БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ • BULGARIAN ACADEMY OF SCIENCES

ПРОБЛЕМИ НА ТЕХНИЧЕСКАТА КИБЕРНЕТИКА И РОБОТИКАТА, **59**
PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS, **59**

София • 2008 • Sofia

Trends in Service Oriented Architectures for Robot-CNC Manufacturing Control

Theodor Borangiu

University Politehnica of Bucharest, Faculty of Automatic Control and Computers,
313, Spl. Independentei, Sector 6, RO-060032 Bucharest, Romania
E-mail: borangiu@cimr.pub.ro

1. Introduction

Manufacturing control systems have dynamically evolved during the last years; one of the most significant facts is the emergence of decentralized systems capable of better dealing with the rapid changes in the production environment than the traditional hierarchical, centralized ones. The quest for agility and re-configurability requires a new class of production control systems, characterized by: (i) a community of distributed, autonomous and intelligent building blocks, designated as control units; (ii) autonomy of each control unit which has its own objectives, knowledge and skills; (iii) global decisions are obtained by the cooperation of more than one control unit; (iv) control units adapt to changes without external intervention; (v) control units of mechatronic devices such as machine tools, robots, conveyors and vision sensors, are part of reconfigurable, fault-tolerant planning and control architecture of global manufacturing system; (vi) input data and formats for part processing are subject to increased diversity, requiring both high-speed cooperation between external sensors (range finder, vision, dexterous robots and CNC multi-axis machine tools, and easy reconfigurable, task-driven software systems capable to detect, recognize, locate, qualify, inspect complex objects and accurately represent in digital form their 3D surfaces/shapes. This allows a fast set up of production in response to customer orders (*make-to-order* process) ranging from one complex part to large batches.

The extension of traditional material-handling control to *material-conditioning* functions (detecting, recognizing, qualifying, locating, inspecting, manipulating) allows the relaxation of material flow and process-constraints imposed to the transportation means, as well as:

- Rapid object surface scanning, accurate generation of its 3D digital representation, and further processing producing a corresponding $2^{1/2}$ description to be used in CNC machining.
- On line integration of quality control in the global manufacturing system.

This functional extensions became possible in external sensor (vision, range)-robot-CNC machine architectures by taking over: (i) the powerful description of material flows through an efficient set of multiple form-, surface-, and pose features of circulating parts, based on *real-time, high-speed image processing,* (ii) the scanning of complex shaped objects by help of anthropomorphic robots moving laser scanning devices along configurable motion patterns, and by applying *Artificial Intelligence* concepts to reach a global autonomous, self-learning, task- and context-dependent behaviour of the robot tended machining device (CNC machine tool) with adaptation to the working environment.

Thus, industrial robots equipped with range finder devices and machine vision software represent the smart control components of future intelligent, autonomous transportation and manufacturing systems [1].

As for the topology of manufacturing control, hierarchical systems are characterized by a strong master-slave relationship, whereas *heterarchical systems* provide an attractive alternative with a simpler management structure in which there is a bias towards less monitoring and control and more accountability and responsibility [2]. Heterarchical control structures have been usually investigated using the concepts offered through *agent-based technology* [3, 11].

Within the majority of the contemporary distributed manufacturing applications there is a trend towards adoption of a *semi-heterarchical structure* in which organizational control is arranged into two levels, often referred to as global and local [2, 10]. The global level assumes the responsibility for coordination of cell/line/system/factory level activities and the resolution of conflicts between local objectives, whereas the local level possesses the autonomy over the planning and control of internal activities within a subsystem (e.g. a *team*). The *team-based manufacturing* paradigm has been adopted to provide the flexibility, agility and responsiveness required to cope with the volatility of production demands [11], see Fig. 1. The process of grouping manufacturing resources to form teams is based on a wide range of criteria, such as similarity of activities, definition of business processes, and production planning and control requirements [12].

