Reconfigurable machine tool programming– a new approach

A.EPUREANU, F.B.MARIN, V.MARINESCU, M.BANU, I.CONSTANTIN
Manufacturing Science and Engineering Department
Dunarea de Jos University
47, Domneasca St., Galati
ROMANIA
alexandru.epureanu@ugal.ro

Abstract: - This paper presents a new programming approach based on a proposed hardware and control architecture for the reconfigurable manufacturing tool machine. A new CAM is described based on virtual machining algorithm that optimizes feedrate scheduling based on geometric simulation and aims to improve the efficiency of cutting process developed using a reconfigurable machine, configured as lathe with a specific architecture. A searching procedure is proposed to find the parameter values corresponding to cutter locations along tool path during generation of the final surface.

Key-Words: - *RMT* programming, feedrate scheduling, reconfigurable machining system, numerical control, optimal control, virtual machining

1 Introduction

Nowadays, companies building manufacturing machines must face fast changes happening on technical, commercial and economical fields. These emerging evolutions are the following:
- economy globalization, with the consequence the emphasis of competition;
- individualization of needs, what means products customization;
- capital dynamization, generating high requirements concerning investment efficiency;
- high versatility of small companies, for fast adaptation to market. At the time, the companies responses to these changes are based on the idea of extending some of the attributes of classic manufacturing system to define reconfigurability. Whereas Dedicated Manufacturing Systems (DMSs) have been designed to produce a specific part and Flexible Manufacturing Systems (FMSs) are designed to accommodate a large variety of parts even though the parts are not specified at the system design stage, the Reconfigurable Manufacturing Systems (RMSs) combines the advantages of DMS and FMS providing an suitable solution[1][2]. It is defined the new peculiar manufacturing system, reconfigurable manufacturing system, using six key characteristics: *i)* universatility, *ii)* convertibility; *iii)* integrability; *iv)* scalability; *v)* diagnosability; *vi)* customisation.These systems are designed to adapt manufacturing to mixed products and volumes. The reconfigurable machine is conceived with standard and inter-operable components including actuated modules, fixtures and tools to be assembled into the machine with arbitrary architecture and degrees of freedom. The informations describing modules and tools are stored in the machine database; for instance, the shape of the tools and the modules geometry.

The reconfigurable “plug-and-play” machine kinematics and dynamic modeling algorithms need to be developed. These algorithms are the basis for the control and simulation of reconfigurable manufacturing system. The concept of machine configuration optimization is introduced for the effective use of the rapidly reconfigurable machine. The issues involved in the reconfiguration process are represented by
- hardware modules reconfiguration
- software reconfigurability (both programming and control)

Examples of physical configurations (hardware reconfiguration) include building a new machine using the hardware modules. So far that concerns software reconfigurability is a must have to develop architectures and techniques to provide short ramp-up time. Control design determines appropriate process variables (modules displacements and velocities etc.), so that a configuration can be operated to fulfill the task. The concepts of the control paradigms, such as holonic manufacturing, bionic manufacturing, fractal companies, are proposed for the next-generation manufacturing systems. So far that concern the control concept, it was proposed [15] a software architecture based on a combination of object-oriented models and executable formal specifications. In this architecture, the machine control software is viewed as an integration of a set of reusable software components, each modeled with a set of event-based external interfaces for functional definitions, a control logic driver for execution of behavioral specifications, and a set of service protocols for platform adaptation. The behaviors of the entire software can be viewed as an integration of behaviors of components and their integration.

The needs for interoperation of modules from different vendors and for flexible configuration of various functions are challenges to NC systems. In response to such challenges, various efforts have been made, such as Open Modular Architecture Controllers (OMAC) [11], Open System Architecture for Controls within Automation Systems (OSACA) in Europe [12], Open System Environment for Controllers (OSEC), Hierarchical Open Architecture Multi-processor for CNC (HOAM-CNC), Open Architecture Controller (UMOAC) by University of Michigan, Ann Arbor, USA etc. The common approach employed in these efforts is the adoption of an open, modular architecture to promote interoperation [13].

The nowadays CNC control system is inappropriate to consider as basis for the control of RMTs. Existing CNC systems support only CNC builder-specific NC program input and this limits the potential application of many NC programs that have the same functions with different formats varying one machine to another[6][7][8]. In the case of the RMT, where there is a wide range of configurations, using nowadays programming standards will be time consuming. Because of the issues of software reconfiguration the ramp-up time is considerable, as

However, so far that concerns programming of RMS there is no reference.

In nowadays CNC systems, the NC program processor is a very important component that determines the accurate resolving of machining intention generated from a CAM system [1]. The major functions of CNC processor include checking the syntax and decoding them into specific outputs such as motion command, PLC command, parameter setting, or error messages.

There have been three basic standards: RS274D (USA), ISO6983 (ISO) and DIN66025 (Europe) for NC program since CNC was invented. However, new RMT machine tool and control technology have undergone great development since then. Even for today's CNC there are a lot of new functions needed and controller-specific features not supported by these NC standards.

