Productivity Assessment from Advanced Integration of Canned Cycles to CAD/CAM Software

JEAN-FRANÇOIS CHATELAIN[1], YAN BOUTIN[2], FRANÇOISE MARCHAND[1], ANTOINE TAHAN[1]

Mechanical Engineering Department
[1]École de technologie supérieure, Université du Québec
1100 Notre-Dame Street West, Montréal, Québec, H3C 1K3
CANADA

Mechanical Engineering Department
[2]École Polytechnique de Montréal
yboutin@mec.etsmtl.ca http://www.polymtl.ca

Abstract: CAD/CAM systems and CNC machines are now commonly utilized in the manufacturing industry. The integration level between these two technologies is one key element which directly impacts the productivity of NC programming and that of machining operations. With advanced integration between these technologies, significant benefits can be expected for the relevant industries. This paper proposes an approach for integrating advanced machine-tool canned cycles into CAD/CAM software. The integration associates the machining features of a CAD model with the corresponding cycles of machine-tools for which tool paths are directly processed into their respective controllers. An application prototype has been developed using PRO/Engineer which proposes an upgraded interface with new machining strategies based on available cycles related to specific machines. The machining parameters and the feature itself only need to be identified to generate the appropriate G-code through an adapted post-processor. The machining of prismatic parts is performed in order to validate the approach and to perform an evaluation of the integrated approach in comparison with a traditional NC programming approach using a CAD/CAM software application. The tests performed are based on circular pocket roughing and finishing, profiling of islands and surfacing. The machining of prismatic parts with the proposed approach shows interesting benefits for small batch production applications as compared to the case with CAM generic tool paths.

Key-Words: Computer-Aided Manufacturing, Integration, Machine-Tool, Post-Processor, Canned Cycles, Machining.

1 Introduction

CAD/CAM systems and CNC machines are widely used in the manufacturing industry. The integration level between these two technologies is a major element which directly affects the productivity of NC programming, and which is at the kernel of machining operations. The type and quality of the information transferred from the programming environment to the machine controller certainly constitute one element to consider in this quest for productivity enhancement.

The earliest technological developments in CNC machine-tools made available to customers include advanced canned cycles in standard CNC environments. When properly utilized or integrated into CAD/CAM environments, these functionalities can significantly improve the productivity of the complete manufacturing cycle. Currently very few of these proprietary macros seem to be integrated into CAD/CAM software applications using a feature-based manufacturing approach. Most macros are available through a G-code syntax, for instance, including all the dimensional and metal cutting parameters required to machine a specific feature.

The new developments proposed by the CNC industry also include open architecture controllers, which offer high flexibility for users wanting to program customized applications. However, these approaches involve proprietary programming environments [1], which constitute a severe drawback for every manufacturing industry seeking standardization and compatibility with complex multi-machine-tool environments.

Several research projects are however related to the standardization of languages or control architectures in CNC machining. Altintas and Erol [2] propose a library of functions specifically developed for open architecture controller developers. Zhang et al. [3] propose an architecture which refers to manufacturing features defined using the STEP standard EXPRESS language. Furthermore, certain programming languages have been developed to replace G-codes. Open-G [4] offers a modern programming structure compatible with G-codes, while the BCL (EIA/ANSI) [5] offers an APT-like syntax, which is directly readable by the CNC, and which complies with the standard. Another language, which is currently standardized, is the STEP-NC (ISO 14649). It proposes a complete machining information set from which trajectories are directly computed into machine controllers, which must comply with the standard [6]. The flexibility and performance evaluation of the combined STEP/STEP-NC approach is currently in progress and known as the “Super Model Project” [7]. The goal of all these innovative research developments is an improved CAM-to-CNC machine integration. These research endeavors and new standards will bring about productivity enhancement only if machine tool/controller manufacturers decide to fully support them.

In the meantime, manufacturing industries are currently using machine-tools equipped with interesting functionalities, which are nonetheless under-utilized due to a lack of integration with most of the CAM software applications through which relevant toolpaths are all commonly defined.

We propose an approach for dealing with this integration problem. The proposed integration aims to complement all the research works related to the manufacturing extraction problem, e.g., Chandrasekaran [8], Gupta [9], Regli et al. [10], or to the CAD/CAM approach based on machining features.

With specific respect to the integration of actual CNC technologies into the CAD/CAM world, Blumfield et al. [11] developed an automatic post-processor generator tool adapted to the custom cycles related to a sheet metal laser cutting machine. Zietarski [12] developed an advanced post-processor which integrates all the proprietary functions of a turning-milling CNC machine. Both of these approaches, based on specific machine technologies, are not closely linked to the CAD/CAM features of the model (not associative). Further, no uploading capabilities are consider in the approach, which could save, within the CAM machining model, all the feedback information derived from the fine tuning, or tool proving, operation on the machine. The proposed integration approach described in the next section is based on an associative concept, and stores the machining feedback information into the model as a reverse post-processor.