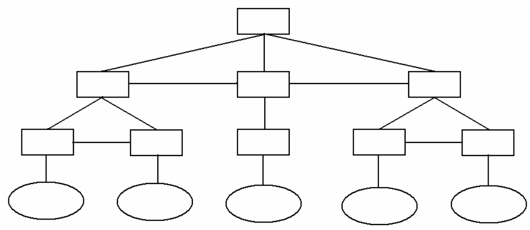


Fig. 1. Team-based semi-heterarchical manufacturing planning and control structure

To compensate for the deficiencies of both hierarchical and heterarchical control systems, in recent years the research community introduced several new concepts for the design of manufacturing systems such as *Fractal Factory, Bionic Manufacturing,* and *Holonic Manufacturing Systems* [3]. Each concept attempts to model a manufacturing system based on some analogies with other existing theoretical, natural or social organization systems.

The agent-based and holonic paradigms symbolize these new approaches and PROSA [6], ADACOR [7], and Bussmann [5] are successful examples of a system. They deal with the re-configurability in manufacturing systems by introducing an adaptive production control system that evolves dynamically between a more hierarchical and a more heterarchical control architecture, based in self-organization and learning capabilities embedded in individual holons. Most of the studies concerning holonic manufacturing systems [10] focus on factory architecture and/or how to assign a production task to each manufacturing holon. Workable Holonic Manufacturing Execution Systems (HMES) design principles are exposed by Cheng in [8] and Babiceanu in [9].

A current challenge in production control is to approach multi-agent systems with new emergent technologies, such as Service-Oriented Architectures (SOA) which support the development of more powerful reconfiguring and interoperability mechanisms, using more complex self-organization and learning techniques [1, 3, 9]. Service composition in multi-agent manufacturing systems (MAS) is the combination of single services and all the interaction patterns between them. The re-configurability and evolution of a production structure is facilitated using multi-agent systems supported by *web services technology* since it is possible to add, remove and modify dynamically resources and services without interrupting the processes. The development of orchestration and choreography mechanisms and tools, including orchestration engines for composition, coordination and collaboration, will play a crucial role to support intelligent, reconfigurable and modular manufacturing control systems [5].

2. Team-based manufacturing integrated in Holonic Manufacturing Execution Systems

Three main areas of activities are considered when defining the team-based planning and control within manufacturing structures (Fig. 2):

- *pre-production activities*: business-driven design and engineering (CAD/CAE), reverse engineering, and process- and production planning (CAPP);

- *production activities* including the range of material conditioning tasks: feeding, machining, robotized manipulating, assembling, grinding, polishing, painting – all of them carried out at shop-floor level (CAM);

- *post-production activities* such as sorting, measuring, quality inspection (CAQC), packaging and marking – carried out before delivery.

