the machining of the part. At this early design of the system, the environment for the evaluation is the simulation of machining path/pass of the selected tool and the goal is to achieve the fastest cycle time. A modified stochastic learning automata (SLA) (Narendar and Thathatchar, 1989) structure with generic learning algorithm is chosen to be the foundation of the implementation because of its flexibility and simplicity. Future design of the learning module will include multiple goals in various properties such as surface finishing, machining

dynamics, tool characteristics, tool longevity, etc. The standard SLA structure is defined as the following quintuple:

$$SLA = \{\alpha, \beta, p, T, c\},$$

where α and β represent the sets of input and output states, p is the corresponding probability vectors, T is the learning algorithm and c is the corresponding penalty probability defining the environment. The modified SLA for this system is much simpler with only three elements:

$$SLA = \{\beta, P, T\},$$

where β is the output state, P is the probability vector for the cutting tools of the machine and T is the learning algorithm. A tool is represented by an element p_i of the probability vector P :

$$P = \{p_1, p_2, \dots, p_n\},$$

where n = number of tools preset on the machine that is qualified for the job.

T evaluates the return from the simulation and updates P accordingly. After the i th iteration, T updates P^i to P^{i+1} for the next simulation based on β^i :

$$P^{i+1} = T\{P^i, \beta^i\}.$$

The expansion of T can be described in two reactions to counter β^i .

Favorable reaction is when the return from the i th simulation is positive and k th tool was chosen:

$$p_j^{i+1} = p_j^i - ap_j^i; \quad \forall j, j \neq k, \quad 0 < a < 1$$

$$p_k^{i+1} = p_k^i + \sum_{j=1, j \neq k}^n ap_j^i.$$

Unfavorable reaction is when the return from the i th simulation is negative and k th tool was chosen:

$$p_j^{i+1} = p_j^i + \left\{ \frac{b}{(n-1)} - bp_j^i \right\};$$

$$\forall j, j \neq k, \leq 0 < b < 1$$

$$p_k^{i+1} = p_k^i - \sum_{j=1, j \neq k}^n \left\{ \frac{b}{(n-1)} - bp_j^i \right\},$$

where a is the reward parameter and b is the penalty parameter. They can be strategically set to offset the weights of the favorable or unfavorable return according to the desired scenario.

Initially, all qualified tools are set to have equal probabilities so that they would have an equal chance to be chosen to perform the next pass of the cutting path. Based on the result from the simulation, the tool returning

the best performance (e.g. shortest cycle time) is chosen for the pass and its probability is asymptotically increased, while probabilities for other tools are decreased. On the other hand, if the performance of the chosen tool is degraded from its previous pass, the opposite actions are applied. Note that the reward would be greater than the penalty, if the penalty parameter were set to equal the reward parameter. This gives the degraded tool other chances to prove itself until its probability is penalized to below the level of the others. This tool selection iterates until the path is completed.

2.4 Integration

The three modules described work cooperatively from extracting the machinable features to generating the optimal CNC programming plan. While the modular nature of the system permitted the implementation of these modules individually, an effective integration of these modules to achieve the planning goal is needed. The classical goal of integration of modular systems is high cohesion but low coupling. This allows the autonomous operation of each module and lessens the interferences between modules. The design of these modules follows this principle and the goal of integration can be achieved.

The operation and integration of the feature extraction module are straightforward. It represents and stores the machinable features and the updated blank configurations independently of other operations. The input could be a CAD model of the part to be machined and/or specific preferences from the operator or programmer. The output is a simple set of feature and updated-blank pairs. Hence the integration of this module into the system is simple. The capturing techniques and data representation of the input are outside the scope of this paper. Relevant information will be described in future publications.

The integration between the tool competition module and the tool optimization module is also simple but with an added element. Their operations must be synchronized in order to generate an optimal path with multiple passes. The input of the tool competition module is the feature and updated blank pair from the feature extraction module. It outputs a set of capable tools that can machine the given feature that in turn becomes the input for the tool optimization module. The tool optimization module examines each tool in the set for each pass in the cutting path of the feature and outputs the best tool for that pass. These cooperative

operations iterate until the feature is completely machined. While the generated tool set is being examined, the tool competition module can process the next feature from its input. However, it must output the next capable tool set in synchronization with the input of the tool optimization module to reduce storage and processing overheads. This tool competition and optimization processes propagated to all the features of the part to be machined. Integration of these two modules into the system can then be carried out as a unified module. Figure 1 shows the schematics of the integration.

3. Machining parts

To machine a part, the feature extraction module divides the part into a number of machineable features and arranges them into an ordered set along with the corresponding blank configurations according to the machining topology or operational preferences. For each of this ordered pair, the tool competition module recommends a set of capable tools to machine the feature and the tool optimization module, through simulation, evaluates and selects the optimal tools among the recommended tools. The process of tool recommendation and selection iterates until tools for all the features are selected. The outcome is an optimal plan for the CNC programming. Figure 2 shows the schematic of the operation of the system.

The following is an example to illustrate the operation of this intelligent planning process for CNC programming and to verify its feasibility.

The part: 2in \times 2in long cylindrical sleeve with a 1½in \varnothing through hole:

$$p_0 = \{f_1 = \text{bore}, f_2 = \text{o.d.}, f_3 = \text{front_face}, f_4 = \text{back_face}\};$$

The blank: 2¼in $\varnothing \times$ 4in cold rolled steel bar with 1½in chucking length.

$B = \text{as_described};$

The machine: two-axis CNC lathe with eight-tool turret ATC; spindle speed and feed

rate are negligible for this example. The K-base:

$$S = \left\{ x_1 \left(\begin{array}{l} \text{tubular_part;} \\ \text{rough_cut;} \\ z_1 = \text{bore,} \\ z_2 = \text{o.d.,} \\ z_3 = \text{front_face} \end{array} \right), \right. \\ \left. y_2 \left(\begin{array}{l} \text{finish_cut;} \\ z_4 = \text{front_face,} \\ z_5 = \text{bore,} \\ z_6 = \text{o.d.,} \\ z_7 = \text{back_face,} \\ z_8 = \text{parting} \end{array} \right), \dots, x_n(\dots) \right\};$$

Preset tools: 1in \varnothing drill with 4in drilling depth:

$$T_1 = \left\{ c_1 \left(\begin{array}{l} \text{drill_hole;} \\ r_1 = |1'' \phi \text{ hole} \\ r_2 = \leq 4'' \text{ deep} \end{array} \right) \right\}.$$

½in $\varnothing \times$ 4in finish-cut boring bar; optimal cutting depth 0.005-0.01in:

$$T_2 =$$

$$\left\{ c_1 \left(\begin{array}{l} \text{machine_through_bore;} \\ r_1 = \text{requires } \geq 0.625'' \varnothing \text{ existing hole,} \\ r_2 = 0.005''\text{-}0.01'' \text{ optimal cutting depth,} \\ r_3 = \text{bore } \leq 4'' \text{ long,} \\ r_4 = \text{finish cut quality; } 0'' \text{ under_}\phi\text{-size} \end{array} \right) \right\}.$$

¾in $\varnothing \times$ 4in rough-cut boring bar; optimal cutting depth 0.02-0.05in:

$$T_3 =$$

$$\left\{ c_1 \left(\begin{array}{l} \text{machine_through_bore;} \\ r_1 = \text{requires } \geq 0.875'' \varnothing \text{ existing hole,} \\ r_2 = 0.02''\text{-}0.05'' \text{ optimal cutting depth,} \\ r_3 = \text{bore } \leq 4'' \text{ long,} \\ r_4 = \text{rough cut quality; } \cong 0.01'' \text{ under_}\phi\text{-size} \end{array} \right) \right\}.$$

Left side rough-cut tool; optimal cutting depth 0.02-0.1in:

$$T_4 =$$

$$\left\{ c_1 \left(\begin{array}{l} \text{machine_o.d.;} \\ r_1 = 0.02''\text{-}0.1'' \text{ optimal cutting depth,} \\ r_2 = \text{rough cut quality; } \cong 0.01'' \text{ over_}\phi\text{-size} \end{array} \right) \right\};$$

Left side finish-cut tool; optimal cutting depth 0.005-0.02in:

Figure 1

Integration of the feature extraction, tool competition and tool optimization modules

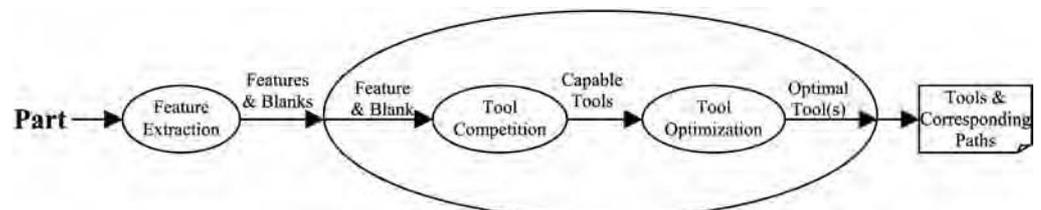
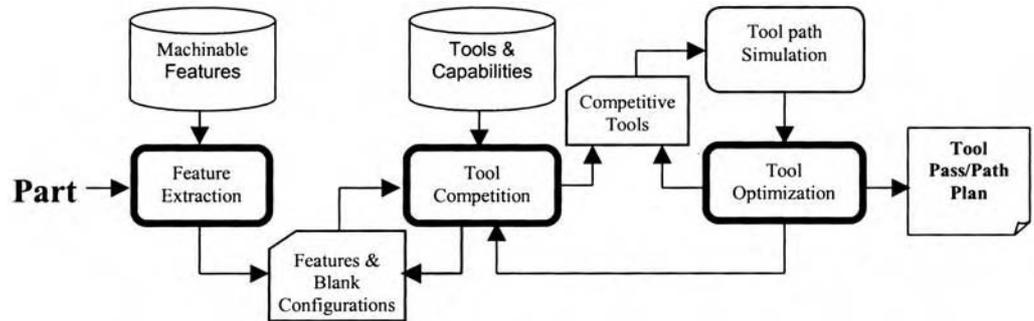


Figure 2
 Operation of the intelligent process-planning system



$$T_5 =$$

$$\left\{ c_1 \left(\begin{array}{l} \text{machine o.d.;} \\ r_1 = 0.005''\text{-}0.02'' \text{ optimal cutting depth,} \\ r_2 = \text{rough cut quality; } 0'' \text{ over } \phi\text{-size,} \end{array} \right) \right\}$$

$1/4 \times 3/4$ in parting tool:

$$T_6 =$$

$$\left\{ c_1 \left(\begin{array}{l} \text{parting or grooving and facing both sides;} \\ r_1 = 1/4'' \text{ cutting width,} \\ r_2 = \leq 3/4'' \text{ cutting depth} \end{array} \right) \right\}$$

With this set-up, the feature extraction module returns the set:

$$F =$$

$$\{(z_1, b_1), (z_2, b_2), (z_3, b_3), (z_4, b_4), (z_5, b_5), (z_6, b_6), (z_{7,8}, b_7)\};$$

$$\text{with } b_{i+1} = b_i + z_{i-1}; \quad i = 1, 2, \dots, 8; \quad z_0 = \phi; \quad b_1 = b_0.$$

For each of the (z_i, b_i) , the tool competition and tool optimization modules work together to define the optimal set of tool-passes. Each tool-pass is represented by the ordered pair $(Tool_i, Pass_i)$ or (T_i, P_i) . The complete set of tool-passes generated by the system for the machining of the cylindrical sleeve is

$$CNC_Plan =$$

$$\{(T_1, P_1), (T_6, P_2), (T_3, P_{3,4,5,6}), (T_3, P_7), (T_4, P_8),$$

$$(T_4, P_9), (T_6, P_{10}), (T_2, P_{11}), (T_5, P_{12}), (T_6, P_{13})\},$$

where:

P_1 = drilling;

P_2 = roughing front face;

P_{3-6} = rough-boring at 0.05in cutting depth each;

P_7 = rough-boring at 0.045in cutting depth;

P_8 = roughing o.d. at 0.1in cutting depth;

P_9 = roughing o.d. at 0.02in cutting depth;

P_{10} = finishing front face;

P_{11} = finishing bore;

P_{12} = finishing o.d.;

P_{13} = parting and finishing back face;

The plan can be incorporated into CAD/CAM systems for the CNC programming of the actual machining.

4. Conclusion

An automated process-planning system can reduce the bottle-neck burden of CNC machining. The system presented utilizes knowledge base, blackboard system and machine learning technologies to automatically plan and optimize the CNC programming process. Its flexibility and simplicity allow the inclusion of meta-knowledge and operation-specific preferences, as well as dynamic updating. Users can adjust the granularity of the knowledge and information input into the system to satisfy their needs and preferences. It can also be incorporated or integrated into CAD/CAM systems for direct CNC programming. Future expansion of this system will include the development of new machining and automation technologies and relevant knowledge such as high-speed machining, motion dynamics, geometric error compensation, characteristics of machines, tools and materials, etc. The intelligent process-planning system will accommodate and facilitate the implementation and operation of these technologies and processes. It is a true, fully automatic planning system for CNC programming and automation controls within its knowledge domain. It can evolve and grow dynamically along with the acquisition of new knowledge and techniques.

References

- Budde, W. and Imbusch, K. (1992), "EXAPT process planning and NC planning with database-supported management of production data", *IFIP Transactions B*

- (*Applications in Technology*), Vol. B-3, pp. 119-30.
- Chin, S.C. (1988), "Developing a feature-based knowledge system for CAD/CAM integration", *Computers & Industrial Engineering*, Vol. 15, pp. 34-40.
- Christensen, G.K. and Mogensen, O.B. (1995), "Backward form feature recognition and removal for an automatic CNC-programming system – BCAM", *Advanced CAD/CAM Systems – State-of-the-Art and Future Trends in Feature Technology*, Chapman & Hall, London, pp. 205-16.
- Fuh, J.Y.H., Ji, P. and Zhang, Y.F. (1995), "Future development trends in CAM/CAPP-NC systems", *International Journal of Computer Applications in Technology*, Vol. 8 No. 3/4, pp. 203-10.
- Mantyla, M. (1989), "Feature-based product modeling for process planning", *Organization of Engineering Knowledge for Product Modeling in Computer-Integrated Manufacturing*, 2nd Toyota Conference, Elsevier, Amsterdam, pp. 303-24.
- Martin, J.M. (1989), "A strategy for NC programming", *Manufacturing Engineering*, Vol. 102 No. 2, February, pp. 82-4.
- Narendra, K.S. and Thathatchar, M.A.L. (1989), *Learning Automata – An Introduction*, Prentice-Hall International, Englewood Cliffs, NJ.
- Norton, N. (1993), "Controlling the shop floor", *Manufacturing Engineer*, Vol. 72 No. 6, December, pp. 272-5.
- Pande, S.S. and Prabhu, B.S. (1990), "An expert system for automatic extraction of machining features and tooling selection for automats", *Computer-Aided Engineering Journal*, Vol. 7 No. 4, August, pp. 99-103.
- Tanimoto, S.L. (1990), *The Elements of Artificial Intelligence*, Computer Science Press, W.H. Freeman & Company, New York, NY, pp. 311-13.
- Waterman, D.A. (1986), *A Guide to Expert Systems*, Addison-Wesley Publishing Company, Reading, MA.
- Yeo, S.H. (1995), "A multipass optimization strategy for CNC lathe operations", *International Journal of Production Economics*, Vol. 40 No. 2-3, August, pp. 209-18.

