Creating a Multi-axis Machining Postprocessor

Petr Vavruška[1]

[1] *Department of Production Machines and Equipment, Faculty of Mechanical Engineering, Czech Technical University in Prague, Horská 3, 128 00 Prague 2, Czech Republic*

Correspondence to: p.vavruska@rcmt.cvut.cz

Abstract

This paper focuses on the postprocessor creation process. When using standard commercially available postprocessors it is often very difficult to modify its internal source code, and it is a very complex process, in many cases even impossible, to implement the newly-developed functions. It is therefore very important to have a method for creating a postprocessor for any CAM system, which allows CL data (Cutter Location data) to be generated to a separate text file. The goal of our work is to verify the proposed method for creating a postprocessor. Postprocessor functions for multi-axis machining are dealt with in this work. A file with CL data must be translated by the postprocessor into an NC program that has been customized for a specific production machine and its control system. The postprocessor is therefore verified by applications for machining free-form surfaces of complex parts, and by executing the NC programs that are generated on real machine tools. This is also presented here.

Keywords: postprocessor, NC program, interpolation.

1 Introduction

Most CAM (Computer Aided Manufacturing) systems are able to export general data, called CL data (Cutter Location data), in a text file. The CL data file consists of information about the coordinate system, tools, etc. It is necessary to adjust this CL data file for a specific production machine. Each production machine has a different structural design and a unique control system. A file with CL data is then translated by the postprocessor into an NC program that has been adapted to a specific production machine and its control system, see [3].

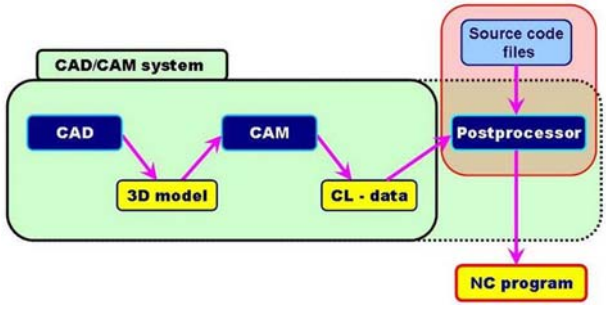

Figure 1: NC program creation process

Especially for multi-axis machining, it is necessary to include further features for generating the relevant NC programs, e.g. the location of the rotary axes of the machine tool. However, when using standard commercially-available postprocessors, it is often very difficult to modify its internal source code and it is a very complex process, in many cases even impossible, to implement the newly developed functions. It is therefore very important to be able to create a postprocessor for any CAM system, which generates CL data to a separate text file. Subsequently, the postprocessor that is created must be verified by applications for machining free-form surfaces of complex parts, by executing the NC programs that are generated on a real machine tool.

2 Method for creating a postprocessor

The postprocessor falls within the chain shown in Figure 1, which illustrates the idea that the postprocessor is not necessarily a part of the CAM system, but can also be a stand-alone program, and is in essence a compiler. The issue of compilers is taught e.g. at the Faculty of Electrical Engineering (CTU in Prague), and can be found in [2]. The CL data file contains all necessary information for creating a specific NC program. Besides the coordinate system and tools mentioned above, a CL data file consists of information about the toolpath, the orientation of the tool, tool changes, the feed-rate value, the spindle speed value and the direction of rotation, the desired cooling, the type of interpolation, etc. A method for creating a postprocessor for CL data analysis has been proposed. This method is based on using software that can be found as open-source or free software. FLEX and BISON software generate source codes in C language for lexical and syntactic analyzers on the basis of lexical and syntactic analysis

rules created by the programmer. The software is distributed under General Public License (GPL) and forms part of the GNU project. Lexical and syntactic analysis is needed to decode the input text file (CL data). After this decoding process, the data is passed to other postprocessor functions, where it is further processed according to the programmed algorithms. Other postprocessor functions (transformations, modality of words, block formatting, etc.) will be written in the source code of C language as a separate text file. The final software that we need is a compiler of source codes written in C language to obtain a binary file of the postprocessor. A compiler called Dev-C++ is used for this purpose, but of course a different C language compiler can be used. This is summarized in Figure 2.