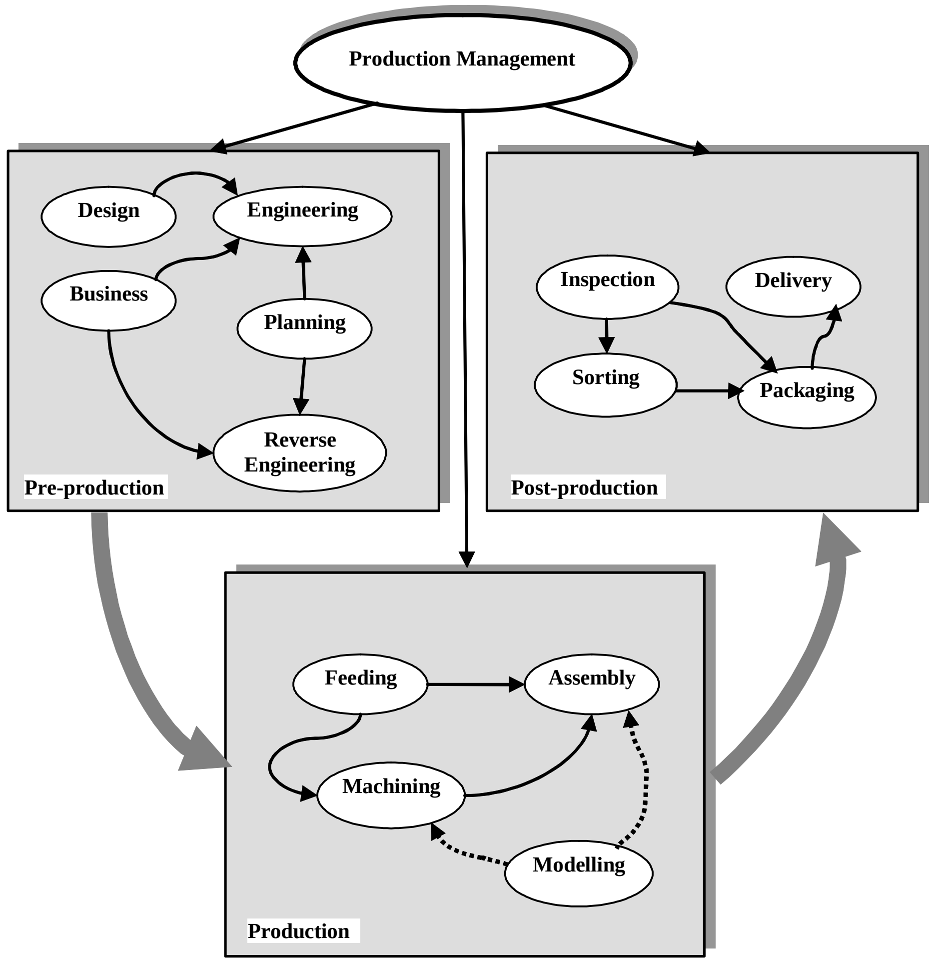


Fig. 2. Activity grouping in team-based manufacturing planning and control

In the proposed approach, material conditioning intensively uses feature description of materials, components, parts and assemblies; this description is provided by machine vision systems taking and processing pictures of robot workstations, machine tool workplaces, part storages, and conveyor belt windows [1].

Consequently, extending traditional material-handling control to *material-conditioning* functions, allows the relaxation of material flow and process constraints imposed to the transportation means, as well as on line integration of quality control in the global manufacturing system. This functional extension became possible in a **robot-vision architecture** by taking over: (i) the powerful description of material flows through an efficient set of form-, surface-, and pose features of circulating parts, based on *real-time, high-speed image processing* and (ii) applying *Artificial Intelligence* concepts to reach a global autonomous, self-learning, task- and context-dependent behaviour of the robotized manufacturing system with adaptation to the working environment. Robot Vision is in this approach the smart control component of intelligent, autonomous production and transportation systems.

Original solutions based on intelligent image processing are proposed for all three production stages represented in Fig. 2; they provide a unique combination that simultaneously offers a maximum of performance and efficiency, by a new approach of manufacturing automation defined as Rapid Deployment Automation (RDA), which considers the design of Robot Vision (RV) and Automated Visual Inspection (AVI) systems as modular development processes. Instead of dedicating strongly personalized systems to complex measuring, inspection or guidance tasks, each RDA component – video camera, range finder, vision library, manipulator, robot controller, machine tool, CNC unit, conveyor, and even utility, development and debug software – is conceived as standard part perfectly adaptable to the puzzle of any flexible manufacturing task. In this approach, the need of redesign and reconstructing of a complete system which must quickly respond to new functionalities is eliminated; it suffices to remove, add, or update individual RDA components according to current requirements.

The objective of the reported research consists in developing a **multifunctional, autonomous, self-adaptive manufacturing platform** capable to perform all three team-based activities, for single-product orders to large batch production:

1. Pre-production activities:

- Scan complex object surfaces to provide their accurate digital 3D description from depth map images; compute iso parametric machining toolpaths by interpolating between image points of equal grey levels; generate G-code for CNC machine tools (*planning, reverse engineering*).
- Take pictures of part models; extract part contours and surface details from grey level camera images; transpose feature data into toolpaths; generate G-code for CNC (*reverse engineering*).
- Directly download CAD files describing part geometry; post process CAD data to generate CNC toolpaths (*design, engineering*).
- Prioritize activities according to rush orders; change class model description or toolpaths spec according to customer requests (*business*).

2. Production activities:

- CNC machining (*roughing, finishing*).
- Robotized mounting, grinding, polishing, cutting (*assembling, material conditioning*)

3. Post-production activities:

- Automated visual measuring, defect detection (*inspection, quality control*).
- Feature-based part qualifying, classifying, palletizing (*sorting, packaging*).