Agent-based architecture for manufacturing system control

C.K. Fan

Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong

T.N. Wong

Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong

Keywords

Flexible manufacturing systems, Object-oriented programming, Modelling, Machine oriented languages

Abstract

A flexible manufacturing system (FMS) is a complex manufacturing system and it demands a robust control software for its scheduling, planning and control functions. This paper describes the development of an agent-based infrastructure for the control of a cellular FMS. The FMS in this project is a flexible assembly cell (FAC), comprising two assembly robots and a conveyor system. The aim is to establish a multi-agent control system with good expandability and to be able to cope with dynamic changes in the FAC. The proposed agent-based FAC control system comprises a collection of agents implemented in a distributed control network. The approach of the agent design is based on the object-oriented modelling technique. According to the proposed control architecture, a standard agent template has been designed for the establishment of individual agents in the agent-based system.

1. Introduction

Flexible manufacturing systems (FMS) are able to manufacture products in small quantities with an efficiency comparable with that of high volume production. An FMS consists of flexible manufacturing units (robots, CNC machines, etc.), automated material-handling devices such as automatic storage and retrieval system (AS/RS) and automatic guided vehicle (AGV). The equipment and workstations are linked to a computerized control unit. Traditionally, manufacturing systems are commonly established with centralized hierarchical control software. In hierarchical control structure, control decisions and decomposition of the tasks are distributed downward to the lower hierarchy and the information of individual components is sent upward to components in the higher hierarchy. The tight working relationship between components is hardly changed, even if one of the components in the same hierarchy is added or removed. However, the control software of a hierarchical control architecture is in general difficult to develop and implement for advanced manufacturing systems such as an FMS. It is also not flexible enough to cope with the dynamic changes of manufacturing processes. This is due to the lack of a dynamic model to deal with the real time product and process changes (Naylor and Volz, 1987).

Since the advent of computers, the rapid progress of computer applications in manufacturing has been focused on achieving autonomous control (Barber, 1999; Huang and Nof, 2000), distributed control (Yeung and Moore, 1995; Hiroki *et al.*, 2000), intelligent manufacturing (Sobolewski, 1996), conceptual and system integration of the manufacturing system (Siemiatkowski and

Przybylski, 1997; Harish and Peihua, 1997; Gong and Hsieh, 1997), fast response to the external environment and adaptability to changes in the manufacturing system.

In recent years, researchers have attempted to apply agent technology to manufacturing enterprise integration, supply chain management, manufacturing planning, scheduling and control, materials handling, and holonic manufacturing systems. For applications in decentralized manufacturing control, agents communicate and negotiate with one another to perform the operations based on the available local information. The overall performance of a decentralized manufacturing system depends on the negotiations among the agents and the quality of the data used to make these decisions.

This paper describes the development of an agent-based infrastructure for the control of a flexible assembly cell (FAC). The proposed agent-based FAC control system comprises a collection of agents implemented in a distributed control network. The approach for designing agent-based FAC control system is based on the object-oriented modelling technique. According to the proposed control architecture, a standard agent template is proposed to implement individual agents in the agent-based system.

2. Agent applications in FMS control

For FMS design, researchers have recognised the importance of establishing an intelligent and distributed framework in the development of the control system. The application of agents has received much attention in the distributed AI (DAI) community. The benefits of using agents in problem-solving and decomposition ability



Integrated Manufacturing Systems
14/7 [2003] 599-609

© MCB UP Limited
[ISSN 0957-6061]
[DOI 10.1108/09576060310491955]

The Emerald Research Register for this journal is available at
<http://www.emeraldinsight.com/researchregister>



The current issue and full text archive of this journal is available at
<http://www.emeraldinsight.com/0957-6061.htm>

can be demonstrated in the works by Lesser (1995) and Fraile *et al.* (1999). According to Kouiss *et al.* (1997), agents are:

... the entities in a system which can perform a task in an intelligent way but it can make decisions with other agents in the system. The addition or removal of an individual agent will not affect the overall system performance.

In agent-based applications, agents have the capabilities in communication, negotiation and thinking (Shoham, 1993). Researchers have also identified the potential of DAI in solving various complex manufacturing system problems with multi-agent systems (MAS) (Basran *et al.*, 1997; Nagata and Hirai, 1994; Franklin and Grasser, 1996; Noronha and Sarma, 1991).

Agent models are built and implemented in order to attain the flexibility, reusability and fast response to both internal and external uncertainties in the shopfloor environment (Monostori and Kadar, 1999). Agent-based FMS control systems are mostly established in the hybrid structure to enhance the inter-communication, negotiation and cooperation. This is a compromise between the hierarchical and heterarchical control structures. The main purpose of the agents is to obtain global objective from local agent solutions (D'Ambrosio *et al.*, 1996). The use of agents enriches the reliability and flexibility of manufacturing-planning and scheduling functions in the dynamic manufacturing system (Maturana and Norrie, 1996). The agent structure provides reconfigurability to changes. The agents also provide the fault tolerance capability to the manufacturing cell. This capability is achieved by the re-allocation of the resources. Other examples of applying agent technology in the development of FMS control system are found in Miyashita (1998), Sousa and Ramos (1998, 1999) and Iima *et al.* (1999). Peng *et al.* (1998) used agents in applications in manufacturing design, planning, scheduling and simulation in a distributed intelligent environment. Butler *et al.* (1995) applied the agents to represent manufacturing resources such as workers, cells, machines, tools and AGV to improve the planning, scheduling and execution of the FMS. Fox *et al.* (1993) incorporated an agent-based scheduler into FMS manufacturing-planning and scheduling systems. Agents with bidding mechanism are proposed by Shaw and Fox (1993) in FMS control system.

Recently, researchers have attempted to apply intelligent agents and object-oriented modelling in developing FMS control systems (Kadar *et al.*, 1998; Lei *et al.*, 1998). The object-oriented modelling methods are

appropriate modelling and implementation approaches to represent the flexibility, real time response of the complex agent-based manufacturing control systems (Reveliotis, 1999; Gambin *et al.*, 1999). The object-oriented method is able to reduce the design complexity and increase the reusability of the FMS control software.

3. Configuration of the FAC

Figure 1 depicts the configuration of the FAC. The FAC comprises two Adept SCARA-typed assembly robots and a conveyor loop with four conveyors. The two robots are controlled by a single Adept programmable controller. Assembly operations and material-handling tasks can be handled by these two robots. Stoppers are located at the front of the two robots (the area is called the robot-processing area), so that pallets are clamped for robot processing. The robot controller is equipped with two serial communication ports for communication.

There are four conveyors in the FAC. They form a loop connection for pallets transportation. Sensors, pneumatic cylinders, signal indicators are controlled by a programmable logic controller (PLC). The PLC is employed as the central processing unit for the motion control of the conveyor system. A ladder diagram is used to prepare the control program for the control logics of the transportation system. The PLC is equipped with a serial communication port for communication.

In the current implementation of the prototype control infrastructure for the FAC, a network of four computers is used. That is, there is no single central control computer. The control software is established as a multi-agent system with different agents installed in different computers in the network. The computers are configured into a local area network with the TCP/IP communication interfaces. Within the system, serial communication connections are the basic interfaces for connecting the hardware devices and the computers, while the agents communicate in the network through the TCP/IP. Using this approach, it is straightforward to connect the control system, and hence the assembly cell itself, to the Internet to establish Web-based applications.

According to the design of the FAC, only six pallets are allowed to co-exist in the system at the same time. During the assembly operations, the working area over the pallets is restricted by the working envelope of the robots. The payload is also

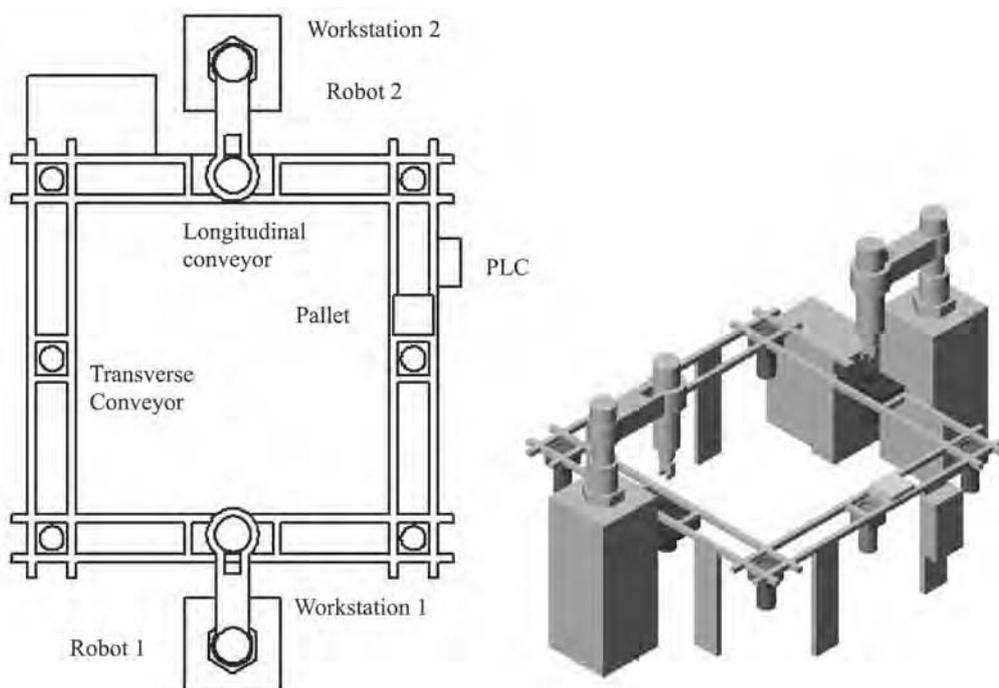
limited by the default setting of the robots. TCP/IP and serial communication ports are the two main communication media between agents. Error detection of the network status have also been built to prevent the loss of data transfer of the TCP/IP communication.

4. Agent in the FAC control system

The prototype agent-based system consists of a number of individual agent units corresponding to specific functionalities in the physical configuration of the FAC. They are listed as follows:

- *Scheduling agent.* The scheduling agent is responsible for coordinating the tasks between the pallet identity agent, the barcode reading agent and the conveyor control agent. This agent links up with the central database to retrieve and save the overall system information. It is embedded with a specific scheduling algorithm to cater for the scheduled instruction in the FAC.
- *Pallet identity agent.* The pallet identity agent is responsible for the provision of the local pallet information and pallet position information from the sensor signal interpreter agent. The information is sent to the scheduling agent, barcode reading agent and conveyor agent. The agent has its own pallet information database.
- *Barcode-reading agent.* The barcode-reading agent is responsible for the provision of the local barcode information from the barcode signal-polling agent. This information is sent to the scheduling agent, the pallet identity agent and the conveyor agent. The agent has its own barcode information database.
- *Conveyor control agent.* The conveyor control agent is responsible for the provision of the local solution from the two sub-level conveyor agents. The information is required by the scheduling agent, the pallet identity agent and the barcode-reading agent. The agent has its barcode information database.
- *Sensor signal interpreter agent.* The sensor interpreter agent is responsible for the provision of the local sensor information from the sensor position translation agent. The agent co-operates with the pallet identity agent to identify the pallet locations. The agent has a common pallet information database, sharing the information with the pallet identity agent.
- *Barcode signal-polling agent.* The barcode signal-polling agent is responsible for the provision of the local barcode information from the barcode readers. The information pending from the barcode readers will be read to the agents in default time intervals.
- *Sensor position translational agent.* The sensor position translational agent is

Figure 1
Configuration of the FAC



responsible for the provision of the sensor signal interpreter agent. The agent translates the raw signal into the pallets' positional readable format to the controller.

- *Robot control agent.* The robot control agent is responsible for the provision of the local robot manipulation. The agent also assigns the required control program to the robot to meet the order requirements.
- *Order-handling agent.* The order-handling agent provides the interface for the cellular FMS connecting with the external order-requesting agent through the Internet communication or mobile communication media.

Each individual agent has its own ability to monitor, evaluate and solve problems. Intelligent algorithm is designed to represent the specific behavior of the agent. Implementation of these agents with the algorithms in the system forms specific autonomous process for the assembly cell. Figure 2 depicts some of the agents in the control system. In this example, four agents (scheduling agent, pallet identity agent, conveyor agent, and robot agent) are used to form the core working elements in the system.

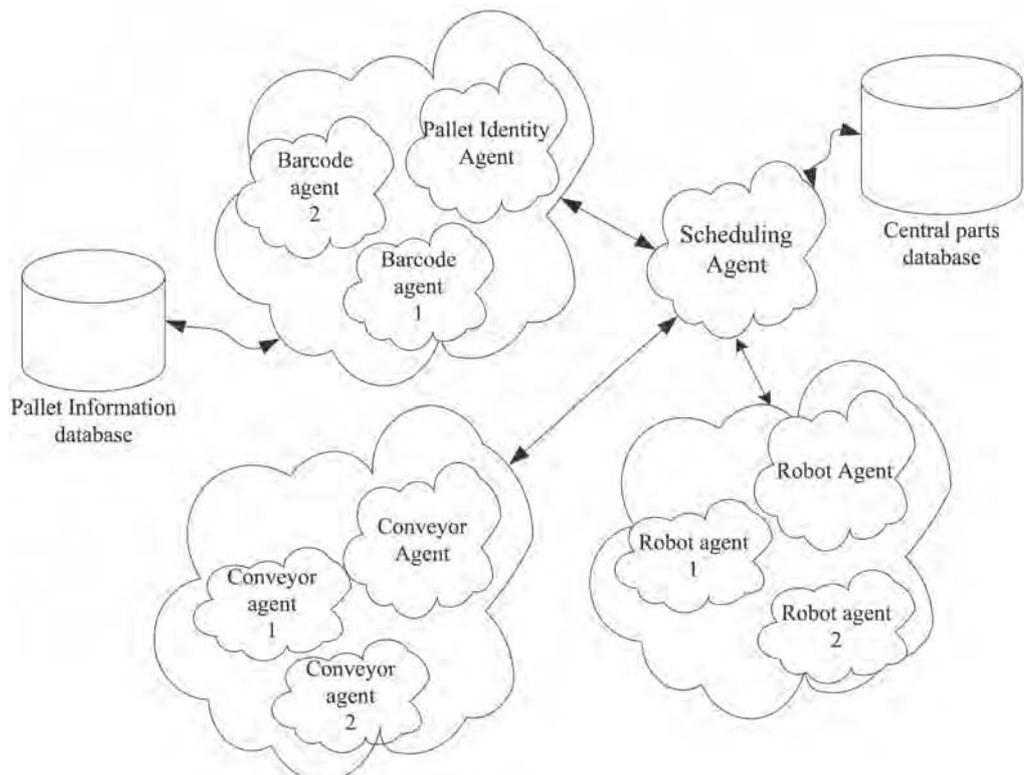
5. Object-oriented design and modelling

The object-oriented approach is adopted in the development of the agent-based FAC control system. Basically, agents in the system are modelled by specific object relationship, class attribute and behaviour of the physical components or operating functions in the FAC.

