Preprints of the 9th International Symposium on Robot Control (SYROCO'09)
The International Federation of Automatic Control
Nagaragawa Convention Center, Gifu, Japan, September 9-12, 2009

STEP-NC Based Industrial Robot CAM System

B. Solvang*, L. K. Refsahl, G. Sziebig*****

**Narvik University College, Narvik NO-8514 NORWAY (Tel: +4776966227; e-mail: bjs@hin.no).*
***Narvik University College, Narvik NO-8514 NORWAY (Tel: +4795867555; e-mail: lakrre@hotmail.com).*
****Narvik University College, Narvik NO-8514 NORWAY (Tel: +4776966323; e-mail: gabszi@hin.no).*

Abstract: In this paper a computer aided manufacturing (CAM) system for industrial robot machining operations is introduced. The system is based on the new machining standard STEP-NC which is rapidly making its way into the world of numerical controlled (NC-) machines (e.g. milling and turning machines). Existing research, related to STEP-NC development, focus mainly on the typical implementation of the standard into these traditional production equipments. However, industrial robots are getting more and more capable of taking onto machining operations and it is necessary to couple the industrial robot towards the new standard.

Keywords: Industrial robot, STEP-NC standard, offline programming, flexible manufacturing systems

1. INTRODUCTION

In manufacturing systems the key role of an industrial robot has been material handling and welding. A survey in Guizzo, (2008) shows that handling covers 35,4 % of all applications while welding covers 28.9 %. At the other end of the scale we find cutting and milling applications with only 2,5% of all the robot installations.

Subtractive machining (e.g. polishing, grinding, milling, turning etc.) has been challenging for the industrial robot, mainly due to the lack of stiffness in robotic arms (Zhang et al., 2005), but also because the industry did not focus much onto such application areas. Today, stiffer and stronger structures are available, mainly through the parallel structured manipulators (e.g. hexapods). However, the serial linked manipulator is as well improving, via stiffer structures, but also through more advanced process control. In example force sensing and control can be utilised to keep the overall stress on the robot manipulator within certain limits.

Initially, machining operation starts with a design stage where computer aided design and computer aided manufacturing (CAD/CAM) software is used to generate a computer model and the necessary cutter movements to produce this model, from a given starting geometry (raw material). The generated cutter location file (CL-file) needs then to be translated (post-processed) to machine specific language. In the case of robot machining this is a challenge because robots do not share a common language. Robot languages are vendor specific and often even model specific. As a result, CAD/CAM software, generally, do not support post-processors for robots. Production companies with several kinds of robots are having a hard time to handle all these languages and self-development of post-processors is time consuming. Program exchange from one robot to another is often impossible and seen from the small and medium sized enterprises (SMEs) the whole idea of using robots in machining applications is quite challenging.

For the typical NC-machine the development has been a bit different. From the very beginning a common language standard ISO 6983, (1982) (often referred to as G-code) has been available. CAD/CAM software has as well been building around this ISO standard and the necessary connection between software and the different machines has been commonly available.

However, the ISO 6983 standard has met criticism related to the fact that it is a low level language which outputs a long code list, hardly understandable and difficult to edit. G-codes represents simple movements of machine axes and do not relate these movements to the work piece geometry. In other words, there is no clear connection between the generated code and its impact on the item to be machined (Zhu et al., 2006). To deal with this, a new standard ISO 14649 (ISO 14649-1, 2003), referred to as STEP-NC, was introduced in 2003 and represents a new paradigm in programming of machining operations. STEP-NC is a high level object oriented programming language where the work-piece geometry and the machining task are strongly connected. STEP-NC based files specify the steps of the machining process rather than information on the cutter location movements alone (Wu, 2006). Around the world various NC-machine and software manufactures are working to implement the STEP-NC in their products. Large companies and many research units are working with implementations and experimental issues related to the new standard: Xu et al., (2004), Newman et al., (2008), Xu, (2007), Suh et al., (2003).

It seems that most development and research related to the STEP-NC standard focus onto the classical NC-machines (milling-, turning machines etc.) and there is a lack of initiatives towards the industrial robot.

So, this paper seeks to couple the new standard towards the industrial robot and the article is organised as follows: In paragraph 2 an overview of a framework for STEP-NC/Robot path generation is laid out. In paragraph 3 follows some system specifics. Experimental results are given in section 4 while paragraph 5 concludes the paper.