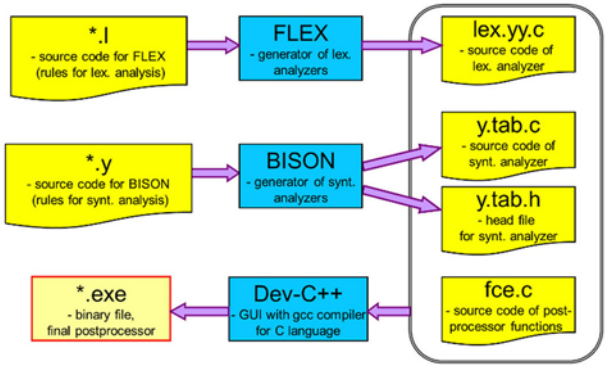

Figure 2: Postprocessor creation method

3 Postprocessors for multi-axis milling

For more than three-axis machining, it is necessary to transform the coordinates of the tool reference point in each block of CL data. The transformation depends on the nomenclature of the actual machine tool. It is also necessary to consider the required machining principle. In multi-axis machining there are two basic machining principles: positioning (indexing), and continuous machining. In the case of positioning, the workpiece is set to the desired position using rotary axes, and the cutting moves are realized using linear axes only. The postprocessor then transforms the coordinates according to the same angular coordinates. The limits of the rotary axes are checked only in the block by setting the desired positions of the rotary axes. Whereas in continuous machining it is necessary to check the positions of the rotary axes of the machine tool in each block, in the case of four-axis machining we can speak of plane coordinate transformation (the coordinates of the linear axis, which is parallel to the axis of rotation, remain unchanged), and the axis of rotation also usually does not have limits. Figure 3a shows the nomenclature of the Haas TM1 four-axis machine tool, which is available at the Department of Production Machines and Equipment (CTU in Prague, Faculty of Mechanical Engineering).

Five-axis machining is a larger issue. In this case we must consider the spatial transformation, which is ambiguous. In a CL data file, we can find positions of the tool that may be represented by two identical positions on the machine tool, see Figure 4a. We know only the position of the tool reference point and the orientation of the tool axis, and we look for the position of all machine tool axes. This is an application of inverse kinematics. Multi-axis machine tools are distinguished by their nomenclature (location of the rotary axes), and hence transformation equations can be created. The transformation equations are usually created using transformation matrices, which can be found e.g. in [4]. Figure 3b shows the kinematics of a five-axis machining center, which is available at the Department of Production Machines and Equipment (CTU in Prague, Faculty of Mechanical Engineering). It is based on the MCV 1000 three-axis milling machine tool (made by Kovosvit MAS), and is supplemented by the rotary/tilting table made by Nikken. In this case, the two rotary axes are on the work-piece clamping device. This is known as the table-table concept. The transformation matrices for this machine can be found in [5].

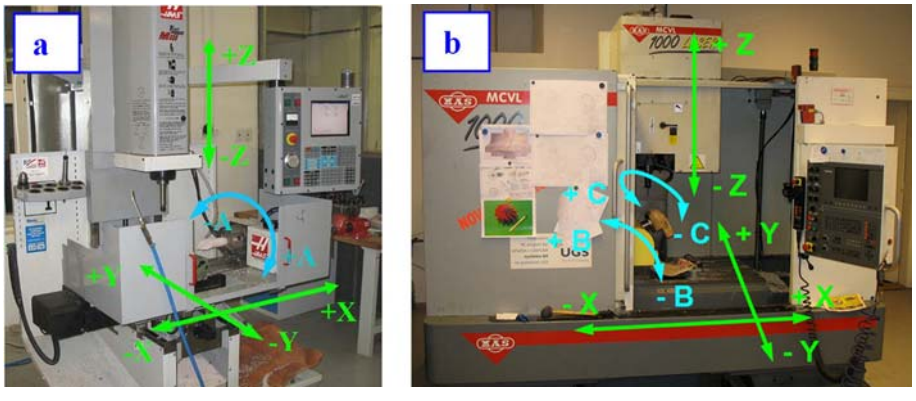

Figure 3: Nomenclature of machine tools (a – four-axis Haas TM1, b – five-axis MAS MCV 1000)