Also there is the issue of coding free-form surfaces using nowadays CNC programming languages, as these surfaces are used in a variety of applications. The conventional approach for surface machining is to use a series of straight lines to approximate the part surface. These straight lines are subsequently translated into linear G codes by the computed-aided machining (CAM) software, and then sent to the computer numerical control (CNC) system. For a general free-form surface or parametric surface machining, this method is inefficient and error-prone. The conversion of surfaces into linear segments produces a large amount of data, which usually cause a lot of problems, such as lower feedrate, and longer data storage requirements [14].

Also, feedrate optimization problems are the subject to several possible constraints, building the framework for generating model. Firstly, the common restriction used in feedrate scheduling is constant material removal rate (MRR), [3]. In the MRR based approach, feedrate is dependent to either average or instantaneous volumetric removal rate. Secondly, the restriction is based on the process mechanics [4]. In this case the force-based models, feedrate is set to values which keep either average or instantaneous machining forces to prescribed values. These models computes either MRR or force values as a minimum allowable values, and are maintained at a constant level during processing.

The disadvantages of keeping at allowable level are in the first place, the efficiency is lowered and secondly, the optimization is incomplete performed, as it does not take into account several variables in the process such as chip width (determining risk of chatter), roughness or thickness (determining force overloading).

Commonly used CAM programs and NC code generators are based on geometric and volumetric analysis, but they do not concern the current blank surface shape to be processed. CAM software considers only the nominal dimensions inputted from CAD model [5] [6] [7].

Also planning the isoparametric tool path it is addressed to considerable long paths (due to the G-code limitation and the nowadays machine control system architecture) and not for adjacent cutter location resolution.

It is critical, but often difficult, to select optimal cutting conditions to achieve high productivity while maintaining high quality of parts.

Generating the optimal cutting conditions scheduling, described by NC programs, to produce the required geometry involves determination of the optimal cutter paths and machining parameters. The common practice is to set machining parameters such as feedrate, to constant value for the number of cutter paths, using machinability handbooks and experience of skilled machinist. The selected parameters are often so excessive limited that efficiency is very low for the machining process.

In this paper it is proposed a new programming approach based on a proposed hardware and control architecture for the reconfigurable manufacturing tool machine. A new CAM is described based on virtual machining algorithm that optimizes feedrate scheduling based on geometric simulation and aims to improve the efficiency of cutting process developed using a reconfigurable machine, configured as lathe with a specific architecture. Also, a searching algorithm is proposed to find the parameter incremental values of all cutter locations along tool path for the generation of final surface. The proposed algorithm includes the following two main steps: evaluating blank real surface shape and searching for adjacent non-isoparametric cutter locations, in the conditions to meet the the restrictions.

The algorithm shall be embedded in a virtual machining system that optimizes machining process, based on geometric simulation, to improve the process efficiency, under the assumption of reconfigurable machines architecture.

Besides the efficient cutting condition scheduling goal, there is the advantage by using such virtual machining system, represented by the optimized choosing of the tool for a specific operation and a specific blank. The example given is taking into account a reconfigurable machine configured as a lathe. However, the algorithm may be extended to all cutting processing.

Paper has the following structure: section 2 present the problem formulation, section 3 contains the problem solution and section 4 summarizes the main conclusions achieved.

2 Problem Formulation

2.1 Variables and terms definition

We formulate the problem of optimizing the values of cutting processing to improve its efficiency. We consider to be addressed to a RMT configured as lathe with a specific architecture.

In order to define the problem, the issues involved and determine a solution, we need to define several variables according to our approach.

As shown in Fig.1, the insert cutting edge profile is divided defining remarkable (m) points; when we addressing to a current remarkable point on the cutting edge as *j* point. Similarly as depicted in Fig.2 the workpiece profile is divided in remarkable n points and we referring to a current remarkable point on the workpiece profile as *i* point.

Current cutter location is described by *(i,j)* pair; while *i*, is defining the current cutter location, *j* is current cutter profile. Another variable in our optimization program is the variable k, which is the indexing variable for *(i,j)* pair.

As described below, the reconfigurable machine is configured as lathe with an additional degree of freedom represented by rotation of the tool assembly, with an φ angle, variable that determines j cutter profile.

Our search algorithm is using as reference calculation the previous cutter location *(k-1)*, as in Fig.2, representing the last selected cutter location. Consequently, every *k* position is associated with the φ*(k)*, Z*(k)* and X*(k)* variable values.

In the proposed search optimization algorithm we use current L variable value defining the length {P,Q} which is the length of active cutting edge, as seen in Fig.2.

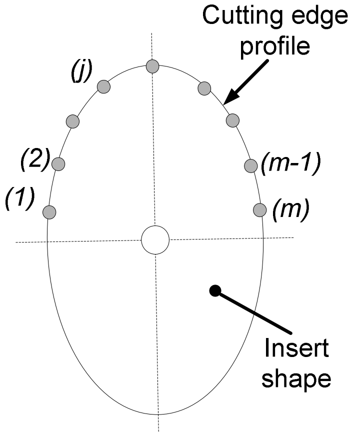


Fig.1 Insert

According to our approach there are several data to be used as input data:

Firstly, workpiece profile and cutting edge profile is inputted from CAD model, as we assume that all tool modules are described in the database stored in the reconfigurable machine information system.