Object-oriented modelling and analysis have been a popular choice for developing real time distributed control applications (Nett *et al.*, 1998). The applications benefit from the fast processes, quick response, and complex interaction with synchronous inputs and environmental adaptability between objects in these software design models. By using the fundamental ideas of objects, classes, methods, messages, encapsulation, inheritance, polymorphism (Becker *et al.*, 1994), a very complex system can be decomposed into many simple systematic and object-oriented structures.

Three types of system models are usually established in the implementation of the object-oriented technique: the object model, the dynamic model, and the functional model. Application of an individual model displays an incomplete picture of the system.

Figure 2
Some agents in the example FAC



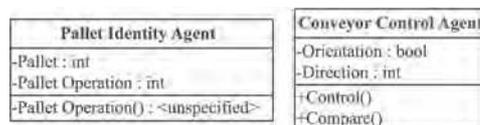
Using the three models together, we can develop a complete model for the FMS control system. The three models-building stages can be seen as different phases for the design and analysis of the system. The object model analysis phase is the stage for finding the static structure objects in the FMS control system and their relationship. The most important objects for the system in the real world will be extracted and added into the object model. The association between the objects will then be identified. The object-oriented design and analysis approach using the object model provides clear pictures for building potential objects and then finding out the messages required to be sent between the objects in both the dynamic and functional models.

5.1 Object model

The object model is the essential model for the construction of classes of objects. With this model, the unified representation of the objects can be constructed into an object-oriented program. Examples of detail object declaration of the object model are illustrated in Figure 3 and the object model diagram (Figure 4) illustrates the relationships between the classes or objects. The agent relationships give the indication for the construction of the object model. The object model provides the program's definition on the objects or classes required for the agents. A group of objects or classes may represent an individual agent but the object model provides the basic object units definition for each element inside the cell. The object model in Figure 4 provides more practical information for the connection between the devices and the logical or controlling units inside the flexible manufacturing cell. The model shows the set of objects exist in the system, together with their corresponding relationship, attributes and operations. As object classes are the essential elements in the object-oriented representation, the object model depicts the static view of the overall agent system and shows the relationship of dependency between classes. In the object model, physical communication is represented by the edge

Figure 3

Detail object declaration in object model in UML



connecting two objects. In the current implementation of the FAC, TCP/IP is used for the connection between the scheduling agent, the conveyor agent, the barcode-reading agent and the pallet identity agent. This interface is used for the controlling functions involving bidding contract, device availability status and problem information reporting. On the other hand, serial communication and direct electronic signal wiring are used in the physical connection between the other agents, to transmit signals involving low level controllers (PLC, special hardwired circuit, etc.), barcode readers and sensors. The structure of the object model shows the preliminary infrastructure of agents. The necessary agents and objects are clearly shown during the system development stage. Additional agents can be built at any time as objects in the system program with proper message interface.

5.2 Dynamic model

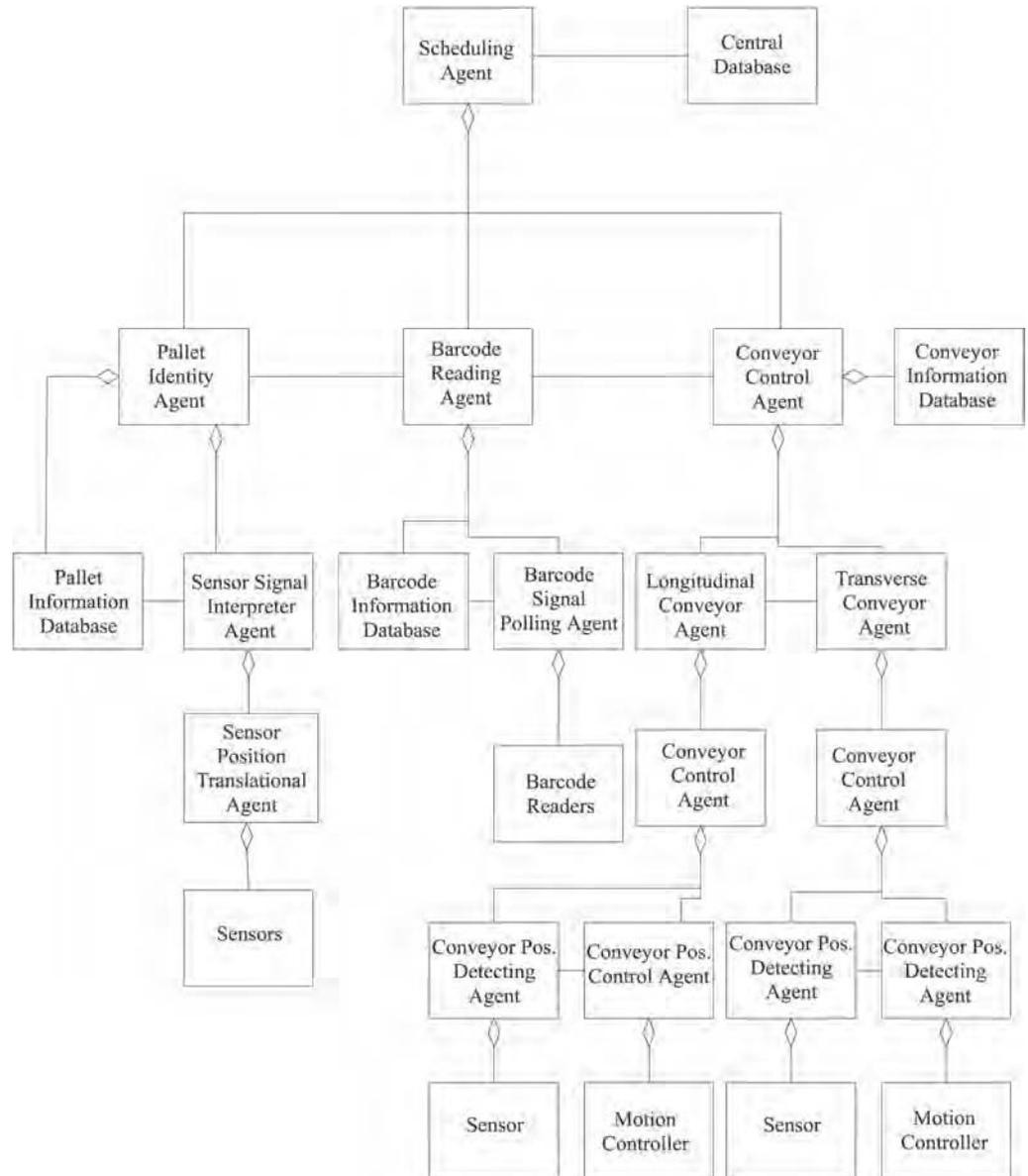
Figure 5 shows the event trace diagram of the system, when the bid request is sent from the scheduling agent. The diagram shows the sequence of the events. It provides the location of the information flow for any particular event. The dynamic model contains information not available in the static object model and use case diagrams. It therefore provides a picture indicating the activities of the FAC in the dynamic time domain. The sequences of the operations for different objects are illustrated in the dynamic model, which can help the system programmer to develop the behaviours of the agents. The origin and destination of a message are shown explicitly in the diagram. The sequence is initialized at the top of the diagram. The scheduling agent requests vacancy resource from both the robot agent and the conveyor agent. If the two agent resources are available, allocated resources will then be distributed to the corresponding agents from the scheduling agent and the process will then be assigned to the equipment agents.

The use case diagram (Figure 6) shows states of the FAC from the user point of view. The diagram presents the set of required actors (manufacturing devices or virtual entities/agents) and relevant processes; therefore, the relations of the inputs to the outputs are clearly shown in this use case diagram.

5.3 Functional model

The functional model provides additional information which is not present in the object model and the dynamic model. Figure 7 depicts the data flow diagram representing

Figure 4
 Object model for the FAC



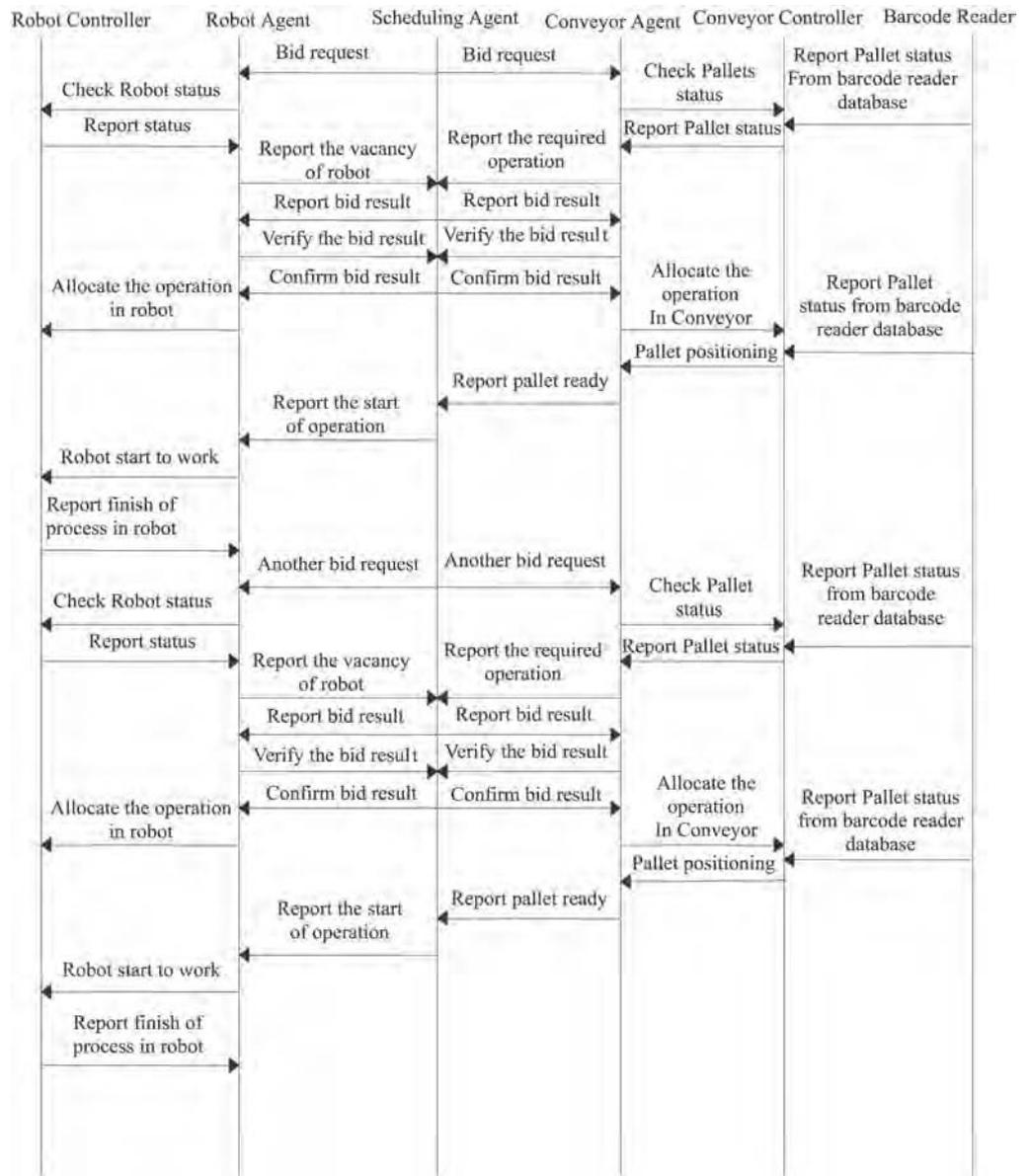
the functional model. It is used to illustrate the output values in the prototype system. The functional model shows the dynamic information flow only, ignoring how and when the information changes in the system.

6. Implementation of the agent system

Corresponding to the implementation of the agent-based infrastructure proposed for the FAC control system, a formalized agent structure is proposed. In this FMS control system, each agent comprises six important layers. They are communication layer, data

layer, preceptor layer, effector layer, agent's logic layer, and error-handling layer. The communication layer deals with the TCP/IP communication task with other agents. The preceptor and effector layers are responsible for the control of the real time shopfloor information. The agent's logic layer is the "brain" of the agent. The degree of autonomy of an agent depends on this layer design structure. Intelligent objects can be added inside this layer to increase the flexibility of the agents. Besides, this layer handles rapid responses to the dynamic changes to the shopfloor environment of the agent. Error handling, such as machine breakdown, or other undesired events, are handled by the

Figure 5
 Dynamic model of the FMS control system



error-handling layer of the general agent structure. Java programming language format is used to illustrate the mapping of individual agent structure to the object-oriented programming structure. Additional layers are reserved for other purposes in the future development of the agent.

Figure 8 shows an example on the mapping of the agent design structure to the Java programming structure. For a new agent, a name should be given for identification. In the example, the name of the new agent is "agentx"; "agentx" can be inherited from an existing agent, "agenty". That is, agenty is the parent of the new agent. All the characteristics of the parent agent can be re-used by this new agent. Additional common

structure of the agent's methods can also be shared by using the keyword "implements". In this example, "agentx" shares the methods of "agentz".