This versatile, multi-team based manufacturing platform was designed to integrate a short range, high-precision 3D laser scanning probe displaced by a

6-d.o.f. vertical articulated robot manipulator relative to a model (object) to be reproduced on a 5-d.o.f CNC milling machine, with automatic division of the material to be removed and adaptation of the feedrate and spindle speed during the roughing stage of part machining. The laser probe is able to measure distances from 70 to 250 mm, with an accuracy of achieving 30 µm. The robotic arm moves around the work piece – eventually placed on a rotary table with closed-loop position control – being scanned by computer-generated adaptive scanning paths (Fig. 3).

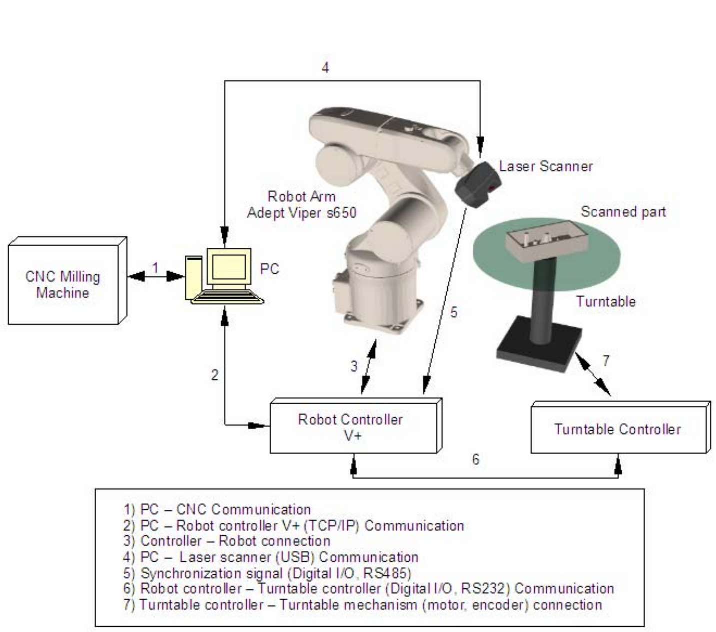


Fig. 3. Hardware architecture of the laser scanner-robot-CNC machine platform for *on-demand* reverse engineering tasks

These scanning paths are computed in real-time by the robot controller from a predefined motion pattern, while the range finder device generates depth map-type information describing the object's surface, synchronously with the motion of the laser scanner probe. The scanned 3D models will be then reproduced on a CNC milling machine for any ordered batch size.

In addition to this *reverse engineering* capability, the multifunctional manufacturing platform can process parts from already available CAD data (*design, engineering*), perform *assembly* operations under Visual Guidance of the Robot manipulator (GVR), or Automatic Visual *Inspection* (AVI) of fed components, machined parts or assemblies, by using up to four video cameras (three stationary, mounted along Cartesian directions, and one mobile, arm-mounted).

The scanning device is a class-2, short distance, triangulation one, and has two CMOS sensors allowing the scanning of complex object surfaces. The optimal scanning distances range from 71 up to 242 mm. The width of the scanning line varies between 31 and 83 mm, and the average measuring precision at point level is 31 µm.

The acquisition rate is between 50 and 150 frames per second, the number of points which are read on a scanning line being 480. The laser range finder system is interfaced to a 3.2 GHz IBM PC-type station by means of a standard USB input port, and uses additionally a digital RS485 line for synchronization with the robot controller.

The on demand reverse engineering platform uses a vertical articulated, 6-d.o.f. Viper 650 Adept robot with a repeatability of 0.02 mm. The displacement domains of the six rotational joints are respectively: joint 1: ± 170°, joint 2: –170°,+45°, joint 3: –29°, +256°, joint 4: ±190°, joint 5: ±120°, joint 6: ± 360°; the maximal linear resultant end tip speed is 8.2 m/s.

Fig. 4 shows the software structure for surface scanning and processing of depth map images.