2. STEP-NC BASED ROBOT PATH PLANNING

In this section the robot path generation based on STEP-NC standard is laid out. The methodology is adapted to certain software solutions but the concept may be transferred to other environments as well.

A small introduction to the STEP-NC standard is given before the robot CAM system is introduced.

2.1 Introduction to STEP-standards

As already mentioned in the introduction the bottleneck of the older programming standard is the low level language, due to the old architecture, which is still limited in computational capabilities. The existing standard requires machine specific post-processors to translate the high level language to a lower level machine code. This one-way transformation converts the work-piece geometry to machine axis movements. The old programming approach specifies how (axes movement only) instead of what (work-piece geometry related) to do (Xu et al., 2004). The problem with this approach is that the geometry information is lost and from the tool axis movement it is nearly impossible to recreate the work-piece geometry. This effects program portability from one machine to another machine (axis movement in one machine do not correspond to movements at another) and also the change tracking possibility (any change on the final, translated machine code cannot be converted back to work-piece geometry).

The ISO 10303 (ISO 10303-1, 1994), introduced in 1994, is meant to replace the variety of geometry description languages (e.g. IGES, SET, VDA, VRML, PRO) and serves as a standard to describe any geometrical object. ISO 10303 standard, which is built on an object oriented language (so called EXPRESS (ISO 10303-11)), then eliminates the need for different geometry converters. The full name of the standard refers to this: Standard for the exchange of product model data (STEP). By using a general description language for geometrical modelling, the generalisation of the connected machining operations was expected next.

So, first a sub-standard, named as ISO 10303-238 (ISO 10303-238, 2004), was established and was followed by a standalone, but compatible standard, named ISO 14649. This new standard carries both the information of work-piece geometry and the planned machining operation. All data can be stored in files following a predefined structure (normal ASCII text file (ISO 10303-21, 1994) or XML file (ISO 10303-28, 2003)). It can also be directly stored in a database. Every file can be opened and edited by suitable CAD/CAM/CAPP software to make the necessary changes, or it can even be edited directly on the manufacturing machine (if capable). Edited files/changes can be fed back to the designer at any stage. This possibility is the most important change since the appearance of the ISO 6983. As mentioned, editing of G-codes on the shop-floor has unpredictable changes to the original design. The new STEP family standards result in a machine independent language.

However this makes the NC machine vendor's task somewhat harder. These vendors have to provide a STEP-NC interface for their machines since the planning of the tool-path is normally expected to be done internally, resulting in a more complex controller. As already mentioned, initiatives for such implementations can be found around the world, but applications are generally for classical subtractive manufacturing machines and not targeting the industrial robot.

2.2 STEP-NC based robot CAM system

Fig. 1. summarises the layout of the proposed Robot CAM system. The main contribution and focus in this publication is the Robot CAM module (C) itself, but for the sake of completeness information on the other modules are given as well.

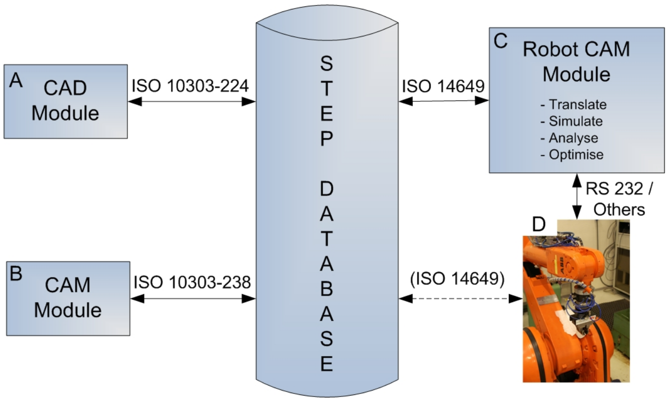

Fig. 1. From design to robot manufacturing, through STEP-NC standard.