Secondly, the blank profile is determined by on-machine measurement. Thirdly, it is inputted the restricted variables: allowable force -Fa, allowable L –La (the length of cutting active edge as shown in Fig.2 represented by PQ length), allowable chip thickness - aa (chip thickness), allowable roughness –Ra.

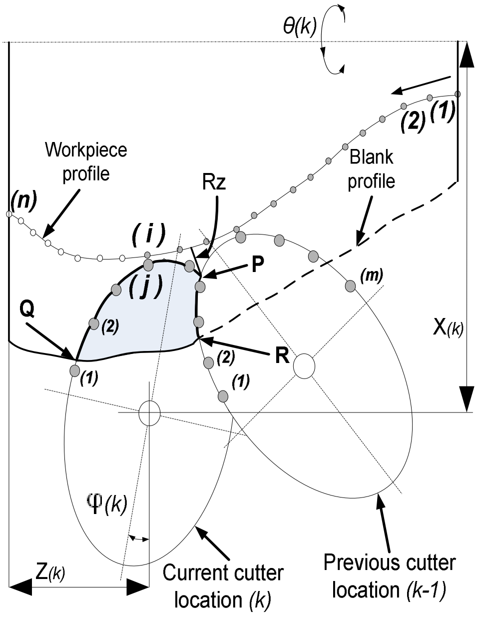


Fig.2 Current and previous cutter locations

2.2 Reconfigurable lathe description

For a better understanding, the reconfigurable lathe hardware configuration and the control system is described bellow.

Our proposed architecture for the reconfigurable lathe is composed of virtual processing planning system, machining database, and "plug-and-play" hardware modules.

The reconfigurable lathe considered is build out of four independent modules, numerical controlled, with independent sensor and motors. These modules are main spindle, carriage, and the rotary tool assembly. The reconfigurable lathe, configured for cutting of longitudinal profiles, as shown in the picture, is made of the following parts: bed 1, main spindle 2, workpiece 3, cutting tool 4, tool holder 5, carriage 6, rotary tool assembly 7, cross-slide 8, slideway 9. Having a supplementary degree of freedom represented by rotation of cutting tool it is an advantage for tool to positioning, and using interpolation for x and z axis and the rotation of the tool it can be achieved peculiar surface processing. Using a configuration with rotary assembly tool parallel with the axel of the spindle, it can be processed surfaces such as poliexcentric revolution surfaces, complex surfaces or cams. The hardware modules (main spindle, tool holder, etc) are design as "plug and play" and are controlled directly using optimal control variable, by-passing the post processing phase of G-code generation. The control of the modules it is performed by sending simultaneously successive position of controlled variables X*(k)*, Z*(k)*, θ*(k)* and φ*(k)* computed from *(i,j)* pair.

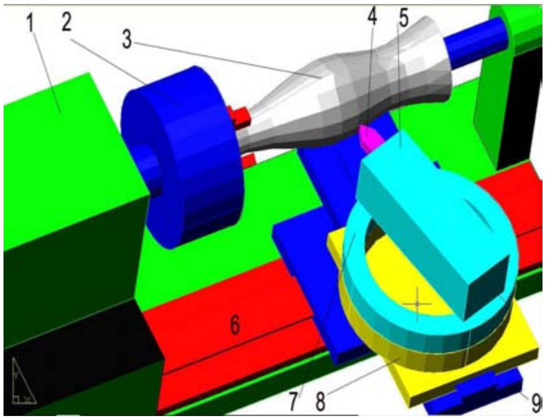


Fig.3 The reconfigurable lathe

Part surface generation it is performed by moving tool profile along helicoidally generating path, obtained by combining revolution of the part and the translation of the tool. The trajectory of the helicoidally path it is divided, as described above, by coordinates computing of a many successive *(i,j)* pair points to control motors that performs movement actions of the machine.

2.3 Problem statement

Our objective is to determine *(i,j)* pair under the criteria objective of maximizing locally chip area.

While the search criteria in the MRR and force model are the MRR and the force respectively, in our approach the search criterion is the locally maximized chip area.

In the above two approaches the variable for controlling the process is the feedrate, whereas we are using two distinct variables, cutter location CL (or federate) and cutter profile (CP). The cutter profile variable is the additional freedom degree determined by rotary tool assembly.

Concerning the restrictions, in the case of MRR model, is the chip area which is set to a minimum allowable (A<Aa), and in the case of force model, the restriction is the force (F<Fa). Our approach defines four restrictions: (Aa>A*(i,j)*, Ra>R*(i,j)*, aa>a*(i,j)*, La>L*(i,j)*.

The search technique used in our approach is exhaustive search by virtual machining.

Online control such as used in force model, need monitoring hardware; the proposed optimizing method needs a measurement system, which is performing every blank exploring.