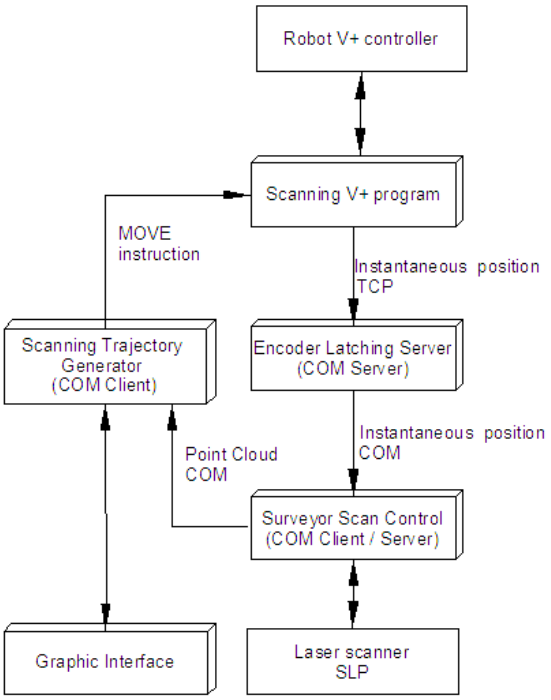


Fig. 4. Software architecture for laser object surface scanning and depth map image processing for toolpaths generation

To integrate the laser scanning and range finder system with the robot system, the following software modules must be developed:

- *Encoder Latching Server*: provides integration of the laser scanning control with the robot motion controller; the instantaneous Cartesian position of the Robot is transmitted to the distance acquisition software Surveyor Scan Control.
- *Trajectory planner*: computes the robot paths along which scanning of the surface of interest will be done, according to the user defined strategy (motion pattern).
- *Graphical User Interface* (GUI): is installed and runs on the PC.

The adoption of a team-based, distributed planning and control of sensory, robot and machine tool resources relies on integrating them in a modern information system developed as a workable Holonic Manufacturing Execution System (HMES), see Fig. 5.

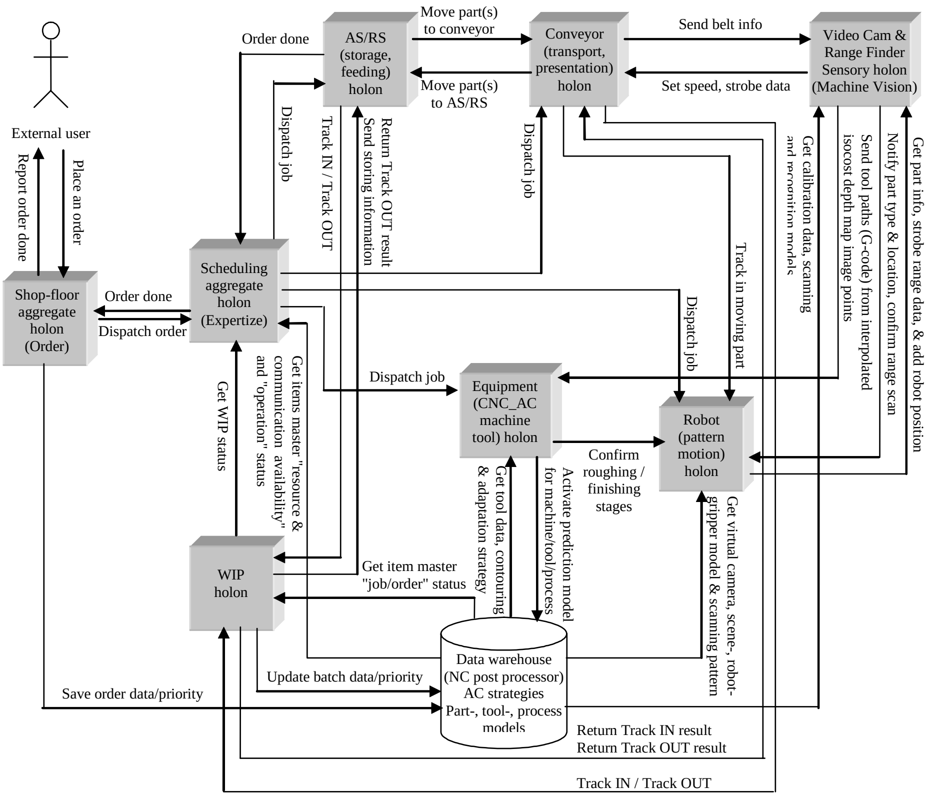


Fig. 5. HMES holarchy implementing team-based, multi-agent integrated sensory-robot-machine manufacturing platforms