First of all, it will be necessary to have a fully operating CAD Module (A) capable of geometry handling according to the STEP standard. In this module, the design of the work-piece, to be manufactured, is carried out. Next, a STEP based CAM Module (B) capable of generating the "cutting orders" and add it to the generated STEP geometry is necessary. Several manufactures of CAD/CAM software are following the STEP-NC development closely and are likely to implement STEP-NC file generation in their CAM-modules. Input to the Robot CAM module (C) is the STEP-NC based manufacturing orders. In our implementation we have selected to develop the CAM module within a standard offline programming and simulation software. In general such software has a common language for programming and visualisation of all the supported machinery. Machine specific translators (post-processors) are available for generation of the necessary machine code. In specific, we have developed our system in the DELMIA Digital Manufacturing & Production software IGRIP® and VNC®. In this environment, a graphic simulation language (GSL) is commonly used to visualise the movements of the programmed machinery. Machine code translators are available. The goal of Robot CAM module is to translate the STEP-NC code into commonly understandable language (GSL code) in order to simulate, analyse and optimise the generated robot machining trajectory. Finally, the real robot can execute the machining orders, as indicated in Fig. 1. (D).

3. SYSTEM COMPONENTS

This section will present some system details of the suggested Robot CAM module; from reading and parsing a STEP-NC file, running a simulation and finally carry out a machining process with an industrial robot.

Fig. 2. clearly indicates the overall processing procedure. The process contains of these four main steps; Read and parse file; Generate tool path with reference to work-piece coordinate system K_W; Path verification and optimisation through simulations, with reference to the virtual world coordinate system K_V; Path execution, with reference to real world coordinate system K_R. As indicated in the figure the resemblance between the virtual- and the real world should be within accepted tolerances, $K_V = K_R \pm TOL$, otherwise calibration procedures must be undertaken.

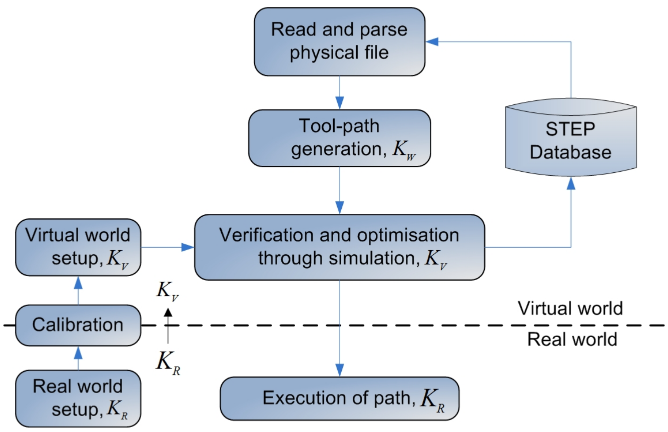

Fig. 2. Robot CAM structure

3.1 Reading and parsing the STEP-NC file

The process starts with a STEP-NC file which contains information about a stock piece to be milled. The STEP-NC file is an ISO 10303-21 file and contains information about tools to be used, geometric features, tolerances, material properties and other information such as "author" and production dates. In this STEP-NC file each line is a command. Each command can contain references to other lines in the file-structure, or it can contain information which controls the milling of the work piece. The structure of the STEP-NC file can be represented like a tree which gives an easy visual representation of the structure, as seen in Fig. 3.

The STEP- NC file is read from a text file and loaded line for line into the program internal memory. Here it is stored to give an easy access for further interpretation. When the program has uploaded the file in its memory, it evaluates each line and its underlying structure or say branch, seeking to determine what kind of information is presented and what to do with this information. The program starts at the top and works it way up and down until it has evaluated each command in the STEP- NC file. In such a way, information about each feature that describes the final geometry of the work- piece is determined.

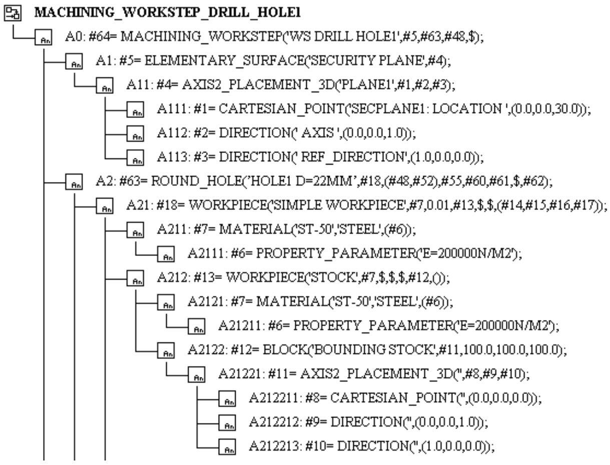

Fig. 3. Tree structure of STEP-NC file

3.2 Tool-path generation

The STEP-NC file does not contain any tool paths; therefore the GSL program has to generate such paths in syntax understandable for the industrial robot. In order to have a fully functional Robot CAM module, translators between STEP-NC and the GSL language should be developed for all STEP-NC commands. Currently we are working with this implementation of the complete standard, and as a typical example the algorithm for milling of a closed pocket will be described in some detail.