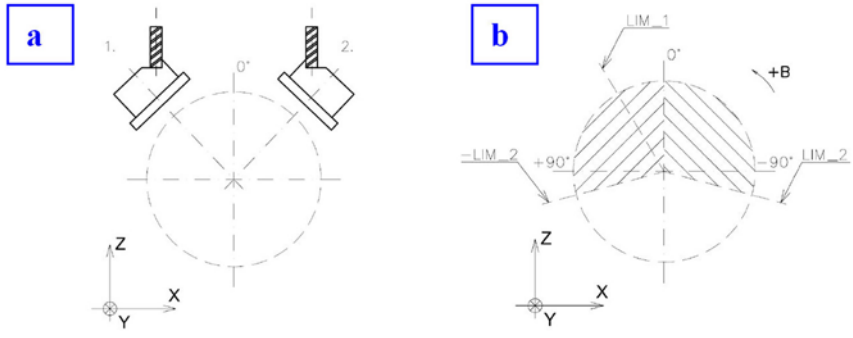

Figure 4: Range of movement of the tilting axis (a – ambiguous positions, b – limits of tilting axis)

Postprocessors for continuous multi-axis machining must include solutions for the limits of the rotary axes. It is usually possible to specify the limits of machine tool axes and then during transformation the postprocessor checks the positions of the axes in the whole machine tool. However, in the event that the subsequent position of some rotary axis would exceed the limit of this axis, and in the next NC program block the postprocessor generates the positions of the alternative machine tool axes (and they do not exist), without the tool retracting from the workpiece in order to change the positions of the machine tool axes, then a collision might occur between a tool and a workpiece. This is because a linear interpolation is programmed between the two consecutive blocks in the NC program. Other places in the NC program where this problem may occur are called stationary points, where the tool axis is parallel to the rotary axis C (in the case of the five-axis machining tool mentioned above). In fact, there is no line during multi-axis machining using linear interpolation, but a spatial curve is created because of the existence of rotary axes. In the case of very short increments of the toolpath, the curve approaches a straight line, but when the angular coordinate is too high the curve is visible on the surface and the workpiece is devalued. However if these critical points are dealt with using a proprietary algorithm, the impending destruction of the product is averted. The limits of tilting axis B of the MAS MCV 1000 machine tool are shown in Figure 4b as "LIM_1" and "LIM_2" ("-LIM_2" is the mirrored "LIM_2", as an area of possible ambiguous solutions for the position of the tilting axis). Figure 5 shows a difference between the characteristics of the coordinates of two NC programs, the first without using the algorithm for flipping of the tilting axis over the stationary point (characteristics of the coordinates in Figure 5a), the second with this algorithm (characteristics of the coordinates in Figure 5b). The green box in Fig. 5b surrounds the area from the stationary point to the point where the limit of tilting axis B is reached. If the tool is at a safe distance from the workpiece, the machine tool axes can change the positions, and the machining process can continue after the tool approaches the workpiece again. The approach is implemented in such a way that at first the tool moves at a rapid speed in the position near the surface, and then it moves at the working feed-rate to the cut.

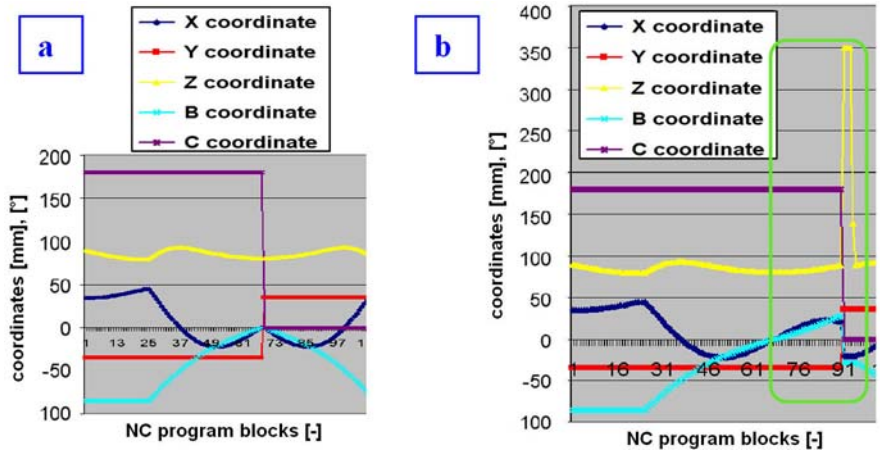

Figure 5: Usage of the algorithm for flipping over the stationary point (a – without, b – with)