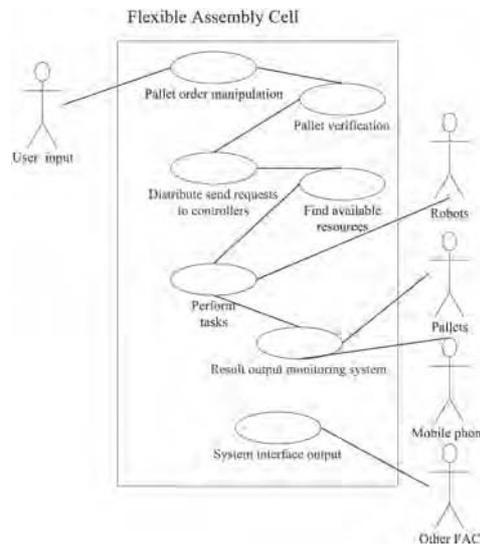
Apart from sharing and inheritance of properties from other agents, it is also possible to create new methods for agents with unique functionalities in the control system. The data layer, preceptor layer and effector layer can be inherited by "agentx" using the object creation syntax:

New_object_name =

new exist_layer_template_object.

Using this syntax, new objects will be created from the template exist_layer_template. The

Figure 6
Use case diagram for the example FAC



newly created object exists in the new agent only and it is independent of other agents. The contents, program logic and execution of the new agent's object will be executed without interference with other agents inherited from the same layer template.

The agent template is the generic agent program structure for the agents throughout this FMS control system. The object or class creation for the agents is similar in object-oriented programming languages. The Java programming example for the generic agent template is introduced to illustrate the generic agent structure for the FMS control system.

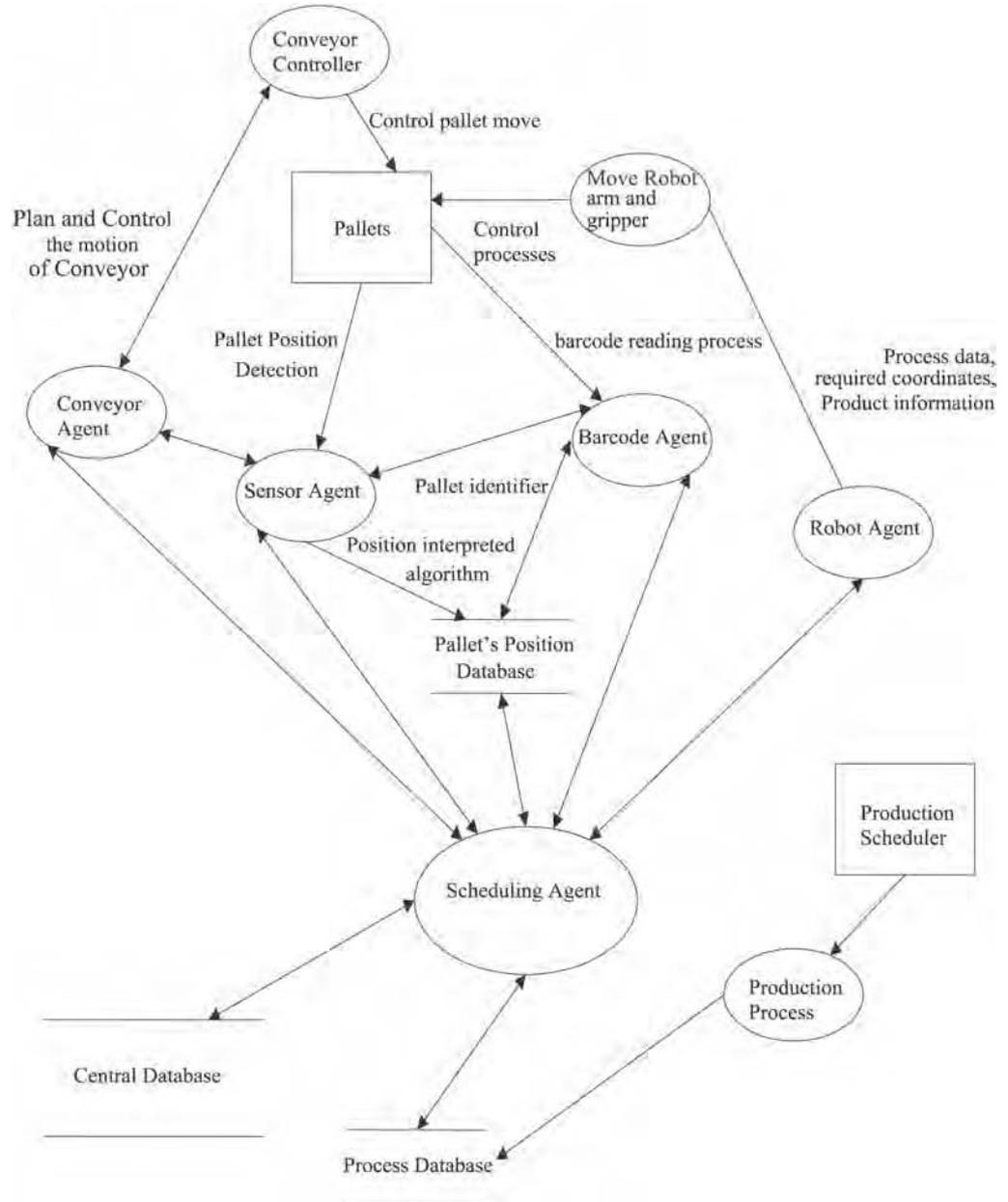
Figure 9 shows an example of the generic agent template in the FMS control system. Line 1 shows the package name for the FMS control software, which is used to define the system program name. The objects inside this package cannot be used by other packages. The agent object is defined in line 2; the RobotControlAgent is the name of the object. The objects, classes and methods below line 2 are encapsulated inside the RobotControlAgent. The underlined statement in line 2 is implemented optionally. Inheritance is achieved in Java using the "extends" keyword. The inheritance of the agents depends on the similarity of the object. If a prototype of the agents is built, agents can inherit from the prototype object by using the "extends" keyword. The keyword "implements" in the statement means the using of the interface object in the agent class "RobotControlAgent". With the "implements" keyword, the interface object can be used by other agents, if it is commonly

used throughout the FMS control system. Lines 4-9 are the contents for the "RobotControlAgent", which is the program entry point for the agent. Agent methods, named "AgentMethod1" and "AgentMethod2" (the methods are the programming code for the operations defined in chapter 4 for the agents) in this example, are then listed. The contents for the agent method are written within the brackets "{}".

By using the object-oriented models in the design and analysis stages, necessary models and interfaces are determined before the actual program codes are developed. This can help to reduce the redundant codes at the development stage. The use of the TCP/IP protocol can also increase the mobility and the communication capability of the agents. Communication is not restricted by the physical environment, and an agent can communicate with the other agents in any location if the TCP/IP connection is installed.

The Java and Visual Basic programming languages are used in the development of this agent-based FAC control system. Both Java and Visual Basic have well developed supports for the TCP/IP communication protocol. It is easy to establish the interface between the shopfloor control and the Internet communication through the TCP/IP interface. In addition, the FAC control system is able to support mobile communication through the Java-based Servlet development tool. The Servlet application provides a powerful interface to communicate with the mobile phone by the wireless application protocol (WAP). It helps to increase the capability of the monitoring and control function by using mobile devices.

Figure 7
 Functional model of the FMS control system



7. Conclusion

In the current implementation of the prototype multi-agent system, agents are installed in different controlling computers in a LAN through the TCP/IP interface. In the corresponding agent control computers, graphic user interfaces (GUI) are established to provide the real-time status of the FAC and the agents. Using the object-oriented models in the design and analysis stages, necessary models and interfaces are determined before the actual program codes are developed. This can help to reduce the redundant codes at the development stage. This can also reduce the

investment cost, because most of the program design problems are found at the early stage of program development. Fast response to the system preceptors (sensors, robot controller messages) is also shown by using distributed agents in the FAC.

For future research, the agent-based infrastructure can be expanded to more complex manufacturing control systems. More objects or agents can be added to the control system. For instance, it is possible to add-in the process-planning agent, product design agent, and machining agent to cope with additional planning and control functions and activities.

Figure 8
 Generic agent template to object-oriented language (Java) structure

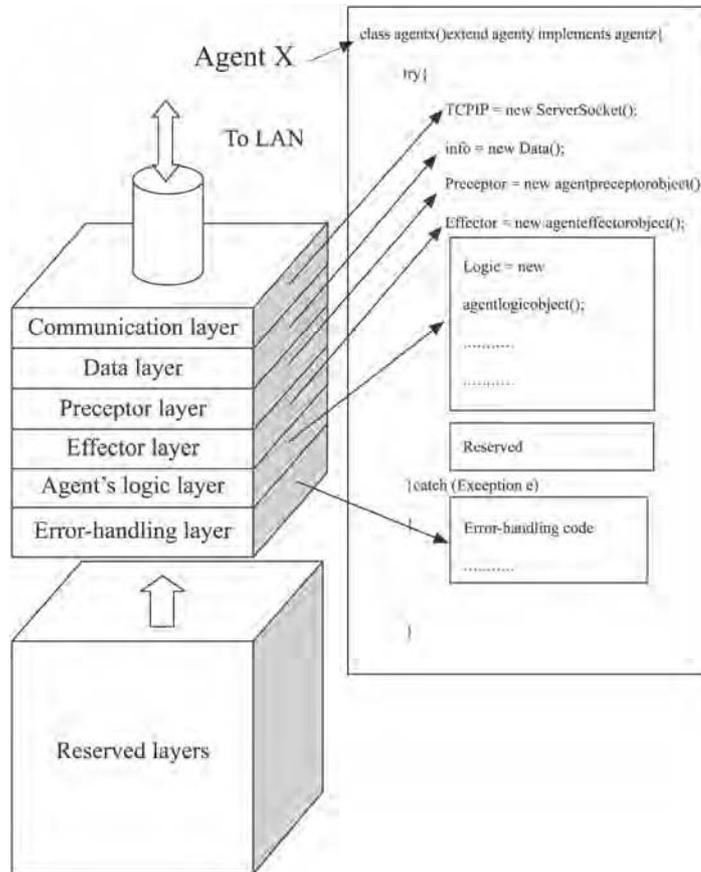


Figure 9
 Generic agent template in Java

```

1 package FMSControlsystem;
2 class RobotControlAgent extends object implements object{
3     parameter initialization;
4     RobotControlAgent {
5         Robotcontrolagentnewobject = new Object;
6         Agent execution statement;
7         .....;
8         .....;
9     }
10    AgentMethod1() {
11        .....;
12    }
13    AgentMethod2() {
14        .....;
15    }
16 }
```

References

Barber, K.S. (1999), "The motivation for dynamic adaptive autonomy in agent-based systems", *Proceedings of the 4th International Symposium on Autonomous Decentralized Systems*, Tokyo, Japan, pp. 402-5.
 Basran, J.S., Petriu, E.M. and Petriu, D.C. (1997), "Flexible agent-based robotic assembly cell", *Proceedings of the 1997 International*

Conference on Robotics and Automation, IEEE, pp. 3461-6.
 Becker, S.A., Beiler, P.J. and Hevner, J. (1994), "Object-oriented box structures for developing complex systems", *Proceedings of the 27th Hawaii International Conference on AR System Sciences, Information Systems, Decision Support and Knowledge-Based Systems*, Vol. 3, pp. 43-52.
 Butler, G., Grogono, P., Shinghal, R. and Tjandra, I. (1995), "Analyzing the logical structure of data flow diagrams in software documents", *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Vol. 2, pp. 575-8.
 D'Ambrosio, J.G., Darr, T. and Birmingham, W.P. (1996), "Hierarchical concurrent engineering in a multi-agent framework", *Concurrent Engineering: Research and Applications*, Vol. 4 No. 1, March, pp. 47-57.
 Fox, M.S., Chionglo, J.F. and Barbuceanu, M. (1993), "The integrated supply chain management system", internal report, University of Toronto, available at: www.ie.utoronto.ca/EIL/iscm-descr.html
 Fraile, J.C., Paredis, C.J.J., Cheung, H.W. and Khosla, P.K. (1999), "Agent-based planning and control of a multi-manipulator assembly system", *Proceedings of the International Conference on Robotics & Automation*, Detroit, MI, May, IEEE, pp. 1219-25

- Franklin, S. and Grasser, A. (1996), "Is it an agent, or just a program?", *Proceedings of the 3rd International Workshop on Agent Theories, Architectures and Languages*, Springer-Verlag, Heidelberg, pp. 54-66.
- Gambin, A.J., Piera, M.A. and Riera, D. (1999), "A Petri nets-based object-oriented tool for the scheduling of stochastic flexible manufacturing systems", *Proceedings of ETFA '99, the 7th IEEE International Conference on Merging Technologies and Factory Automation*, Vol. 2, pp. 1091-8.
- Gong, D.C. and Hsieh, Y.W. (1997), "Conceptual design of shopfloor control information system", *International Journal of Computer-Integrated Manufacturing*, Vol. 10 No. 1-4, pp. 4-16.
- Harish, A.R. and Peihua, G.U. (1997), "Design methodology and integrated approach for design of manufacturing systems", *Integrated Manufacturing Systems*, Vol. 8 No. 3, pp. 159-72.
- Hiroki, O., Weng, J.H. and Hisashi, O. (2000), "Characteristics of distributed autonomous production control", *International Journal of Production*, Vol. 38 No. 17, pp. 4205-15.
- Huang, C.Y. and Nof, Y.S. (2000), "Autonomy and viability – measures for agent-based manufacturing systems", *International Journal of Production*, Vol. 38 No. 17, pp. 4129-48.
- Iima, H. et al. (1999), "An autonomous decentralized scheduling algorithm for a scheduling problem in a metal mold assembly process", *Journal of Intelligent Manufacturing*, Vol. 10 No. 2, pp. 161-7.
- Kadar, B., Monostori, L. and Szelke, E. (1998), "An object-oriented framework for developing distributed manufacturing architectures", *Journal of Intelligent Manufacturing*, Vol. 9, pp. 173-9.
- Kouiss, K., Pierreval, H. and Mebarki, N. (1997), "Using multi-agent architecture in FMS for dynamic scheduling", *Journal of Intelligent Manufacturing*, Vol. 8, pp. 41-7.
- Lei, M., Yang, X.H. and Tseng, M.M. (1998), "A CORBA-based agent-driven design for distributed intelligent manufacturing systems", *Journal of Intelligent Manufacturing*, Vol. 9, pp. 457-65.
- Lesser, V. (1995), *Proceedings of 1st International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, CA, 12-14 June, pp. 209-15.
- Maturana, F.P. and Norrie, D.H. (1996), "MetaMorph: an adaptive agent-based architecture", *International Journal of Intelligent Manufacturing*, Vol. 7, pp. 257-70.
- Miyashita, K.C. (1998), "A constraint-based architecture for multi-agent planning and scheduling", *Journal of Intelligent Manufacturing*, Vol. 9, pp. 147-54.
- Monostori, L. and Kadar, B. (1999), "Agent-based control of manufacturing systems", *Proceedings of the 2nd International Conference on Intelligent Processing and Manufacturing of Materials (IPMM '99)*, Vol. 1, pp. 131-7.
- Nagata, A. and Hirai, J. (1994), "Distributed planning for assembly tasks by multiple manipulators", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3522-9.
- Naylor, A.W. and Volz, R.A. (1987), "Design of integrated manufacturing system control software", *IEEE Transactions on System, Man and Cybernetics*, Vol. 17, pp. 881-97.
- Nett, E., Gergeleit, M. and Mock, M. (1998), "An adaptive approach to object-oriented real-time computing", *Proceedings of the 1st International Symposium on Object-Oriented Real-time Distributed Computing (ISORC '98)*, pp. 342-9.
- Noronha, S.J. and Sarma, V.V.S. (1991), "Knowledge-based approaches for scheduling problems: a survey", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 3 No. 2, pp. 160-71.
- Peng, Y., Finin, T., Labrous, Y., Chu, B., Long, J., Tolone, W.J. and Boughannam, A. (1998), "A significant step towards agent-based enterprise integration for manufacturing planning and execution through experimentation with a real manufacturing scenario involving real legacy MES and schedule", *Proceedings of PAAM'98 – A Multi-Agent System for Enterprise Integration*, London, pp. 323-30.
- Reveliotis, S.A. (1999), "Production planning and control in flexibly automated manufacturing systems: current status and future requirements", *Proceedings of the International Conference on Robotics and Automation*, IEEE, Vol. 2, pp. 214-20.
- Shaw, M. and Fox, M. (1993), "Distributed artificial intelligence for group decision support integration of problem solving, coordination and learning", *Decision Support Systems*, Vol. 9 No. 4, pp. 349-67.
- Shoham, Y. (1993) "Agent-oriented programming", *Artificial Intelligence*, Vol. 60 No. 1, pp. 51-92.
- Siemiatkowski, M. and Przybylski, W. (1997), "A system solution for integration of process planning and control in flexible manufacturing", *Integrated Manufacturing Systems*, Vol. 8 No. 3, pp. 173-80.
- Sobolewski, M. (1996), "Multi-agent knowledge-based environment for concurrent engineering applications", *Concurrent Engineering: Research and Application*, Vol. 4 No. 1, March, pp. 89-97.
- Sousa, P. and Ramos, C. (1998), "A dynamic scheduling holon for manufacturing orders", *Journal of Intelligent Manufacturing*, Vol. 9, pp. 107-12.
- Sousa, P. and Ramos, C. (1999), "A distributed architecture and negotiation protocol for scheduling in manufacturing systems", *Computers in Industry*, Vol. 38, pp. 103-13.
- Yeung, W.H.R. and Moore, P.R. (1995), "A distributed field message controller for intelligent conveyor systems", *Proceedings of the International Conference on Robotics and Automation*, IEEE, pp. 109-13