A Closed Pocket, as shown in Fig. 4, is a typical geometric feature in a STEP-NC file. Milling of such pocket is to be carried out via two machine operations, named Contour- and Bidirectional- movement. Contour is the movement between the four corner points P1 to P4. Dependant on the tool diameter, Contour operations will leave some material in the middle. Such material is then often removed by the Bidirectional operation (ISO 14649-11, 2004). In Fig. 4. the Bidirectional movement is shown as a line which is indicated with START and STOP.

To be able to mill with Bidirectional movement a tailor made algorithm had to be created in the Robot CAM module. The four corner points P1 to P4, the tool diameter, the pocket depth and tool cut depth are general input parameters to the algorithm. The Bidirectional movement, as described in the ISO 14649-11 standard, consists of a zigzag motion with four directional changes.

The algorithm to calculate these directional changes contains three major steps, first calculating the four corner points in the pocket, then calculating number of sideways steps necessary to do the milling and at last calculating the specific points in the tool path. The number of sideways steps is used to decide one of two cases of formulas to calculate the necessary direction changes and the resulting points. The two cases have the same general structure. Some details about this general structure is given below.

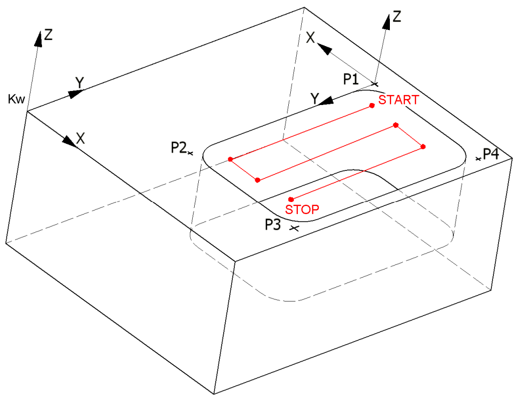

Fig. 4. Geometry of Closed pocket and Bidirectional tool movement

The four direction changes appear in a cyclic order, e.g. repeating each other every fourth time. A method for getting these four cyclic directional changes was created using two integer counters (T and G) and its combinations of odd and even. In general, the algorithm works by adding 1 to the integer G at each direction change, and adding 1 to the integer T at every second direction change, and then examining the combination of odd and even. By following this method, the next direction at any point can be decided by using Tab. 1. Although case Odd-Odd and Even-Odd both give movement in –X direction, the algorithm use a different value to calculate the Y coordinates. This method of deciding the direction of the next movement, together with the other general input parameters is used to calculate the next point in the tool path. Tab.1. shows the correlation between odd and even and the direction in which to place the next point, using the local coordinate system of the pocket.

Table 1. Relationship of cutting direction for bidirectional movement

T	G	Direction
0	0	-
Odd	Odd	-X
Odd	Even	-Y
Even	Odd	-X
Even	Even	Y

Fig.5. shows the integers used and the resulting coordinates to each point in the tool path. Positions P1” to P4” is at the same positions as P1 to P4, only adjusted into the pocket by half the diameter of the tool in use.

The movement between the two first points in the tool path is treated as special cases in the algorithm. To stop the algorithm, the next calculated point is compared with the geometry of the Closed pocket and if it is outside the geometry the algorithm is stopped and the tool path is complete. This algorithm only calculates the X and Y coordinate. The Z position is decided by the cutting depth of the tool in use. The algorithm is run a specific number of times, according to the possible cutting depth of the tool and the desired depth of the pocket. At each point in the tool path, the GSL program sets up a coordinate system (tag-point). These tag points are then stored in the GSL programs internal memory, in the order of which the robot has to move.