Figure 6: Testing the four-axis milling postprocessor

The format of the NC program depends on the control system used in the machine tool. There is also a common language for NC code — the ISO language (ISO code), but the producers of the control systems are still trying to bring new features in their systems. In order to generate these functions in the NC program correctly, the postprocessor has to identify areas in the CL data file where these functions can be used. The postprocessor can be extended by non-standard functions that are derived from specific requirements in production, or from practices within the company where the postprocessor will be used. In these two cases it was necessary to adjust the NC program to the requirements of the Haas control system (in the case of Haas TM1) and to the requirements of Heidenhain iTNC 530 (in the case of MAS MCV1000). An adjusted ISO format had to be used for the Haas control system. The part (blade) for testing the postprocessor functions is shown in Figure 6, together with a part of the NC program (the block with the tool change is marked in red, and the blocks with tool length and radius compensation are marked in green) and the real machining of the blade.

Heidenhain control systems allow for programming using the adjusted ISO language, and along with it they also offer programming in their own language called "dialogue", see [1]. This language is very different from the ISO code. Due to the machine operator requirements, it was necessary to modify the postprocessor so that the format of the NC program corresponds with this language. A comparison between a CL data section and the corresponding section of an NC program is shown in Figure 7, where the blocks that correspond to each other are marked in blue and green, and the feed rates programmed through the parameters are marked in red.

The final figures show a test of the postprocessor. The creation of toolpaths (for roughing and for finishing) is shown in Figure 8, where the test part is an impeller. After this part was machined on a machine tool (see Figure 9a), the surfaces were measured using a Mitutoyo coordinate measurement machine, see Figure 9b. It can be surely stated that the postprocessor algorithms for multi-axis milling have been verified.

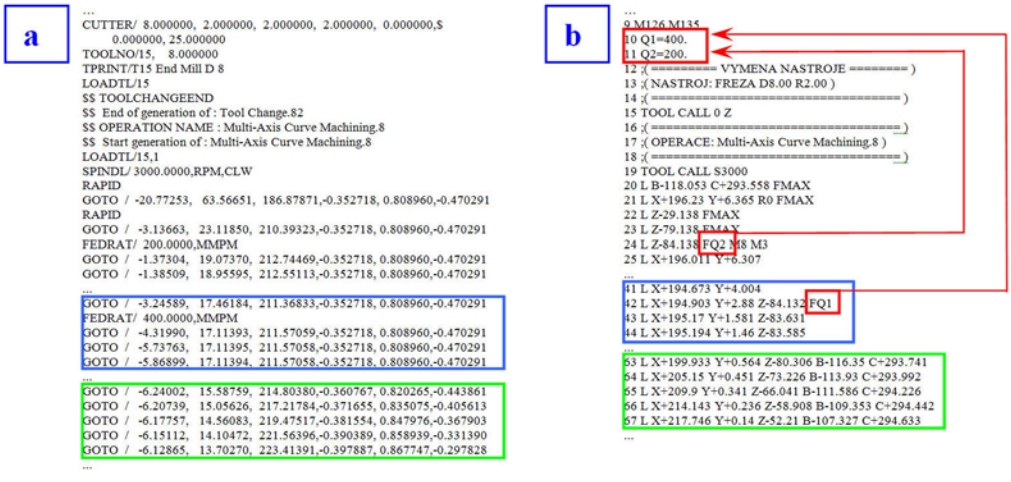

Figure 7: CL data in comparison with the NC program (a – CL data, b – NC program)