Further reading

- Dorador, J.M. and Young, R.I.M. (2000), "Application of IDEF0, IDER3 and UML methodologies in the creation of information models", *International Journal of Computer-Integrated Manufacturing*, Vol. 13, pp. 430-45.

Machining feature extraction for casting parts

B.F. Wang

Department of Mechanical Engineering, National University of Singapore, Singapore

Y.F. Zhang

Department of Mechanical Engineering, National University of Singapore, Singapore

J.Y.H. Fuh

Department of Mechanical Engineering, National University of Singapore, Singapore

Keywords

Manufacturing systems, Programming and algorithm theory, Boolean functions, Modelling

Abstract

An approach to extract machining features for casting parts is presented. It is capable of recognizing interacting machining features. There are five phases in the recognition process: Boolean difference of the final part model and the raw part; identification of machining faces (M-faces) from the final part model and the raw part model; decomposition of the removed simple volumes into delta simple volumes based on M-faces; gluing the delta simple volumes into sets of feasible simple volumes based on M-faces; testing. This strategy is process-oriented and feature-independent. It recognizes all features that can be produced by common machining operations in a uniform way and produces alternative sets of machining features.

Nomenclature

Raw part model (*RPT*) = a 3D stock specified by the user.

Final part model (*FPT*) = a 3D part to be produced from *RPT*.

Total removal volume (*TRV*) = the Boolean difference of the *RPT* and *FPT*, which may consist of one or more disconnected bodies or lumps.

Delta removal volume (*V*) = the disconnected bodies or lumps which, produced by Boolean difference of the *RPT* and *FPT* machining, feature a continuous volume that can be removed by a single machining operation in a single set-up.

Non-machining face (NM-face) = the face on the *TRV* inherited from *FPT*.

Machined face (M-face) = the face on the *FPT* at which material is removed by machining processes from the *RPT*.

Color attribute face (C-face) = the face on the *DSV* produced by partitioning.

Simple volume (*SV*) = a volume produces an empty set partitioned by its M-faces.

Delta simple volume (*DSV*) = the partitioned delta simple volume.

Feasible simple volume (*FSV*) = the glued maximal simple volume.

Introduction

Mechanical parts are typically manufactured using multiple manufacturing processes that include primary and secondary processes. Primary processes, such as casting and forging, realize the primary shape of the part, while secondary processes such as machining generate more detailed shape of the part. The machining operations are

carried out on the component where critical functional requirements like fits and assembly arise, and hence the material removal volume generally falls below 10 per cent of the total stock volume. This is a common phenomenon in automobile- and machine tool-manufacturing industries. This paper focuses on such types of part models and provides a unique approach to extracting their complex features.

Unlike most of the previous research, this approach is based on the general techniques for dealing with intersecting features. The features handled in this paper are restricted to machining features only, which can be considered as the portion of a part having some manufacturing significance and can be created by common machining operations, such as drilling, boring, reaming, milling, shaping, planing, broaching, etc. The initial input to the system is *FPT*, *RPT* and the reference position of *FPT* with respect to *RPT* in the workspace. The *TRV* is obtained by conducting a Boolean decomposition process. Faces on the *TRV* are segregated into M-faces and NM-faces, and the interacting removal volumes are partitioned into delta simple volumes with their M-faces. A generate-and-test strategy is then used to glue the delta simple volumes into accessible and feasible simple volume features. Computational geometry techniques are used to produce the sets of feasible simple volume features. Because some sets of machining features may not be accessible, the features' accessibility and their interactions with others are analyzed and tested in a verification phase. The validity tests ensure that each set of the proposed features is accessible, does not intrude into the desired part, and satisfies other machinability conditions. The process continues until it produces a complete decomposition of the



Integrated Manufacturing Systems
14/7 [2003] 610-623

© MCB UP Limited
[ISSN 0957-6061]
[DOI 10.1108/09576060310491964]

The Emerald Research Register for this journal is available at
<http://www.emeraldinsight.com/researchregister>



The current issue and full text archive of this journal is available at
<http://www.emeraldinsight.com/0957-6061.htm>

volumes into sets of the feasible simple volume features.

The system has been implemented using Parasolid libraries (Unigraphics Solutions Inc., 1999) and in the Visual C++ environment, and on the basis of the work done in 2001. Based on the tests, the developed algorithm is able to recognize most of the common machining operations performed on the casting and forging components.

Related works

There is much literature in the area of feature extraction. The following review focuses on those closely related to the machining feature extraction based on the volume decomposition problem dealt with in this research.

Woo (1982) developed a decreasing convex hull algorithm, which decomposes the workpiece as a series of either additive or subtractive solids. Kim (1992) used an alternative sum of volumes with partitioning (ASVP) derived from some edges to solve the non-convergence problem of convex decomposition. Currently the method can be applied to the polyhedrons because of the complexity of convex hull composition for curved objects. But the conversion from decomposed convex components to machining features is not guaranteed. Wang and Chang (1990) developed a backward-growing feature recognizer capable of decomposing intersecting features into elementary machined shapes. Tseng and Joshi (1994) combined volume decomposition and volume reconstruction to provide multiple interpretations for interacting machining features based on predefined features. Karinithi and Nau (1992) proposed a feature extraction approach based on algebra of features to solve feature interaction problem. Dong and Vijayan (1997) developed a BS-CE approach to minimize the total number of set-ups, and to remove the machining volume as much as possible in a set-up. However, it is hard to get the LBS and NCE (NP), and also some un-machined features may be produced. Shen and Shah (1994, 1998) proposed a process-oriented classification of machining features by using a half-space partitioning method, which recognizes features that can be produced by common machining operations. Bidarra *et al.* (1998) introduced a cellular representation for feature models that can manage the data effectively to overcome the limitation in the feature-modeling system. Shirur *et al.* (1998) also used the process-oriented classification

for mapping machining volumes to machining operations by encoding geometric shapes into algebraic expression. The process-oriented approach has an advantage over the feature-based approach in that an undefined feature can still be handled as long as it can be produced by a process.

Most of the above approaches match the decomposed volume features into "standard" machining features defined by using a feature-based or hint-based method, which nevertheless will cause problems in robustness. First, the number of representative templates may not be enough in number. Second, the processes of characterizing the templates are almost all heuristic and may not be valid.

The approach introduced in this paper employs the process-oriented approach. It first obtains the machining removal volumes by conducting a Boolean difference operation between the *RPT* and the *FPT*. It is noted that the procedure up to this stage is similar to the volume decomposition approach (Tseng and Joshi, 1994; Shen and Shah, 1998; Shirur *et al.*, 1998; Sakurai and Chin, 1994). But from the removal volumes the reported approaches basically map the volumes to processes, and the raw stock used is always a rectangular block. When such an approach is applied to casting and forging stock, problems will arise. For example, the shape of the removal volumes is generally not regular but rather complex in nature. Hence, it is not always possible to map the volumes into processes. To overcome this problem, in this approach, the M-faces are identified from the removal volumes and grouped into clusters, which can be produced by a single process type. The objects handled are faces, not volumes. In such a way, casting and forging components can be handled effectively. The partitioning procedure seems to be the same with Tseng and Joshi (1994); Dong and Vijayan (1997); Shen and Shah (1998) and Bidarra *et al.* (1998), but Tseng and Joshi (1994) partitioning with extended faces, Dong and Vijayan (1997) with the maximal face and Shen and Shah (1998) and Bidarra *et al.* (1998) with half-space. Those are computationally intensive. Here, M-faces are used. Each disconnected body obtained by Boolean difference operations between the *RPT* and the *FPT* of casting and forging parts is always local, and the number of M-faces on it is usually small. Therefore the computational complexity is not so intensive and alternative sets of feasible machining features can be produced.

Basic concepts

Machining face

A machining face is on the *TRP* at which material is removed from the *RPT*. More formally, a machining face is the contribution of the delta removal volume's boundary *V* to the final part's boundary *FPT*, and is defined as:

$$M - faces = \partial\Delta V \cap \partial FPT \quad (1)$$

where \cap denotes non-regularized set intersection, and ∂ is the topological boundary operator. The boundary of the delta removable volume as a source of data for volume searching is more convenient than the boundary of the final part. Because it is easier to show that *V* contains only raw part faces to be removed and final part faces to be generated, where *FPT* may contain faces that are not machined, which therefore are of no interest.

Machining face vectors

These vectors are used to sign the M-faces on the delta removal volumes and the simple machining volumes. For example, if a delta removal volume has four M-faces. it can be signed using a vector (1, 1, 1, 1). It can also be decomposed into four vectors (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0) and (0, 0, 0, 1) to represent the M-face₁, M-face₂, M-face₃, M-face₄ respectively. The footnote 1,2,3,4 represents the order of the M-faces found. The delta simple machining volumes produced by partitioning this delta removal volume can be signed in the same way. For example, if one delta simple machining volume has two M-faces that match the first and the third M-face on the delta removal volume, they can be represented as a vector (1, 0, 1, 0), or be represented as a vector (1, 0, 0, 0) and a vector (0, 0, 1, 0). These M-faces vectors will facilitate gluing the delta simple machining volumes into the feasible simple machining volumes just by classifying them with the same M-face vectors.

Machining volume

Simple machining volume

A simple machining volume is a volume that can be removed by a single machining operation. It may consist of more than one M-face but it produces an empty set by partitioning it using its M-faces. There are two types of simple machining volumes from casting and forging parts, namely, type1 simple machining volumes and type2 simple machining volumes.

Type1 simple machining volumes are to generate features on the part originally. Some examples are shown in Figure 1. They

represent basic form features. The typical type1 simple machining volumes on the casting and forging parts are small holes, but there may be other simple machining volumes too, such as pocket, slot, step, etc.

Type2 simple machining volumes come from cast then machined features. Some examples are shown in Figure 2. They are volumes obtained from the features first produced by casting or forging manufacturing processes and then to be machined in order to obtain higher accuracy. So the removal volumes for these are very small.

Type2 simple machining volumes also represent the features that solid simple machining volumes represent. For example, in Figure 2, (a) presents a pocket, (b) is a hole, (d) is a step, (e) is a slot and (f) is a notch. But they are the boundary of type1 simple machining volumes with a certain small thickness. The difference between type2 simple machining volumes and type1 simple machining volumes lies in process mapping. For example, a hole presented by type1 simple machining volume in Figure 1 must drill first, then bore or ream, but the first operation to machine a hole presented by type2 simple machining volume in Figure 2(b) is boring or reaming not drilling.

Two kinds of intermediate simple machining volumes are produced in this system. They are delta simple machining volume and feasible simple machining volume, which are discussed below.

Delta simple machining volume (DSV)

Delta simple machining volumes of a part are obtained from partitioning the *Vs* of the part with M-faces on it. There is always at least one M-face or C-face on a delta simple volume. Each delta simple volume belongs to a certain *V*. It is the basic element for the gluing algorithm in the feature extraction system.

Feasible simple machining volume (FSV)

A feasible simple machining volume of a *V* on a part is obtained from gluing the delta simple volumes according to the M-faces or C-faces on the *V*. An *FSV* set is a candidate simple machining volume set for a *V* and satisfies the following conditions:

- Machining all the *FSVs* in the set can create the *V*.
- If any one of the *FSVs* in the set is not machined, the complete part cannot be created.
- Not all the *FSVs* in the set intersect with one another.