3 Problem Solution

In order to program the RMT under the above stated conditions we need to state several key ideas to define our proposed RMT control:

-adaptive
- online
- optimal
- self-programming

The part-program should comprise information regarding the characteristics imposed on the manufactured object in order to be considered acceptable, and not information on the way in which the functioning cycle of the machine should develop so that the object may have these characteristics – This means that the part-program should contain a list of targets and not a list of instructions. Starting from the targets included in the part-program, its structure of the machine should allow for negotiation of the plan design, plan analysis and plan proper execution. The part-program shall be very simple as it will contain only the targets written in a high-level language. Consequently the machine programming time shall be much shorter, which is crucial in case of small batch production.

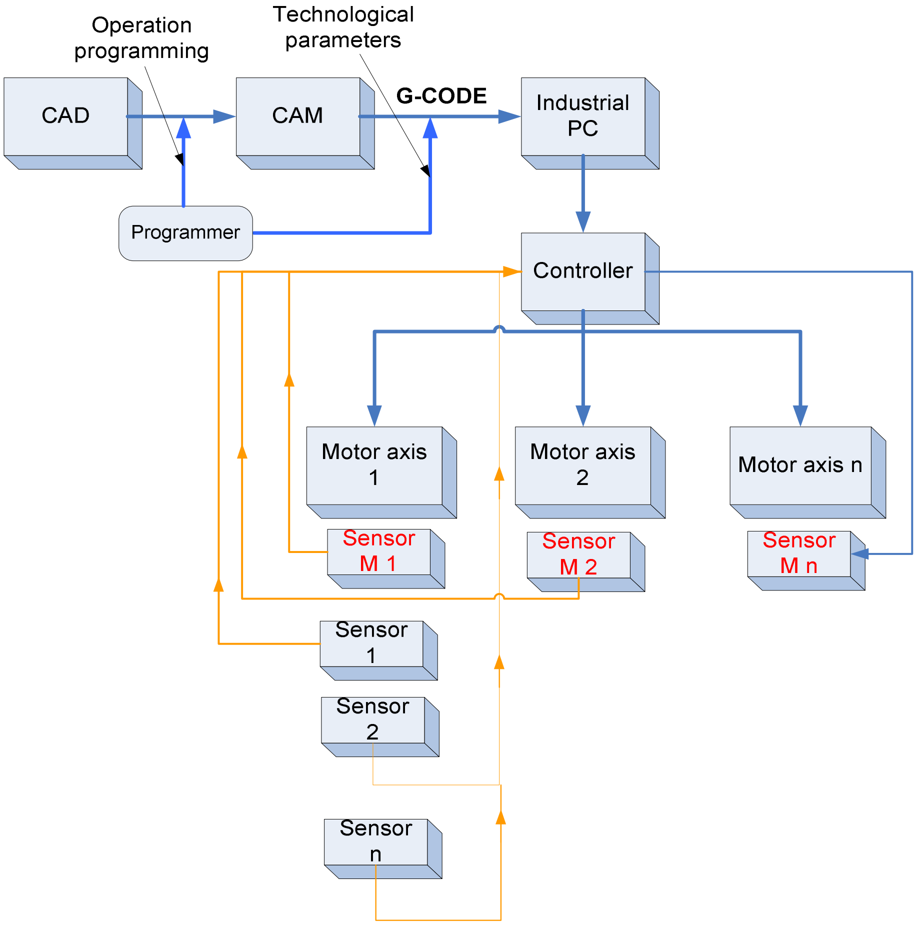


Fig.4 CNC control hardware architecture

For instance in Fig.4 it is described nowadays CNC conceptual control architecture. Traditionally, CAM has been considered as an NC programming tool used to generate CNC code to drive numerical controlled machine tools. Most of current numerical control (NC) systems adopt an architecture as described in Fig.4 A computer system, which is usually a general-purpose personal computer (PC) or another kind of a computer, is chosen as the NC core system platform, where human machine interaction or interface (HMI), NC code editing and decoding functions are implemented. Functions such as interpolation and control, cutter radius compensation, general I/O (or logic control), servo control, etc., are implemented in several auxiliary controllers (such as PLCs – Programmable Logic

Controller). Consequently, the controllers are managing commands to servo drives to control motors and read data from sensors.