2 Advanced Canned Cycles Assessment

Figure 1 shows an example of an advanced canned cycle (ACC). This cycle is related to rectangular pocket machining, defined using geometric parameters (length, width, corner radius, depth) as well as machining parameters such as feeds, speeds, depths and widths of cut, etc. The ACC is defined using a G-code syntax, which includes all definition parameters. For the Hitachi Seiki Seicos 16M controller [13], the rectangular pocket machining is expressed as follows:

G328 X_Y_Z_R_I_J_K_Q_C_A_D_E_U_V_F_ ;

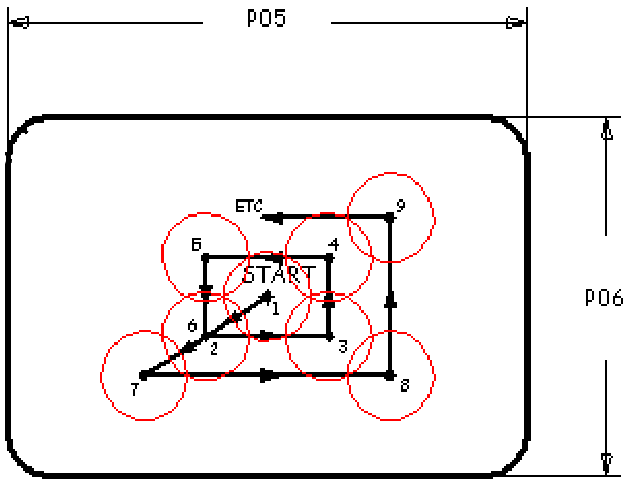

Fig. 1 Advanced canned cycle for rectangular pocket machining

The integration of ACC within CAM software applications requires a specific user interface from which all machining parameters can be defined. As shown in Figure 2, the proposed integration approach includes an ACC generator for any cycle definition available for a machine-tool. The generic CAM software is enhanced with additional menus related to the newly integrated macros, or ACC, customized based on machine-tools available in a manufacturing environment. Being integrated within the CAM kernel, the CL-file produced includes the specific syntax defined through the machining feature’s geometric parameters as well as the machining parameters which are input through the customized user interface. For example, the following generic CL-file syntax is related to rectangular pocket machining, as previously defined through the G328 NC machine syntax.

CYCLE/CAVITE,RECTANGLE,IN,XOR,$p1,YOR,$p2,PROF,$p3,CLRP,$p4,LONG,$p5,LARG,$p6,LC,$p7,PC,$p8,SC,$p9,RAY,$p10,OCOM,$p11,AF,$p12,SF,$p13,AZ,$p14,AE,$p15

Through this approach, a typical manufacturing industry could define its own ACC standard made available within its CAM software, depending on the machine-tool selected for developing a specific part machining toolpath. The post-processing environment translates the CL-syntaxes particular for each machine-tool which completely or partially supports the entire library of the CAM-integrated ACCs. The approach is innovative because it proposes a feedback from the machine-tool operating environment into the CAM system in order to update the NC program and the associated machining features, based on the tool proof process. Any modified ACC G-code syntax from the machine operator is uploaded into the part NC program and features through the reverse post-processor (Figure 2).

3 A CAD/CAM-ACC Prototype Application

An Advanced Canned Cycles NC programming environment has been developed using the PRO/Engineer V2000i2 development kit (PRO/Toolkit). Figure 3 shows a part of the user interface related to the ACC generator module. It shows the dialog boxes related to the circular pocket machining available with the Hitachi Seiki Seicos16M machine controller. The dialog boxes include all the input fields required for the machining parameter values, while the pocket geometry is extracted from the circular cut feature selected from the CAD model tree. From this approach, any change in the pocket size or depth will automatically affect the CL-file output related to the circular ACC. The approach is fully associative.

For our application prototype, the CAMPOST post-processor generator is used to translate the ACC CL-syntaxes into proper G-codes related to specific machine-tools. The reverse post-processor module developed, is responsible for uploading modified ACC G-codes within the related machining features of the CAM program. It consists of a syntax analyzer which looks for any differences that exist between the modified and the G-code programs initially supplied to the shop floor environment (Figure 4).

A total of 32 ACCs available with the Seicos 16M controller are integrated within the PRO/Engineer Manufacturing environment. The ACC families include special drilling cycles, different internal and external pocket machining, as well as certain special machining strategies, such as helicoidal and trochoidal machining.

4 Results

A comparison was made between the traditional machining approach which uses toolpaths defined with standard CAM software functionalities and an approach that refers to the CAM enhancement with ACCs. Both the NC programmer's qualitative appreciation and the machining time of a test part were used as comparison criteria. The test part included circular features from which ACCs are available with the Hitachi Seiki machining center (Figure 5).