Intersecting machining volume

On the other hand, an intersecting machining volume is a volume that produces

Figure 1
 Type1 simple machining volumes

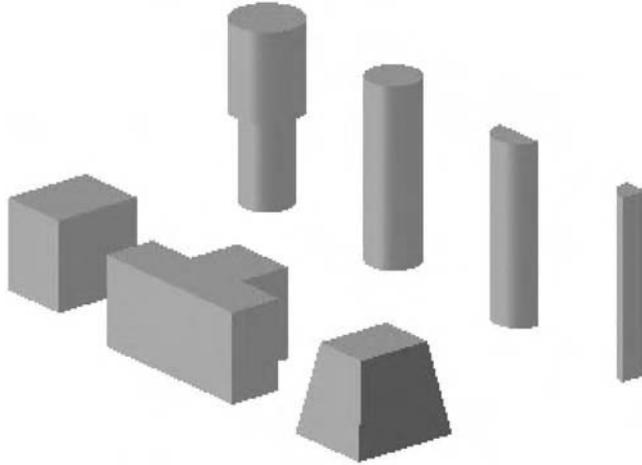
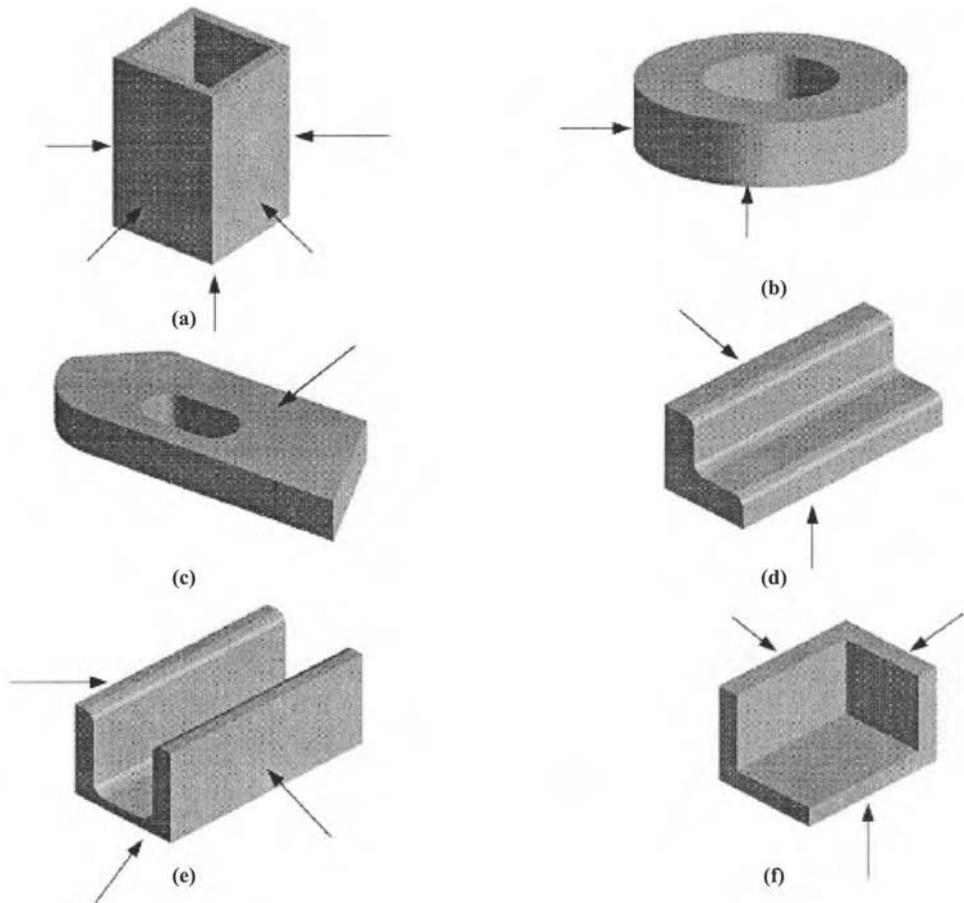


Figure 2
 Type2 simple machining volumes

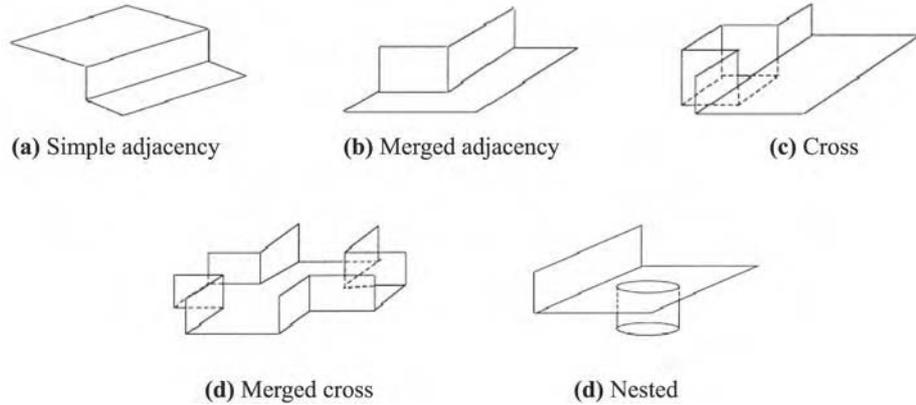


non-empty subsets of the initial volume by partitioning it using its M-faces. Typical intersecting machining volume types are shown in Figure 3.

The number of intersecting machining volume formations is lower, due to the nature

of the input (near net shape) and this is one of the main advantages of this approach. This can be very well visualized from the disconnected volumes of TRV. Some intersecting machining volumes may still remain and five possible ways of forming

Figure 3
Five types of interesting machining volumes



intersecting machining volumes are used, i.e. simple adjacency, merged adjacency, cross, merged cross, and nested, as shown in Figure 3. It is worth mentioning that by no means is this intersecting volume classification complete, since it is a very complicated issue that needs further research.

In the case of simple adjacency, the intersecting M-faces of the two steps meet at a convex edge, and their common edges do not form a closed edge loop in any single face of a volume. Merged adjacency is a special case of simple adjacency, in which one or more M-faces of each of the two volumes are merged together to form a single common M-face, while others remain the same as in the simple adjacency. In the cross-relationship, one or more M-faces in a volume are split into different parts in the solid model by another volume. The merged cross-relationship is a special case of the cross-relationship but it requires one or more intersecting M-faces of each of the two volumes to merge together into a single common M-face. When a volume is nested in another volume, the original configurations of both volumes are kept and their common convex edges form a closed edge loop in an M-face of the nested volume.

An intersecting machining volume may consist of several simple machining volumes. By the partitioning approach introduced in the next section, they can be decomposed into a set of simple machining volumes in the end.

Machining volume validation

Machining volumes can be considered as the portion of a part having some manufacturing significances and can be created by common machining operations, such as drilling, boring, reaming, milling, shaping, planing, broaching, etc. These volumes can be removed in a single machining operation in a single set-up. However, a single machining

operation may consist of several steps by one or more cutters. For example, a hole may require drilling and boring and a pocket may require several cutter steps to clear the entire volume. Each machining volume typically corresponds to two or more machining faces. The three necessary attributes that must be held in a simple machining volume are:

- 1 It is contained in the *TRV*.
- 2 It can be removed from the *RPT* in a single machining operation with a three-axis machining center.
- 3 Its removal creates a portion of the part surface without destroying the part.

Feasible partition verifier

In this system, the machining volume validation rules are guaranteed naturally according to the volume decomposition approach. The volumes obtained in the partitioning procedure are simple machining volumes. But, in the gluing algorithm, some volumes that are not simple machining volumes may be produced. So at this stage, the simple machining volume verifier should be used to ensure every output *FSV* set contains only simple machining volumes. The M-faces convexity relationships are checked to make sure if a volume is a simple machining volume.

To check the convexity of M-faces on a volume is to check the formed edges' convexities. The convexity definition is just like the common meaning defined in the previous literature. As illustrated in Figure 4, M-face₂ and M-face₁ have the convex relationship, while M-face₂ and M-face₃ have the concave relationship. If there is one concave relationship between them, this volume is not a simple machining volume, and is thrown out. The algorithm to judge the convexity between two M-faces is shown in

Figure 4
 Convexity between two M-faces

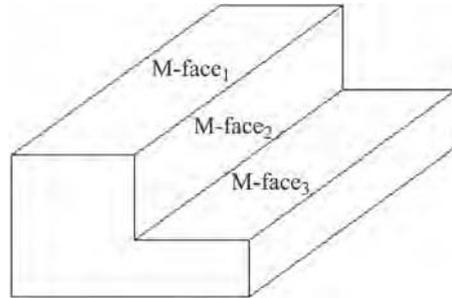
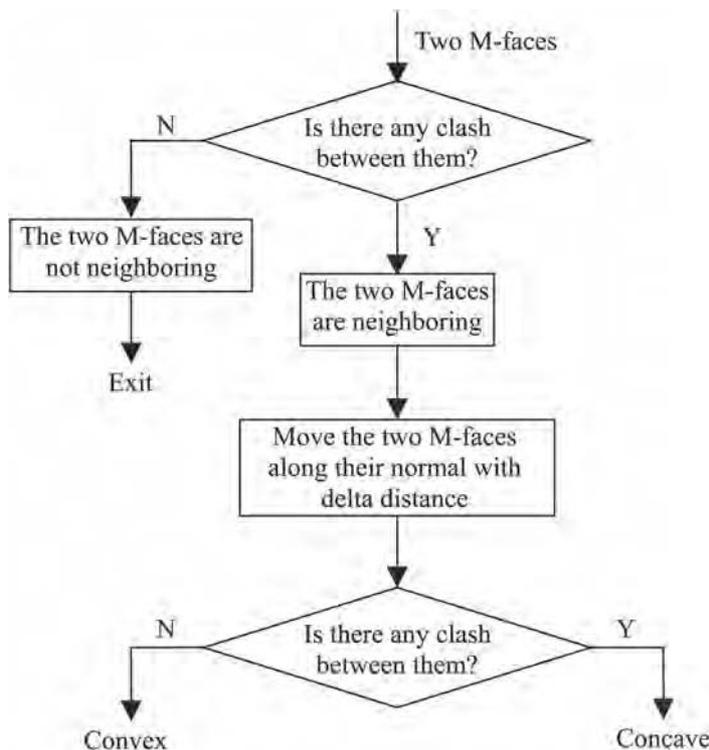


Figure 5. A feasible simple machining volume must satisfy the following two rules:
 1 there must be at least one M-face on it; and
 2 if there are more than two M-faces, all the neighboring M-faces must have the convex relationships.

The proposed feature extraction approach

The inputs to the proposed feature extraction system are the *FPT* and *RPT*, and the reference position and orientation in which the two models overlap. The feature extraction process proceeds with the following four steps, as taken in Figure 6:

Figure 5
 Algorithm to judge the convexity



- 1 generating *TRV* and sets of disconnected body *V*;
- 2 identifying M-faces on each *V*;
- 3 partitioning each *V* into a set of *DSV* with its M-faces, and getting the M-faces vectors of *V* and *DSV*;
- 4 gluing each set of *DSV* into sets of *FSV* based on their M-faces vectors.

The example used to illustrate the system implementation is a link. The *FPT* and *RPT* are shown in Figure 7.

Generating TRP

FPT and *RPT* are the inputs to the system after the machining environment is selected. *FPT* is then translated to the reference position and the orientation of the *RPT*. This is followed by the Boolean difference of *FPT* and *RPT*, resulting in the *TRV*, which may be one body or a set of disconnected bodies. The *TRV* can be expressed as:

$$TRV = \{\Delta V_1; \Delta V_2; \Delta V_3; \dots \Delta V_i; \dots \Delta V_n\} \quad (2)$$

where *n* is the number of disconnected bodies and *V* is certain delta removal volume.

For the link, *FPT* and *RPT* have the same center, so they overlap each other. Their position and orientation are shown in Figure 8. After Boolean difference, one delta volume is produced, as shown in Figure 9. Its *TRV* can then be expressed according to Equation (2), where *n* is equal to one, by:

$$TRV = \{\Delta V_1\} \quad (3)$$

Identifying M-faces

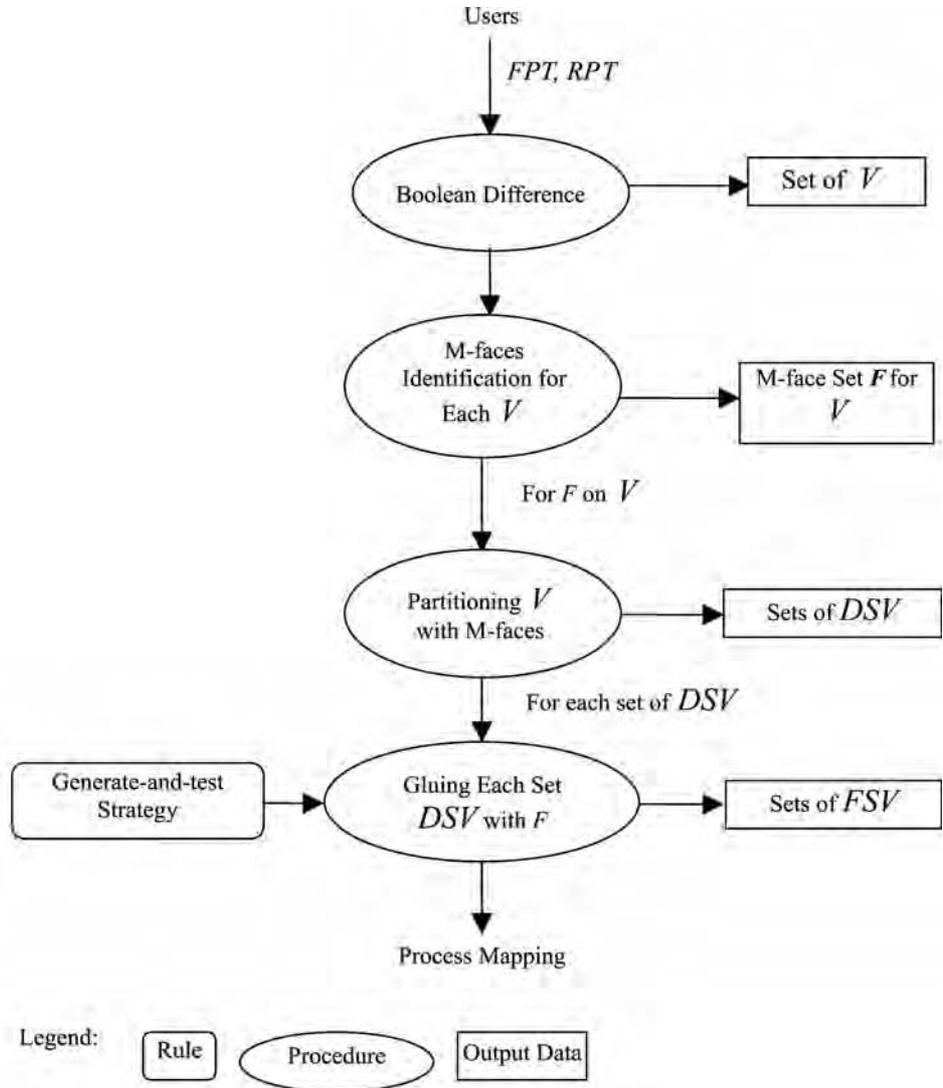
The faces of *TRV* are primarily categorized into two types, namely M-faces, which come from the *FPT*, and NM-faces which come from the *RPT*. In the *TRV*, each *V* possesses at least one M-face. By performing topological intersections between faces of *V* and faces of *FPT*, the M-faces can be obtained. This operation can be expressed as:

$$\Delta \bar{V} = \sum_{i=1}^m M - face_i + \sum_{i=1}^n NM - face_i \quad (4)$$

The segregated M-faces are of more importance than the NM-faces because these faces correspond to surfaces obtained during machining operations. The next step is to partition each *V* into a set of *DSV* using these M-faces. A simple machining volume is a volume that produces an empty set by partitioning the volume using its M-faces, and the M-faces are called simple M-faces. A single machining volume may consist of more than one M-face.