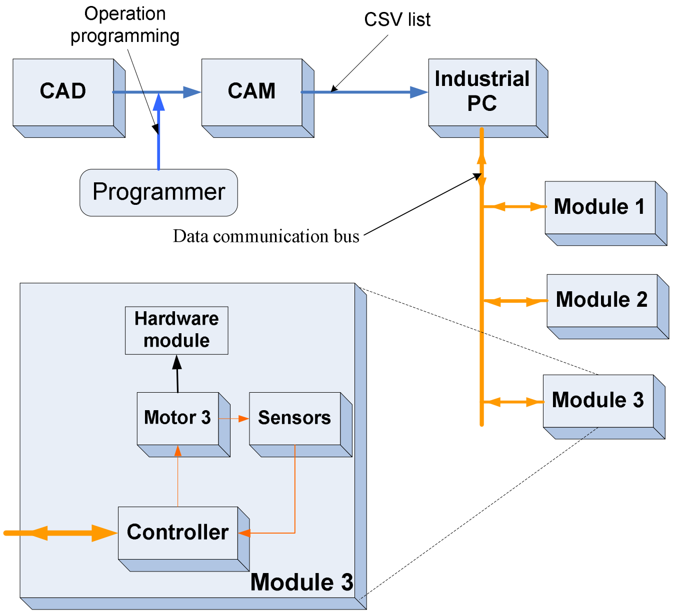


Fig. 5 Proposed RMT control hardware architecture

Our approach considers a bus able to handle communication for all modules. Compared to today's fieldbus architecture the hardware modules have close-loops control represented by an integrated controller able to communicate with industrial computer. The information send to hardware module is the required position only. Also, if a hardware module has specific sensors besides position control of its movement, these are to be handled by the integrated controller on the hardware module. Compared to nowadays CNC architecture, CAM is not providing source code to be interpreted and executed by machine, but an algorithm calculated with nominal values. Whereas modern CNC CAM software outputs source code describing geometric path, and none or few information concerning technological parameters(such as federate), our approach is providing the advantage of providing the values for technological description of the process. Therefore our CAM is adaptive and consequently we name it Adaptive-CAM (A-CAM).The algorithm computed in the A-CAM it is also computed in Algorithm running unit (ARU). The CAM running in process design phase is virtual machining considering nominal values. Consequently, the ARU running on the machine is running with measured values, which are continuously updated and computed. As shown in Fig.5 and Fig.6 it is inputted in the A-CAM the following information: CAD models of required part, CAD model of the rough part, technical requirements of the finished part (such as tolerance range and roughness), operation description, and rough part material description (such as material hardness).

As depicted in Fig.7 modern CNC machine programming architecture is considering the following data flow. A CAD program is used for workpiece design. Based on the workpiece specification process planning is then performed using a CAPP (Computer Aided Process Planning) software package. Then, using a manufacturing process plan as input the part program is generated by CAM software. The CNC process this program and the result is the finished workpiece. No correction is performed during the machine running. In our proposed architecture Online Learning System is providing the parameter correction computed during process. This is not subject to the current paper. Also the stability of the process is not taken in account. Only the stability of each axis is assured during the machine setup by setting choosing a set of parameters.

The basic idea which is based our approach is that the programmer is inputting "the required part", "part to be processed" and "the requirements of finished part". The A-CAM distinctive software modules, is running in the design phase (A-CAM) considering nominal parameters (Fig.7). A CAE program is used to test current kinematical configuration. However this is not subject of our paper. ARU, which is basically an extension of A-CAM, is outputting Continuous State Variable List data set representing the offset moving for each module. According to us CSV data list is the part-program which is generated online and is taking into account the current part to be processed.