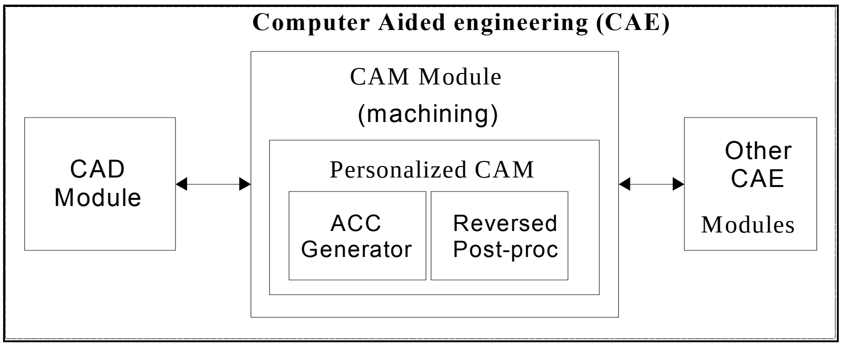

Fig. 2 Integration approach for the ACC

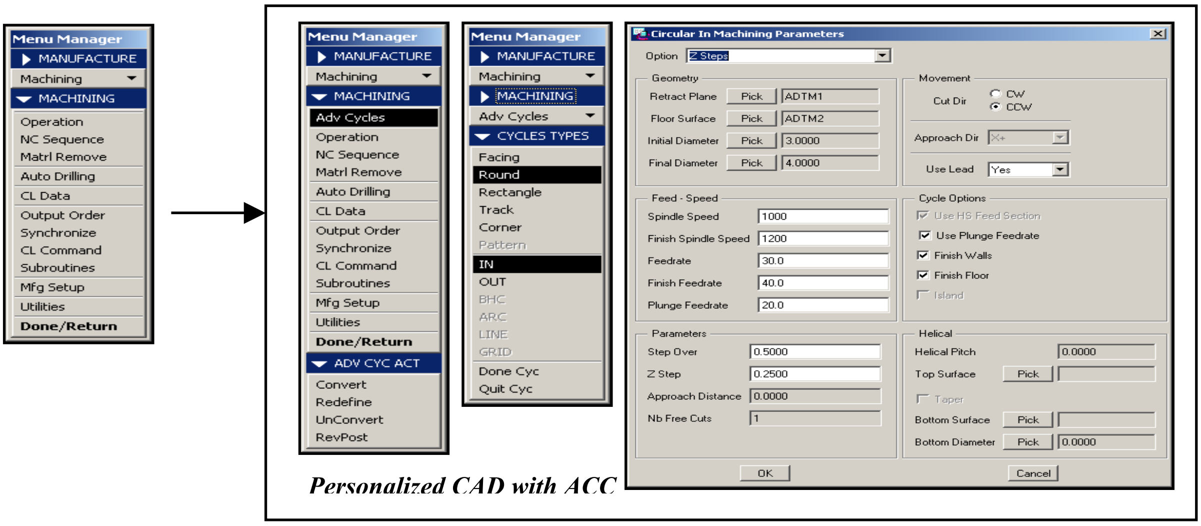

Fig. 3 User interface of the upgraded CAM system with ACCs

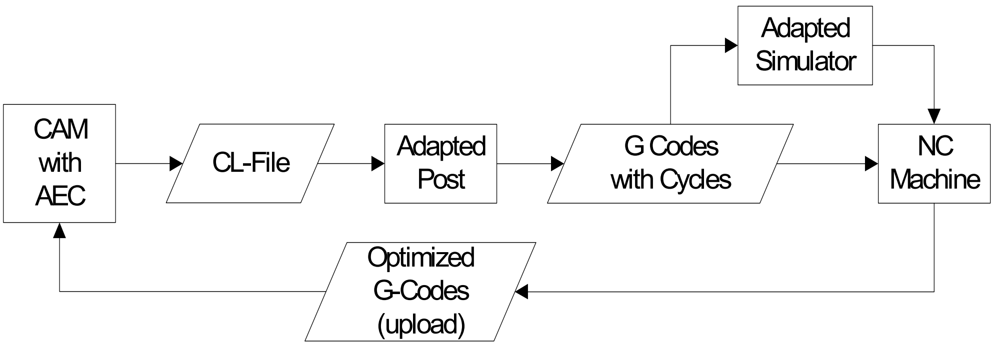

Fig. 4 Integration approach with reverse post-processor

Fig. 5 Test part machining using circular ACCs

Experience shows that ACC NC programs offer more legibility than classical NC programs. The NC programs are also reduced in length, which can be interesting for machine-tools with restricted memories. In fact, this memory limitation often requires a DNC mode of operation, which slows down the machining speed to the DNC transfer speed. On top of the shorter program length advantage, the operator can also visualize and easily modify the toolpath parameters through the ACC special G-code syntax. The enhanced CAM programming interface also reduces repetitive programming selections for the NC programmer. Overall, NC programming is more efficient with the ACC approach. With the reverse post-processor, there is also an improvement in the maintenance of the NC program using the ACCs.