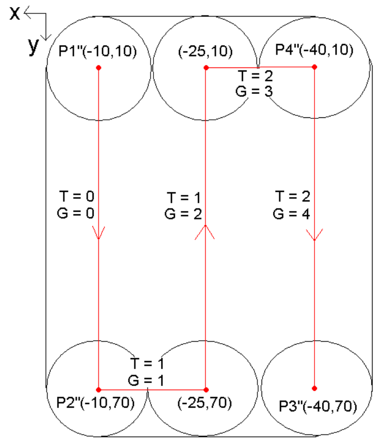

Fig. 5. Tool path, coordinates and correlating integers

3.3 Verification and optimisation through simulations

In the previous section the generated tool- path is stored with reference to the work-piece coordinate system K_w. By such an arrangement the generated path is portable together with the work-piece.

In this paragraph we suggest to bring the work-piece and the generated tool path into a standard visualisation and optimisation module (typical robot offline programming software). In such 3D environment, a replica of a real manufacturing cell is build up and virtual robots are commanded to follow the generated tool-path. In this environment we can check the robot movements for singularities, out of reach, or collisions. Lead times can be measured and the result of the machining operation thoroughly inspected. Fig. 6. shows a typical virtual world robot manufacturing environment.

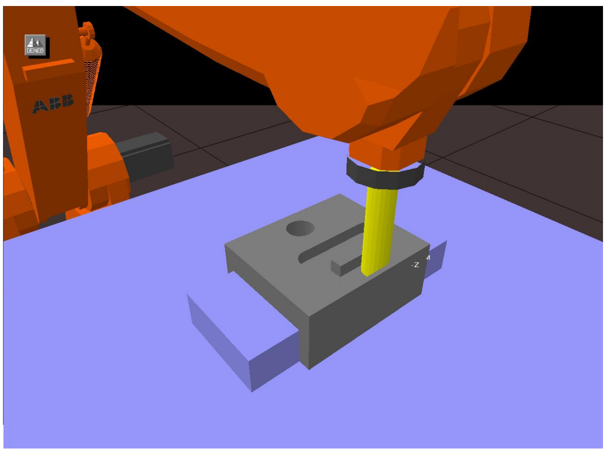

Fig. 6. Robot milling operation

During simulations it may occur situations were the operator wants to modify the generated machining path. However, it is not enough to simply update the robot movements, changes must be reflected in the STEP-NC file as well. Updating the STEP-NC file at any stage of the production cycle is a very important feature of the STEP concept. Modifications made on the shop-floor should be made visible for all participants in the manufacturing process. Our CAM module has implemented such STEP-NC editing and updating possibilities for the selected geometrical features: Closed pocket and Hole. Change of diameter, depth, location and geometry for such objects is possible and the STEP-NC file is updated accordingly.

In the virtual environment, the tool-path coordinates are presented with reference to the virtual world coordinate system K_V. As mentioned, it is of outmost importance that the virtual world coordinates K_V and the real world tool coordinates K_R are within accepted tolerances $K_V = K_R \pm TOL$, otherwise calibration procedures must be undertaken. Solvang et al., (2008) gives recommendations on how to locate the work-piece coordinate system K_w with respect to the real robot base coordinate system K_R.

Finally, after simulations and code optimisation the tool-movements can be translated into machine specific language and transferred to the real robot manufacturing cell.

4. EXPERIMENTS

An STEP-NC file has been used to carry out a simulation experiment with a virtual industrial robot acting as a milling machine.

In the selected work-piece and corresponding STEP-NC file (collected from ISO 14649-11, Annex F (2004)), milling of the experimental work piece is made up by five working steps. These working operations are Planar Face, Drill Hole, Ream Hole, Rough Pocket and Finish Pocket, see Fig. 7 a). Our simulations are concentrated on the last four working steps, since these gives the most visible results. There are used three milling tools, all with the same dimensions as the tools described in the STEP-NC file in the ISO standard. These tools are Endmill-20mm, Drill-20mm and Reamer-22mm.

Fig. 7 b). and c). shows the finished result after the four steps have been carried out on a stock work piece. The virtual model represents an identical geometry as found in the STEP-NC input file.

Second, the generated robot tool-path was downloaded to a NACHI SA-130F robot which was set-up as a XY-plotter (Fig. 8.).