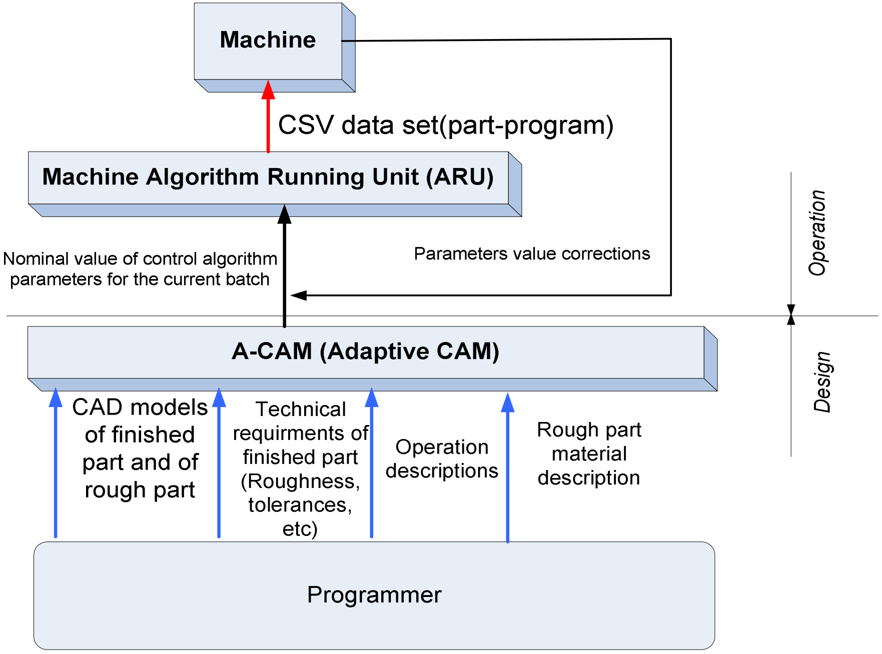


Fig. 6 A-CAM proposed architecture

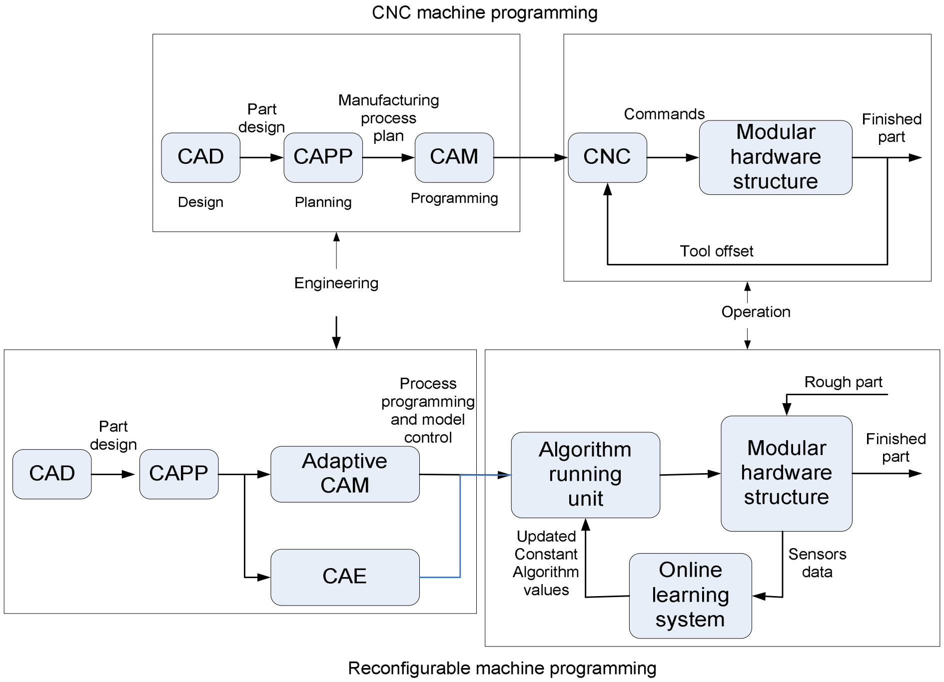


Fig.7 Proposed RMT programming

The programming and control architecture proposed is allowing commanding to the hardware modules optimized trajectory.

The main idea of the optimization algorithm is that, starting with the current *i* point on the profile part generated by *j* point on the tool, it is searched the next *i* cutter location and the next cutter profile *j* coordinate point, taking into account the restrictions defined by restricted variables (A*ij*<Aa, L*ij*<La, a*ij*<aa, R*ij*<R, with the maximum chip area, in order to maximize productivity.

For generation of certain *i* point on the workpiece profile, the tool must be positioned tangent to this profile in a *j* point.

For this current cutter position, it is computed the values of controlled variables A*ij*, L*ij*, a*ij*, R*ij*. Afterwards it is tested restrictions conditions A*ij*<Aa, L*ij*<La, a*ij*<aa, R*ij*<R and also of it is satisfied restriction that *j* point is on the active cutting edge. The results of the testing are consequently stored.

Fig.8 depictures the generation sequence for final surface, using optimization algorithm. For instance, the point *i*=21 on the workpiece profile and point *j*=16 on the cutting profile describes the starting point for the searching of next point for an 360 revolution (pair (21,16) is the previous cutter location).

To achieve this, it is tested one at a time tool position for *i*=22 and the points that verify restrictions are stored. The searching it is finalized when it is found a point in the final surface profile, with all points on the tool profile, tangent, that do not verifies conditions. Points *i*=25 and *j*=15 is the pair points for which chip area is maximized for the given points. It is considered that the new *i* point, in this case, *i*=25 is the correct cutter location and the search algorithm is continued searching the next point referred to the last *(i,j)* pair location.

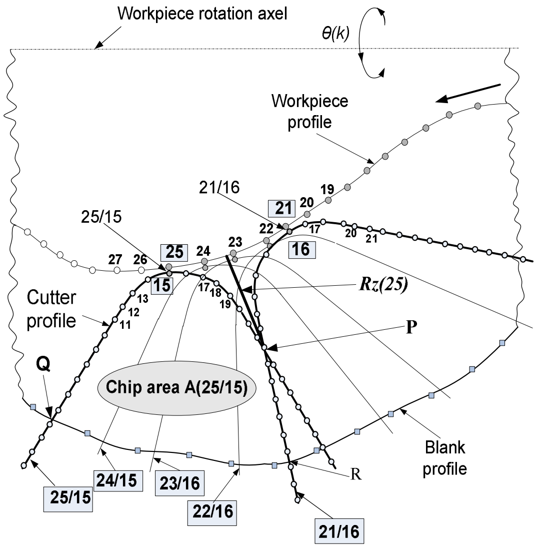


Fig.8 An example for search loop

Trajectory from point *i*=21, *j*=16, to point *i*=25, *j*=15, it is described by the points *i*=22,23,24, dividing the propeller resulted from one revolution, in four intermediary steps, with coordinate X, Z and $\Delta\theta$. All intermediary *(i,j)* pair coordinates are stored with their restricted variables.

In sort, as shown in Fig.8, the proposed algorithm contains the following steps:

Preparatory steps

1) workpiece and tool CAD model are inputted in the optimizing system;
2) the blank profile is measured on the machine;
3) the restricted variables values are inputted in the virtual manufacturing program

Scheduling optimizing steps

4) the tool profile it is positioned tangent with the final profile, successively in all points on the tool.
5) it is computed the restriction conditions for all positions and its characteristics,
3) when all conditions are denied, then it is selected an intermediary position with maximum area,
4) the algorithm it is repeated starting with 1) until the tool reaches the end of the profile.
5) the software output are the *(i,j)* pair coordinates
6) processing cinematic configuration of the lathe by intermediary of the *(i,j)* pair coordinates there is obtained the controlled variables values - X*(k)*, Z*(k)*, $\theta(k)$ and $\varphi(k)$ - to be send to the modules by ARU.