On the other hand, an intersecting machining volume can produce non-empty subsets of the initial volume by partitioning

Figure 6
 Feature extraction process



the volume using its M-faces, and the M-faces are called intersecting M-faces. This partitioning approach handles these M-faces in a general way.

The M-faces on the delta volume of the link in Figure 9 are shown in Figure 10. It is an intersecting machining volume with intersecting M-faces.

Partitioning V with M-faces

In this section, each delta volume in *TRV* obtained by Boolean difference is partitioned into several delta simple volumes by the M-faces on it. The M-faces on a *V* are first obtained and put in *F* set:

$$F = \{MF_1; MF_2; MF_3; \dots; MF_i; \dots MF_t\} \quad (5)$$

where *MF* is the M-face found in a *V*, *t* is the number of M-faces found on the *V*. The footnote shows the order in which the M-faces are found.

Then each *V* is partitioned by these M-faces one by one according to the order in which they are found. This procedure decomposes each *V* into delta simple volumes. Each *DSV* contains at least one M-face. Below is the partitioning algorithm:

Input: *V* list

Output: *DSV* list

Two stacks are defined, one being the input stack, and the other the resulting stack. The resulting stack is empty at first. Take one element *V* as input in the input stack at a time, and record its M-faces. Then partition it with its M-faces one by one. For each partitioning, the partitioned bodies are stored in the resulting stack. After finishing, the input stack is filled with the bodies in the resulting stack and then, as they are partitioned by another M-face, the resulting

Figure 7
 Input models *FPT* and *RPT*

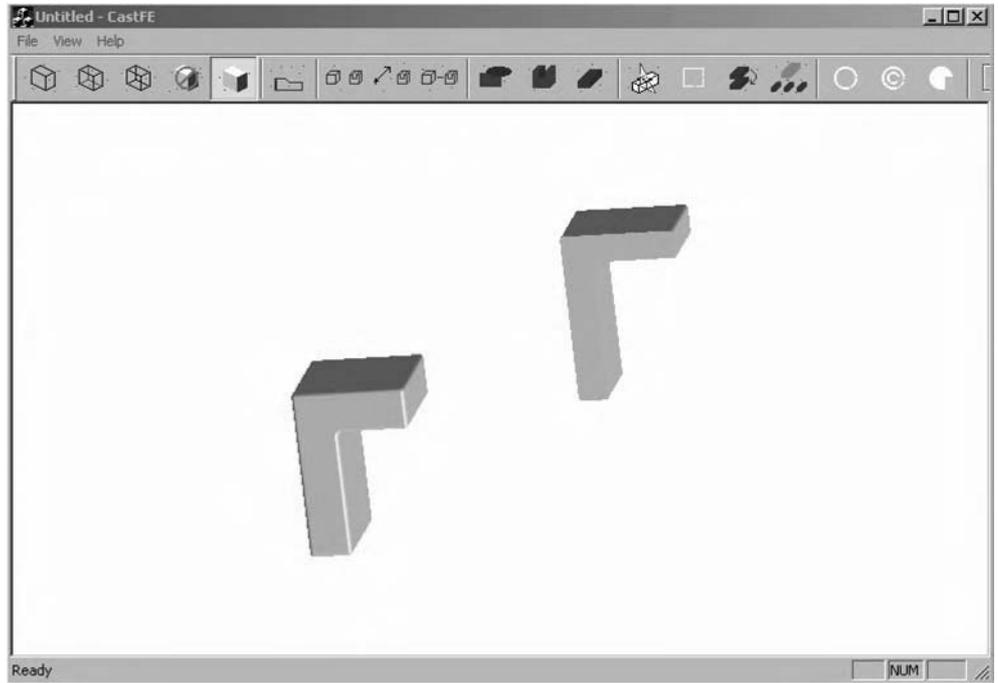


Figure 8
 Position and orientation of *FPT* and *RPT*

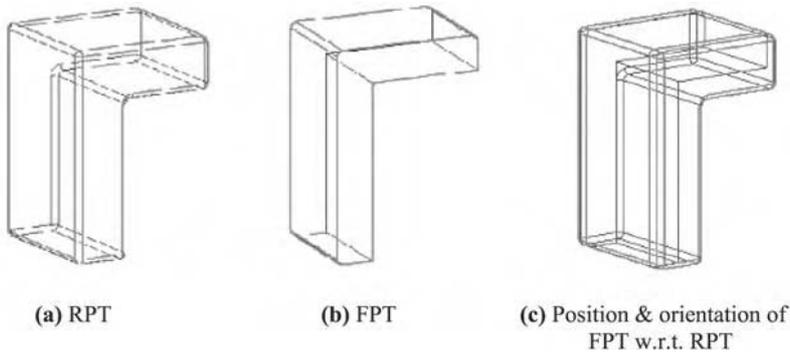


Figure 9
 Delta volume after Boolean difference

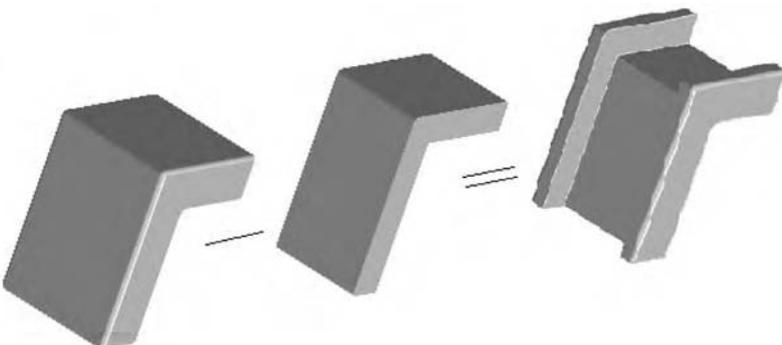
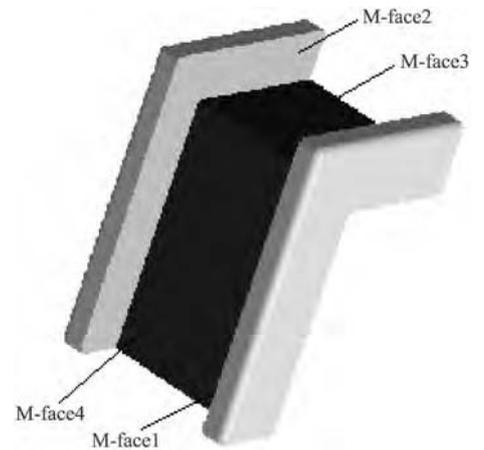


Figure 10
 M-faces on the delta volume



stack is taken by the newly produced bodies one by one. This process goes on until this V is fully partitioned by its M-faces.

The above partitioning procedures continue till all V are partitioned. Then it can obtain the following sets of DSV :

$$\Delta V_1 = DSV_{11} + DSV_{12} + DSV_{13} + \dots + DSV_{1m}$$

$$\Delta V_2 = DSV_{21} + DSV_{22} + DSV_{23} + \dots + DSV_{2m}$$

...

$$\Delta V_n = DSV_{n1} + DSV_{n2} + DSV_{n3} + \dots + DSV_{nm} \quad (6)$$

where n is the number of delta removal volumes, and m is the total number of delta simple machining volumes obtained by the partitioning of each V . Combining Equations (2) and (6), obtains:

$$TRV = \sum_{i=1}^n \Delta V_i = \sum_{i=1}^n \sum_{j=1}^m DSV_{ij} \quad (7)$$

where m and n have the same meaning as in Equation (6).

This is an exhaustive partitioning. Each set of DSV will be reconstructed into sets of FSV next.

For the link, the M-faces are shown in Equation (8). Figure 11 shows the set of DSV obtained by partitioning the intersecting volume with the order of these M-faces shown in Figure 10:

$$F = \{MF_1 + MF_2 + MF_3 + MF_4\} \quad (8)$$

and, according to Equation (7), Equation (3) can be expressed at this stage as the following:

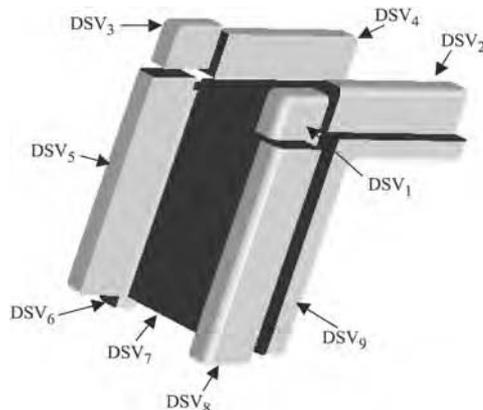
$$TRV = \Delta V_1 = \sum_{j=1}^9 DSV_{1j}. \quad (9)$$

Gluing DSV by M-faces

This stage reconnects and glues the delta simple volumes DSV produced in the partitioning process above. The aim is to get the feasible simple machining volumes sets. Each set of FSV may be machined by a different set of operations and in a different machining sequence. Some sets may result in better manufacturing practice from the engineering point of view and some may not be lucrative in terms of economic aspects, which are more significant during the process selection.

Figure 11

Delta simple volumes after partitioning



Gluing procedure

Figure 12 shows the gluing algorithm. There are several stages in the procedure:

- Find M-faces on V , and get their surfaces and orientations.
- Find M-faces on each partitioned DSV , and get and compare their surfaces and orientations with the ones on the V obtained from step 1 respectively.
- If more than one M-face is found on this DSV , then judge if there are more than one M-face or C-face found in the previous DSV . If true, then copy the previous DSV entities which are classified, and put this entity to the corresponding groups separately according to M-faces on it. If false, there is no need to copy the previous DSV entities classified; just put this entity to the corresponding groups separately according to the M-faces on it. If no M-faces are found, go to step 4. Otherwise, change to the next DSV , and go to step 2.
- Find C-faces on this DSV , which are produced by partitioning with M-faces, so each has the same geometrical and topological information as certain M-faces, get their surfaces and orientations, and compare their surfaces and orientations with the M-faces on the V respectively.
- If more than one C-face is found on this DSV , then judge if there are more than one M-face or C-face found in the previous DSV . If true, then copy the previous DSV entities which are classified, and then put this entity to the corresponding groups separately according to C-faces on it. If false, there is no need to copy the previous DSV entities classified; just put this entity to the corresponding groups separately according to C-faces on it. If no C-faces are found, report error message. Otherwise, change to the next delta simple machining volume DSV , and go to step 2.
- Unite the entities in each M-face group G and get the FSV sets for the V , then go to the next V .

This algorithm glues sets of DSV according to the M-faces vectors.

The above algorithm can be illustrated with the intersecting machining volume of the link. There are four M-faces on the V . So the M-face vector for it is (1, 1, 1, 1) and, after partitioning, it has nine $DSVs$. The M-face vectors on each DSV are shown in Table I.

According to the gluing algorithm, there should be 18 possible FSV sets. The M-faces

Figure 12
 Gluing algorithm

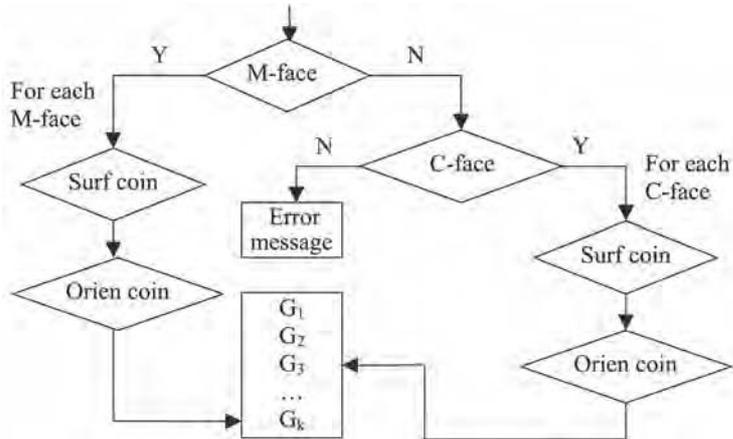


Table I
 M-faces on each DSV

	M-face ₁	M-face ₂	M-face ₃	M-face ₄	C-face ₁	C-face ₂	C-face ₃	C-face ₄
DSV ₁	1	0	0	0	0	0	0	0
DSV ₂	1	0	0	0	0	0	0	0
DSV ₃	0	1	0	0	0	0	0	0
DSV ₄	0	1	0	0	0	0	0	0
DSV ₅	0	1	0	0	0	0	0	0
DSV ₆	0	0	0	0	0	1	1	1
DSV ₇	0	0	1	1	0	0	0	0
DSV ₈	1	0	0	0	0	0	0	0
DSV ₉	0	0	0	0	1	0	1	1

on DSV₆, DSV₇ and DSV₉ have determined the set number. A tree representing the forming of the possible FSV sets is presented in Figure 13. The 18 possible sets are shown in Table II.

So according to the gluing algorithm, more than one set of simple machining volumes for each intersecting V can be produced. The number is determined by the number of M-faces or C-faces on the DSVs and V.