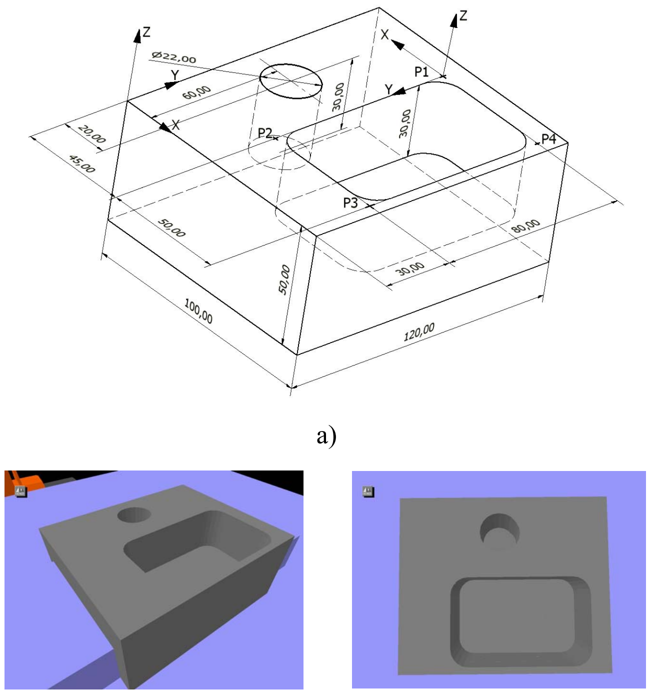

Fig. 7. a) 3D design model (based on ISO 14649-11 Annex F Example 1), b) and c) after virtual machining

The generated tool-path was drawn by the robot on a millimetre-scale paper. The resulting tool-path is shown in Fig. 9. Line lengths were measured and compared with the theoretical lengths found in the STEP-NC input file. All measurements were found identical to their theoretical counterpart. Accuracy of these measurements is dictated by the accuracy of the millimetre-scale paper.

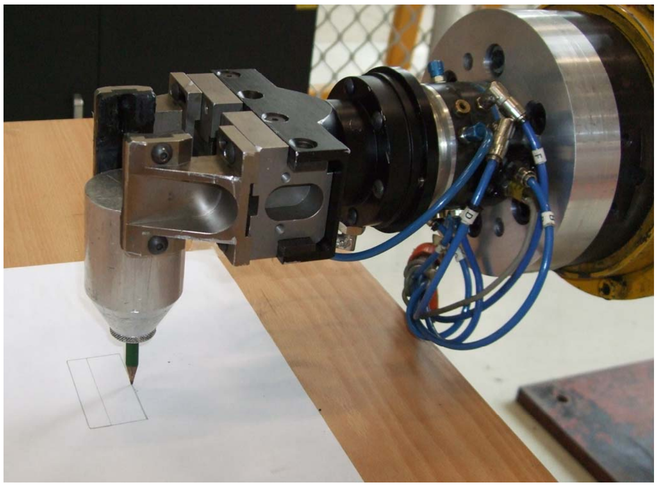

Fig. 8. Experimental setup

Experiments were carried out without any material cutting actions. However, influence on accuracy from the machining process is heavily dependent on the: 1) selected process parameters and 2) the capability (stiffness) of the chosen robot system. Although, these are very important parameters on the overall machining accuracy they do not influence on the accuracy of the Robot CAM system itself.