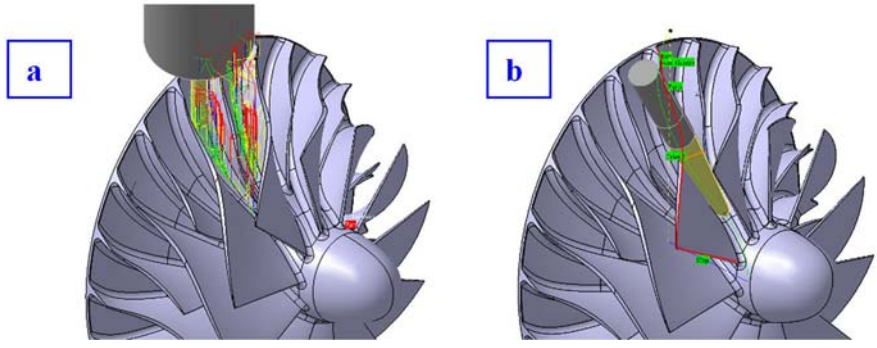

Figure 8: Testing toolpath creation (a – roughing, b – finishing)

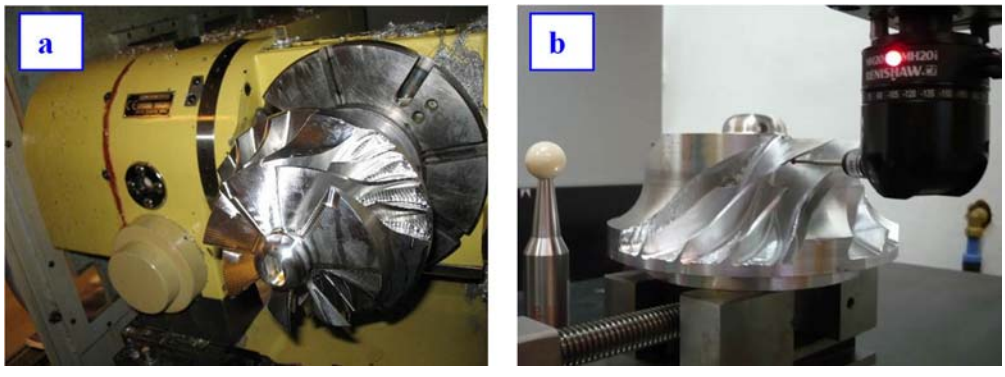

Figure 9: Finished blade (a – after machining, b – during the measurement)

4 Conclusion

This paper has summarized my findings from designing and verifying the multi-axis machining postprocessor creation method. The method is based on translating a text file with CL data. A CL data file is generated by the CAM system, and then it is translated by the appropriate postprocessor into an NC program. The appropriate postprocessor must be created for a specific CAM system, a particular machine tool and a control system. The multi-axis postprocessor method has been verified by the CATIA CAM system. An experiment machining real parts on CNC machine tools proved the functionality and the applicability of the postprocessor algorithms. This postprocessor creation method can also be easily applied (after analysis of the actual format of the CL data and adaptation of the lexical and syntactical analyzers of the postprocessor) to other CAM systems.

Acknowledgement

The results have been obtained with the support of the 1M0507 Czech Ministry of Education, Youth and Sports grant "Research Center of Manufacturing Technology".

References

[1] *iTNC 530, příručka uživatele – programování podle DIN/ISO.* Traunreut : Heidenhain, 2002. 463 p. (in Czech)

[2] Melichar, B. et al.: *Konstrukce překladačů. I. a II. část.* 1. edition, Prague : CTU, 1999. 636 p. ISBN 80-01-02028-2. (in Czech)

[3] Rybín, J.: *Automatické řídicí systémy.* 1. edition, Prague : CTU, 1991. 150 p. ISBN 80-01-00694-8. (in Czech)

[4] Stejskal, V., Valášek, M.: *Kinematics and dynamics of machinery.* 1. edition, New York : Marcel Dekker, INC., 1996. 494 p. ISBN 0-8247-9731-0.

[5] Vavruška, P., Rybnín, J.: *Postprocessing and its applications on free-form surfaces.* Research report no. V-08-088. CTU in Prague, Research Center of Manufacturing Technology, 2008. 54 p. (in Czech)