Testing

The procedures described above are “process-oriented” rather than “feature-oriented”, which makes them independent of feature types or intersection types to be recognized. It evaluates each candidate volume set for its machining characteristics, rather than predefining a closed set of features and hints for recognizing them. In general, many

Figure 13
 Tree representing the possible FSV sets

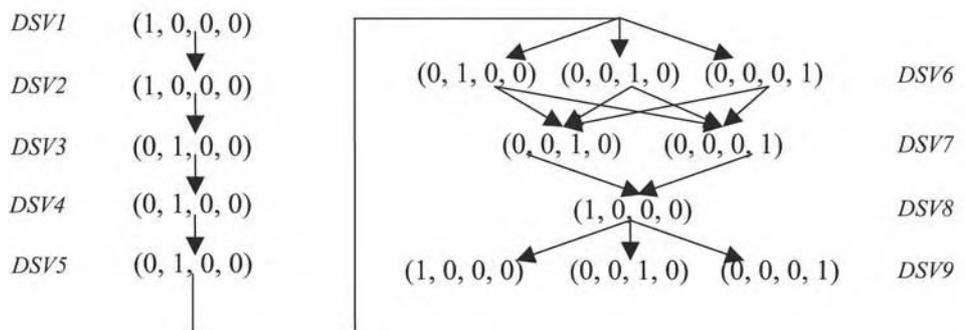


Table II
 Possible FSV sets for *V*

	M-face 1	M-face 2	M-face 3	M-face 4
G1	<i>DSV</i> ₁ , <i>DSV</i> ₂ , <i>DSV</i> ₈	<i>DSV</i> ₃ , <i>DSV</i> ₄ , <i>DSV</i> ₅ , <i>DSV</i> ₆	<i>DSV</i> ₇	<i>DSV</i> ₉
G2	<i>DSV</i> ₁ , <i>DSV</i> ₂ , <i>DSV</i> ₈	<i>DSV</i> ₃ , <i>DSV</i> ₄ , <i>DSV</i> ₅	<i>DSV</i> ₇ , <i>DSV</i> ₆	<i>DSV</i> ₉
G3	<i>DSV</i> ₁ , <i>DSV</i> ₂ , <i>DSV</i> ₈	<i>DSV</i> ₃ , <i>DSV</i> ₄ , <i>DSV</i> ₅	<i>DSV</i> ₇	<i>DSV</i> ₆ , <i>DSV</i> ₉
...
G18	<i>DSV</i> ₁ , <i>DSV</i> ₂ , <i>DSV</i> ₈ , <i>DSV</i> ₉	<i>DSV</i> ₃ , <i>DSV</i> ₄ , <i>DSV</i> ₅	<i>DSV</i> ₇	<i>DSV</i> ₆

combinations exist and the volume gluing can, if desired, find all valid combinations that satisfy the gluing rules. But some sets may contain volumes that are not simple machining volumes, so each glued volume is tested during the gluing procedure by the simple machining volume validation rules and feasible partition rules described in the previous section. Only a set in which all the volumes are simple machining volumes is a candidate to be machined, that is, it is a feasible simple machining volume set. With this procedure, all the *FSV* sets are obtained.

Till now, the *V* and *TRV* can be expressed as follows:

$$\Delta V = \bigcup_{k=1}^l \sum_{q=1}^p FSV_{kq} \quad (10)$$

$$TRV = \sum_{i=1}^n \Delta V_i = \sum_{i=1}^n \bigcup_{k=1}^l \sum_{q=1}^p FSV_{ikq} \quad (11)$$

where *l* is the number of *FSV* sets of *V*, *p* is the number of simple machining volumes in an *FSV* set, and *n* is the number of *V*.

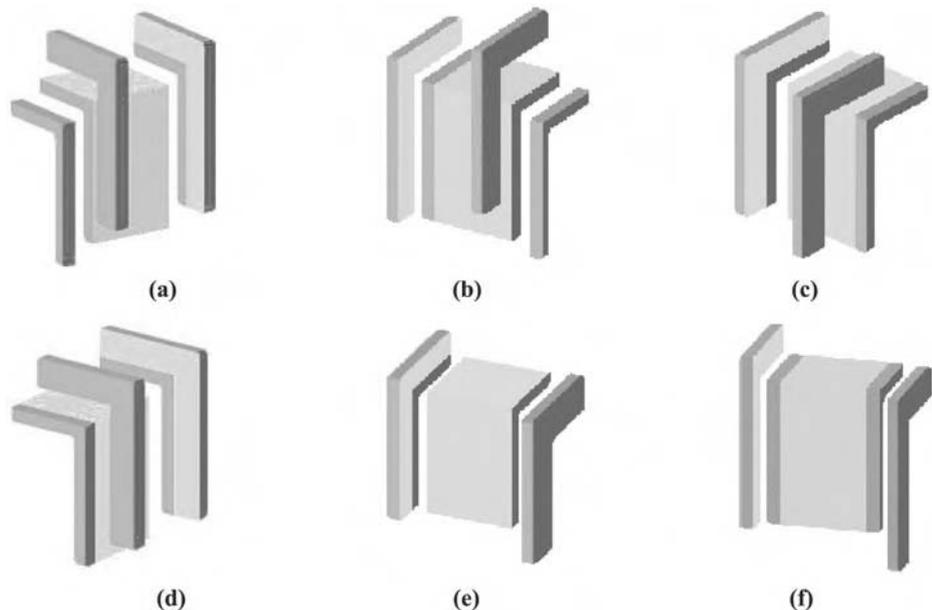
Based on the test, all the 18 sets are feasible sets for the intersecting delta volume in Figure 9. Among them, ten sets have four feasible simple machining volumes each, and eight sets have three feasible simple machining volumes each. Six distinct sets are shown in Figure 14. Using Equation (11) to express the *TRV* of the example, shows:

$$TRV = \bigcup_{k=1}^{10} \sum_{q=1}^4 FSV_{kq} + \bigcup_{k=1}^8 \sum_{q=1}^3 FSV_{kq} \quad (12)$$

Case studies

The current implementation of the system uses Parasolid Library (Unigraphics Solutions Inc., 1999) as the geometry kernels, runs on Windows XP platform and in the Visual C++ environment. Parts are created on Unigraphics 17.0 platform, and transformed to *.xmt_txt* files, which can be recognized by Parasolid. The system has been tested with a large variety of casting parts. The above case is one example tested by the programs developed. A case is used as

Figure 14
 Sets of *FSV*



one of the case study parts, which will be discussed below.

The casting material of the case is HT300. The *FPT* and the *RPT* as the inputs are shown in Figure 15. There are seven simple machining volumes in this case: five holes, two slots, which are the features that the system should extract. The delta volumes after Boolean difference are presented in Figure 16.

There are six delta volumes after Boolean difference. Among them, five holes are simple machining volumes and the other one with two slots cross-merged is an intersecting machining volume, which has to process further to get two separate slots. This classification is shown in Figure 17. There

are nine M-faces on the intersecting volume. After partitioning with these M-faces, it is split into five delta simple volumes, which are shown in Figure 18. After gluing with the nine M-faces, 81 alternative sets of *FSV* are produced. One of them is shown in Figure 19.

The gluing information of these intersecting delta volumes is shown in Figure 19. An *FSV* set for each is also shown.

Conclusions

This paper reports an approach to extract machining features for casting and forging parts. The approach is process-oriented and feature-independent. Each disconnected body

Figure 15
Input models

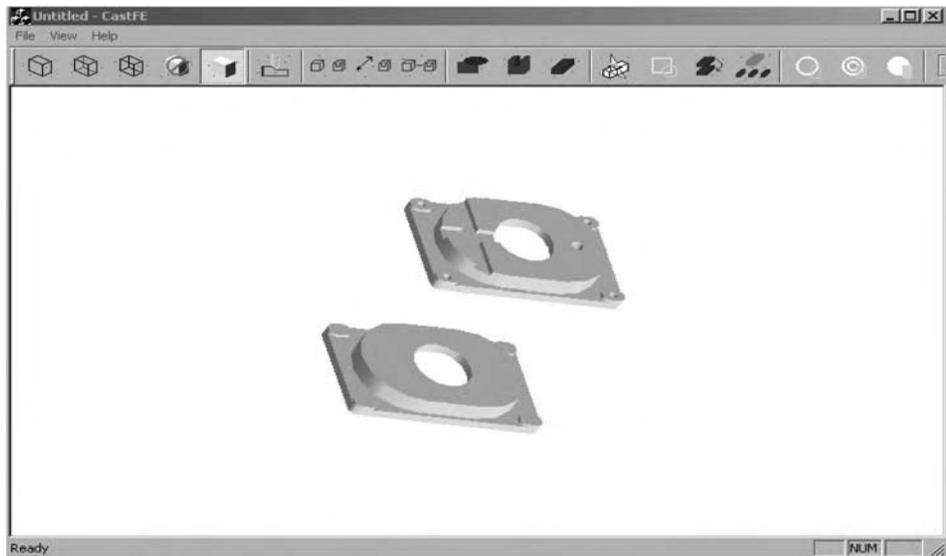


Figure 16
Delta volumes after Boolean difference

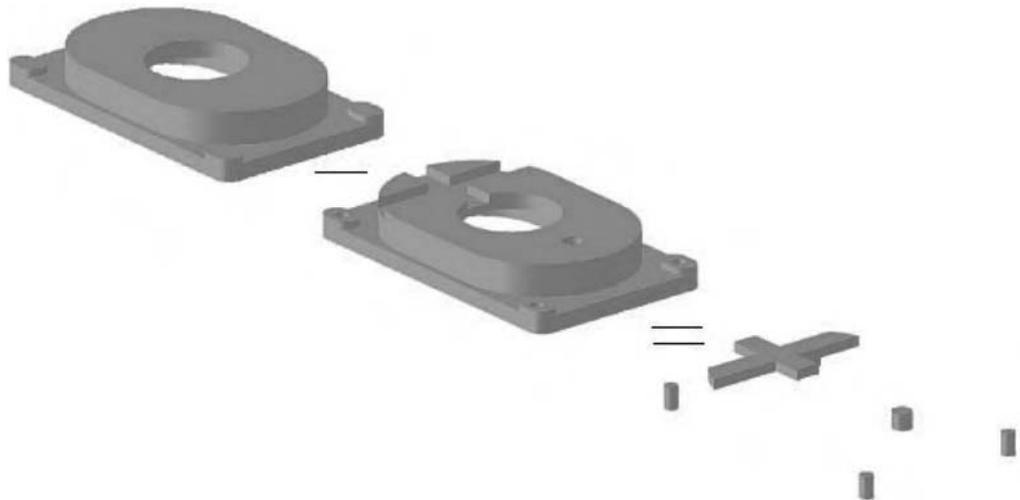


Figure 17
 Delta volumes classification

Intersecting machining volume	Simple machining volume
	

Figure 18
 DSV sets obtained by partitioning

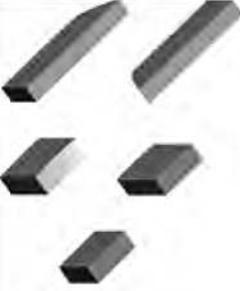
Intersecting machining volume	<i>DSV</i> after partitioning
	

Figure 19
 FSV sets for each intersecting machining volume

Intersecting machining volume	The number of <i>FSV</i> sets	an <i>FSV</i> set
	56	

obtained by Boolean difference operation between the *RPT* and the *FPT* of casting and forging parts is always local, and the number of M-faces on it is usually small. Therefore the computational complexity is not so intensive. Since feature mapping and the gluing procure are based on the machined faces, the glued volumes can be directly mapped into the machining process and the shape of the removed volume does not play any significant role. These characteristics make the system suitable for handling the complex casting and forging parts. The case and the example also show that the system is able to handle not only simple machining features, but also the complex interacting machining features. The feature extracted by the system can be used in other applications in addition to process planning, e.g. NC programming, fixture planning, etc.

References

- Bidarra, R., de Kraker, K. and Broonsvoort, W. (1998), "Representation and management of feature information in a cellular model", *Computer-Aided Design*, Vol. 30 No. 4, pp. 301-13.
- Dong, J. and Vijayan, S. (1997), "Manufacturing feature determination and extraction. Part II: a heuristic approach", *Computer-Aided Design*, Vol. 29 No. 7, pp. 475-84.
- Karinthi, R. and Nau, D. (1992), "An algebraic approach to feature interactions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14 No. 4, pp. 469-84.
- Kim, Y.S. (1992), "Recognition of form features using convex decomposition", *Computer-Aided Design*, Vol. 24 No. 9, pp. 461-76.
- Sakurai, H. and Chin, C. (1994), "Definition and recognition of volume features for process planning", in Shah, J.J., Mantyla, M. and Nau, D. (Eds), *Advances in Feature-Based Manufacturing*, Elsevier, Amsterdam, pp. 65-80.
- Shen, Y. and Shah, J.J. (1994), "Feature recognition by volume decomposition using half-space partitioning", *Advances in Design Automation*, DE-Vol. 69-1, Vol. 1, ASME, New York, NY.
- Shen, Y. and Shah, J.J. (1998), "Recognition of machining features based on HSPCE decomposition, feature composition, and process-oriented classification", *Transactions of the ASME, Journal of Mechanical Design*, Vol. 120, pp. 668-77.
- Shirur, A., Shah, J.J. and Hirode, K. (1998), "Machining algebra for mapping volumes to machining volumes for developing extensible generative CAPP", *Journal of Manufacturing Systems*, Vol. 17 No. 3, pp. 167-82.
- Tseng, Y.J. and Joshi, S. (1994), "Recognizing multiple interpretations of intersecting machining features", *Computer-Aided Design*, Vol. 26 No. 9, pp. 667-88.
- Unigraphics Solutions Inc. (1999), *Parasolid Solid Modeler, Version 10.1, Programming Reference*, Maryland Heights, MO.
- Wang, M.-T. and Chang, T.-C. (1990), "Feature recognition for automated process planning", *Proc. Manuf. Int. 90, Part-2: Adv. Manuf. Syst.*, ASME, New York, NY, pp. 49-54.
- Woo, T.C. (1982), "Feature extraction by volume decomposition", *Proceedings of the Conference on CAD/CAM*, MIT Press, Cambridge, MA, pp. 39-45.

Further reading

- Kailash, S.B., Zhang, Y.F. and Fuh, J.Y.H. (2001), "A volume decomposition approach to machining feature extraction of casting and forging components", *Computer-Aided Design*, Vol. 33 No. 8, pp. 605-17.

Awards for Excellence

David Bennett

Aston Business School, Aston University, Birmingham, UK

is the recipient of a Leading Editor Award for the journal *Integrated Manufacturing Systems*.



*David Bennett, Editor of **Integrated Manufacturing Systems** (Journal of Manufacturing Technology Management from 2004), receiving his Leading Editor Award from John Peters, Director of Research and Author Relations*

David Bennett is a Chartered Engineer, holds both MSc and PhD degrees from the University of Birmingham, UK and is currently Professor of Technology Management and Deputy Head for External Affairs at the Aston University Business School. His teaching is in operations and technology management and his research is concerned with issues relating to production systems design, quality and reliability management, technology management and the transfer of technology between industrialised and developing countries, especially in the Asian region. He is also an Adjunct Professor with the University of South Australia (International Graduate School of Management) involved in supervision of research students in Singapore and Hong Kong. He has authored and co-authored several books as well as numerous articles. He has also undertaken consulting assignments for the European Commission (EC) and the United Nations Industrial Development Organization (UNIDO). He is a member of the Board of the European Operations Management Association (EurOMA) and is also on the Executive Council for the International Association for Management of Technology (IAMOT).