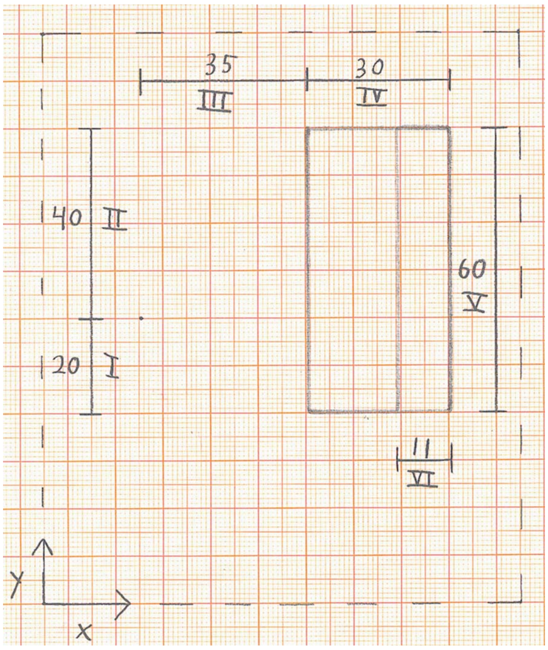

Fig. 9. Resulting robot tool-path

5. CONCLUSIONS

In this paper a Robot CAM module for industrial robot machining operations is introduced. The system is working towards the new machining standard referred to as STEP-NC, which is rapidly making its way into the world of numerical controlled (NC-) machines. By conducting research on the merging of robot systems towards the new standard we may expect to see more robots in machining operations in a near future. Experiments shows that the suggested Robot CAM module can, based on a STEP-NC input file, interpret, translate and transfer the manufacturing orders to an industrial robot.

ACKNOWLEDGEMENT

The research was supported by HUNOROB project (HU0045), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian Focal Point.

REFERENCES

Guizzo E. (2008). The Rise of the Machines. *IEEE Spectrum*, Vol. 45, No. 12, pp. 60, ISSN 0018-9235.

International Organization for Standardization (1982). ISO 6983-1, Numerical control of machine-program format and definition of address words Part-1: Data format for positioning, line and contouring control systems, First Edition.

International Organization for Standardization (1994), ISO 10303-1, Industrial automation systems and integration - Product data representation and exchange Part 1: Overview and fundamental principles.

International Organization for Standardization (1994), ISO 10303-11, Industrial automation systems and integration - Product data representation and exchange Part 11: the EXPRESS language reference manual.

International Organization for Standardization (1994), ISO 10303-21, Industrial automation systems and integration - Product data representation and exchange Part 21: implementation methods: clear text encoding of the exchange structure.

International Organization for Standardization (2003), ISO 10303-28, Industrial automation systems and integration - Product data representation and exchange Part 28: implementation methods: XML representations of EXPRESS schemas and data.

International Organization for Standardization (2004), ISO 10303-238, Industrial automation systems and integration - Product data representation and exchange Part 238: Application protocol: Application interpreted model for computerized numerical controllers.

International Organization for Standardization (2003), ISO 14649-1, Industrial automation systems and integration – Physical Device Control – Data model for computerized numerical controllers Part 1: Overview and fundamental principles.

International Organization for Standardization (2004). ISO 14649-11, Industrial automation systems and integration – Physical Device Control – Data model for computerized numerical controllers Part 11: Process data for milling.

Newman, S.T., Nassehi, A., Xu, X.W., Rosso, Jr. R.S.U., Wang, L., Yusof, Y., Ali, L., Liu, R., Zheng, L.Y., Kumar, S., Vichare, P. and Dhokia, V. (2008). Strategic advantages of interoperability for global manufacturing using CNC technology. *Robotics and Computer-Integrated Manufacturing*, Vol. 24, No. 6, pp. 699-708.

Solvang, B., Sziebig, G., and Korondi, P. (2008). Robot Programming in Machining Operations. *ch. Robot Manipulators*, Vienna, Austria: I-Tech, pp. 479–496.

Suh, S.H., Lee, B.E., Chung, D.H. and Cheon, S.U. (2003). Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. *Computer-Aided Design*, Vol. 35, No. 12, pp. 1069-1083.

Wu, X.W. (2006). Realisation of STEP-NC enabled machining. *Robotics and Computer-Integrated Manufacturing*, vol. 22, Elsevier, pp. 144–153.

Xu, X. W. and He, Q. (2004). Striving for a total integration of CAD, CAPP, CAM and CNC. *Robotics and Computer-Integrated Manufacturing*, Vol. 20, No. 2, pp. 101-109.

Xu Xun (2007). STEP into Distributed Manufacturing with STEP-NC. *Process Planning and Scheduling for Distributed Manufacturing*, pp. 393-421, Springer London.

Zhang, H., Wang, J., Zhang, G., Gan, Z., Pan, Z., Cui, H. and Zhu, Z. (2005). Machining with flexible manipulator: toward improving robotic machining performance. *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1127-1132.

Zhu, X., Wang, Y. and Fu, H. (2006). A 3-D Simulation System for Milling Machining Based STEP-NC. *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, Dalian, China.