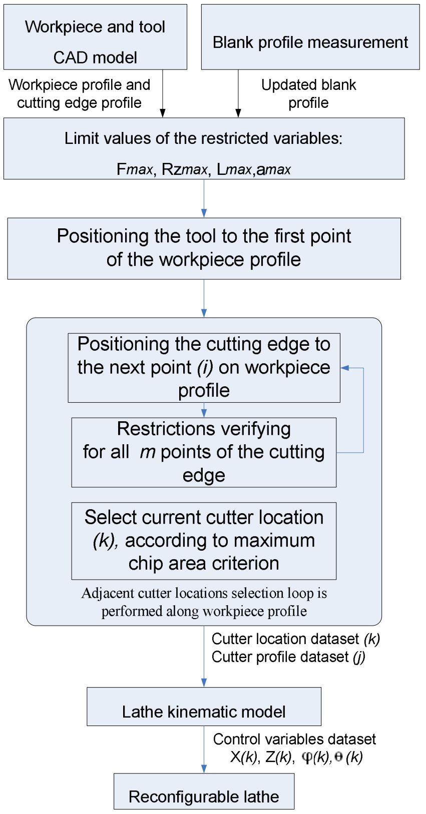


Fig.9 Algorithm basic steps

Algorithm testing was performed using the software developed in this research.
It is important to note that the experimental tests shown in the Fig.10 was executed using arbitrary scale and on few sequences, for an acceptable picture accuracy.

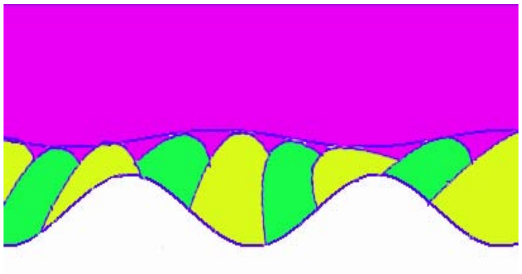
The histogram is given bellow to demonstrate the performances of the algorithm. As shown in Fig.11, a) image is representing the successive position of the cutting edge using our algorithm and lathe architecture, whereas in b) image it was simulated constant MRR based approach with classical lathe architecture.

The yellow and green area depicts the cutting tool successive position.

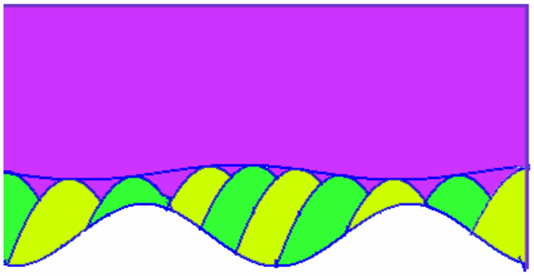
It was noticed an average efficiency improvement up to 20%, resulting a higher cutting productivity. Compared with the classical case, the additional freedom degree represented by rotary tool assembly is the cause for the increased productivity. Secondly, the computation of all four restrictions, instead of one or two, assures that the process is correctly performed according to the specifications.

As seen in Fig.7, when the process is based on MRR approach, and there is no limitation concerning the length of the edge, it is possible to emerge a situation could cause instable process because of the vibration.

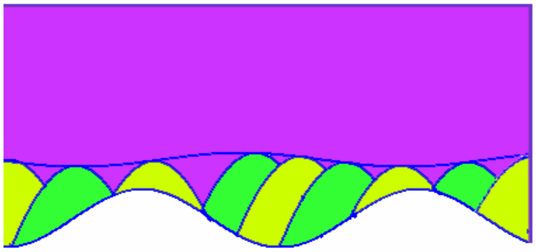
Furthermore, there is the possibility to undermine the roughness or chip width allowable values.



a) Proposed algorithm sequences -F,L,a,R restricted



b) MRR based algorithm sequences
F and R restricted

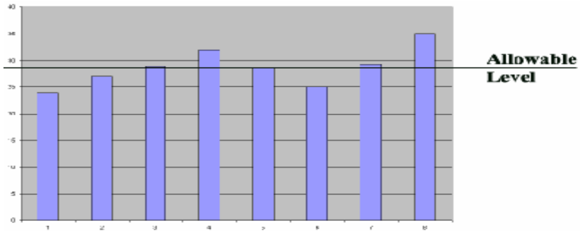


c) MRR based algorithm
F only restricted

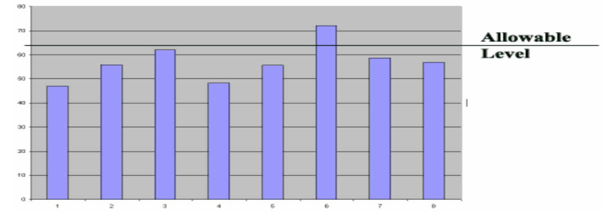
Fig.10 Graphical representation of the cutter location and cutter profile during virtual machining

In the Fig.7 it is shown a histogram representing the situation seen in Fig.6, c showing the roughness, chip width and cutting edge length variations for each cutter location, to evaluate the necessities of more than one or two restrictions.