Regarding the machining time, Table 1 summarizes all the machining steps required to machine the test part, along with their corresponding times, with and without the use of the ACC. Figure 5 refers to the numbered features in Table 1. The times include the feed and the rapid motions for each step. There is a difference of about 2 minutes between both approaches, the classical approach being the fastest with about 16 minutes of machining time. This difference is mainly due to the approach and retract motions at the start and end of each machining step. The ACC retract and approach motions are conservative, allowing a collision-free machining. Access to additional parameters for the approach and retract motions would correct this slight difference.

Table 1 Machining time compared using both approaches

Machining step	Classical approach	ACC approach
Facing until island #1	2 min 27 sec.	2 min 27 sec.
Roughing volume until surface #6	7 min 53 sec.	
Partial roughing		2 min 43 sec.
Roughing around island #1		4 min 12 sec.
Roughing around island #5		2 min 55 sec.
Pocket #4 roughing	34 sec.	25 sec.
Conical pocket #3 roughing	1 min 34 sec.	41 sec.
Conical pocket #3 finishing	38 sec.	1 min 01 sec.
Pocket #2 roughing	54 sec.	41 sec.
Pocket #2 finishing	11 sec.	14 sec.
Island #1 external finishing	30 sec.	1 min 12 sec.
Island #5 external finishing	29 sec.	1 min 08 sec.
Pocket #4 finishing	41 sec.	31 sec.
TOTAL TIME	**15 min 51 sec.**	**18 min 10 sec.**

5 Conclusion

A new integration approach between CAM software applications and the machine-tools is proposed to fully take advantage of the advanced functionalities already available with most existing NC controllers. The approach proposes a customized development within the CAM software in order to offer a user-friendly interface based on associative machining features. It also proposes a machine-tool for CAM software feedback information through a reverse post-processing module. This ensures the automatic maintenance of the NC program that is proven and modified at the shop floor level. The generic cycles integrated within the CAM functionalities are translated, using specific post-processors, into a machine language compatible with each controller which partly or completely supports the new functionalities. The experiment produces great benefits related to the small-batch production of parts with prismatic features. To achieve better productivity for medium to lager batch productions, the ACCs must include more flexibility in terms of rapid motion definition.

References

[1] R. Hillaire, “Whatever Happened to Open Controls?”, Manufacturing Engineering, 124(6), pp. 80-88, 2000.

[2] Y. Altintas, N.A. Erol, “Open Architecture Modular Tool Kit for Motion and Machining Process Control”, Annals of the CIRP, 47(1), pp. 295-300, 1998.

[3] L. Zhang, J. Deng, S.C.-F. Chang, “A Next Generation NC Machining System Based on NC Feature Unit and Real-Time Tool-Path Generation”, The International Journal of Advanced Manufacturing Technology, 2000(16), pp. 889-901, 2000.

[4] Engels, W. Daniel, Sarma, E. Sanjay, “Towards an Open Architecture Specification Language for Machine Control”, SPIE Conference on Sensors and Controls for Intelligent Machining, Vol. 3832, 1999, pp. 17-25.

[5] EIA/ANSI, 32 bits binary CL (BCL) and 7 bit (ACL) Exchange input format for numerically controlled machines, EIA/ANSI 494 Standard, 1992.

[6] M. Hardwick, D. Loffredo, STEP into NC, Manufacturing Engineering, 126(1), pp. 38-50, 2001.

[7] STEP Tools, Inc. Super Model Project, http://www.steptools.com/library/stepnc/-smp.html.

[8] N. Chandrasekaran, Generation of Feature-Based Models, Technical Research Report, University of Maryland, Maryland, 1994.

[9] S.K. Gupta, Generation of Alternative Feature-Based Models and Precedence Ordering for Machining Applications, Technical Research Report, University of Maryland, Maryland, 1992.

[10] W.C. Regli, S.K. Gupta, D.S. Nau, Extracting Alternative Machining Features: Algorithmic Approach, Technical Research Report, University of Maryland, Maryland, 1995.

[11] A. Blumfield, M. Shpitalni, E. Lenz, "A Generator for Creating Adaptive Post-processors", Annals of the CIRP, 41(1), pp. 527-530, 1992.

[12] S. Zietarski, "System Integrated Product Design, CNC Programming and Post processing for Three-Axis Lathes", Journal of Material Processing Technology, 109(2001), pp. 294-299, 2001.

[13] Hitachi Seiki Corporation Limited, Seiki-Seicos Σ 10M/16M/18M, Instruction Manual programming, 1998.