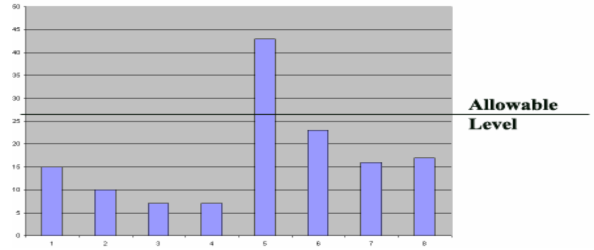
Since a and R values are greater than the allowable value for the given example, see Fig.7 a) and b) respectively, all four variables are required for the establishment of feedrate scheduling. It can be stated that is compulsory requirement for using more than one restriction, in terms of stability process and control.



a) Chip width variation



b) Cutting edge length



c) Roughness variation

Fig.11 Histograms showing undermining allowable levels

4 Conclusion

Compared with the classical case, the additional degree of freedom represented by rotary tool assembly, is determining a 20% productivity increase.

The computation of all four restrictions instead of one or two, assure that the process is correctly performed according to the specifications, and the processing parameters are in the allowable range. The proposed control concept proofing was aimed by virtual simulation by processing cinematic configuration of the lathe by intermediary of controlled variables values ($X(k)$, $Z(k)$, $\theta(k)$ and $\varphi(k)$) to be send to the modules during optimized virtual processing. We conclude the new A-CAM proposed is a viable option for RMT programming and it can be implemented physically.

Several topics remain to be studied in the future:

-firstly, methodologies to rapidly refine the control reconfiguration describing current machine architecture in order to accurately identify hardware modules.

-secondly, it is challenge to design controller embedded in hardware modules to meet RMT requirements.

Acknowledgement

The authors gratefully acknowledge the financial support of the Romanian Ministry of Education and Research through grant PN-II-ID_653/2007.

References:

[1] R. Galan, J. Racero, I. Eguia, J.M. Garcia, A systematic approach for product families formation in Reconfigurable Manufacturing Systems, *Robotics and Computer-Integrated Manufacturing 23*, 2007, 489–502

[2] Jaspreet Dhupia, Bartosz Powalka, Reuven Katz, A. Galip Ulsoy, Dynamics of the arch-type reconfigurable machine tool, *International Journal of Machine Tools & Manufacture 47*, 2007, 326–334

[3] X1. Hwan Pyo Moon, Equivolumetric offsets for 2D machining with constant material removal rate, *Computer Aided Geometric Design nternational 2007 Elsevier B.V ,* Vol.X, No.X, 2007

[4]Jeong Hoon Ko, Won Soo Yun, Dong-Woo Cho, Off-line feed rate scheduling using virtual CNC based on an evaluation of cutting performance, *Computer-Aided Design 33*, 2003, 383-393

[5]Stanislav Makhanov, Optimization and correction of the tool path of the five-axis milling machine Part 1. Spatial optimization, *Mathematics and Computers in Simulation 75*, 2007, 210-230

[6]Stanislav Makhanov, Optimization and correction of the tool path of the five-axis milling machine Part 2: Rotations and setup, *Mathematics and Computers in Simulation 75,* 2007, 231-250

[7] B.S. Soa, Y.H. Jung b,, T.R. Kurfess c, S.M. Hwang, 5-Axis machining speed enhancement by step length optimization, *Journal of Materials Processing Technology 187–188 75,* 2007, 2-5

[8] M. Gallo, G.Guizzi, V. Zoppoli, A methodological approach to develop an integrated simulation system in manufacturing processes, *6th WSEAS International Conference on System and Simulation and Engineering*, 2007

[9] M. Paolucci, R. Revetria, F. Tonelli, An Agent-based System for Sales and Operations Planning in Manufacturing Supply Chains, *6th WSEAS International Conference on System and Simulation and Engineering*, 2007

[10] M. Yurtseven, Design and performance analysis of a linear quadratic Gaussian controller in a manufacturing process, *6th WSEAS International Conference on Circuit, System, Electronic, Control and Signal Processing*, 2007

[11] OMAC API Work Group (1999) OMAC API set, Version 0.23. Work-ing document. http://www.isd.mel.nist.gov/projects/omacapi/

[12] OSACA (1996) Open system architecture controls within automation systems. Final report. http://www.osaca.org/

[13] Yi-Hong Long & Zu-De Zhou & Quan Liu &Ben-Yuan Chen & Heng-Lin Zhou, Embedded-based modular NC systems, *Int J Adv Manuf Technol,* 2008

[14] Wei Li & Yadong Liu & Kazuo Yamazaki & Makoto Fujisima & Masahiko Mori, The design of a NURBS pre-interpolator for five-axis machining, *Int J Adv Manuf Technol,* 2008

[15] Young KW, Piggin R, Rachitrangsan P (2001) An object-oriented approach to an agile manufacturing control system design. *Int J Adv Manufact Technol*,2006