The HMES Holarchy was designed using the multi-stage procedure of constructing an abstract object model based on domain knowledge, partitioning the application domain into components, identifying generic parts among components to form the Generic Holon, developing the Generic Holons, defining holarchy messages and the holarchy framework of the HMES, and finally designing *functional* holons based on *generic* ones: Order Holon (OH), Resource Holon (RH), Product Holon (PH), Work-In-Progress (WIP) and Expertise Holon (EH). The technologies of distributed object-oriented approach, design pattern, framework, N-tyer client/server architecture, and component software are applied to develop the entire HMES, its functional holons and cooperation mechanism.

Team-based manufacturing stations, including the scanner-robot-machine tool reverse engineering platform, are represented in such distributed structures by information entities – or **holons**, having manufacturing counterparts, and cooperating to solve together assigned pre-production, material conditioning (production), and post-production tasks.

A holon is an autonomous and co-operative building block of the manufacturing system for creating, transforming, transporting, storing and/or validating information and physical objects. It consists of an information processing part and a physical processing part. A holon can be part of another holon. It features *autonomy*, i.e. the capability to create and control the execution of its own plans and strategies, and *cooperativeness*, which means that the set of holons develops and executes mutually acceptable plans.

The following objectives of the holonic manufacturing control were proposed: (i) stability in face of disturbances (resource failures); (ii) adaptability, including generation of CAD data from models (reverse engineering) and quick response in face of changes (client orders); (iii) in line product quality control by artificial vision; (iv) efficiency of machine tools usage (adaptation of cutting processes).

Consequently, two *holarchies* (set of basic rules for cooperation between holons, integrating the entire range of manufacturing tasks from order booking to design, production and marketing to do the agile manufacturing enterprise) were designed to automatically switch between them at run time:

1. Hierarchical, optimal production planning and control in normal operating mode. In this mode, the P received customer orders and related specifications are first translated into production plans which contain the number, type, and precedence of design, manipulating, processing,, measuring and inspection operations upon each product. Then, using *Expertise Holons* as scheduling agents, an ordered sequence of *Order Holons* ($\mathrm{OH}_i, 1 \le i \le P$) of variable depth, corresponding to these production plans is off-line computed for the P final products such as to optimize a cost function at batch level: throughput, machine loading, etc. According to the type and configuration of the material transportation system (conveyor), the planned sequence of order holons (products) is split into n packets of $p \le P$ products and executed.

2. Heterarchical, graceful degraded production re-planning and control at the horizon of p products simultaneously in execution, in reaction to a change in customer order (including redefining rush orders), a resource failure or a negative result of operation execution or quality control operation.

Manufacturing resources are described by Resource Holons of three types:

- *Sensory Holons* ($\mathrm{SH}_{l,q}, 1 \le l \le s, 1 \le q \le q_s$) (laser scanners for material description, video cameras for product inspection, and RD/WR magnetic chips for product tracking on pallets). There are s such holons within each of the q_s sensory holon class.
- *Robot Holons* ($\mathrm{RH}_{l,q}, 1 \le l \le r, 1 \le q \le q_r$), describing all robot manipulators, grippers and tools together with their controllers, responsible for moving laser scanning devices to retrieve 3D part surface information, manipulating, mounting, and fastening assembling components, and for moving arm-mounted cameras in picture-taking points where the products are visually inspected.

• *Machine Tool Holons* $(\mathrm{MH}_{l,q}, 1 \le l \le m, 1 \le q \le q_m)$, encapsulating machines, clamping devices and tool sets, together with their embedded CNC and adaptive cutting processors.

In total, the HMES incorporates $\sum_{\mathrm{res}=s,r,m} (\sum_{1}^{\mathrm{res}} i \cdot q_i)$ resource holons.

In normal operating mode a Shop-Floor Holon receives a *place an order* message from an external user (client) and will reply report *order done* when this production command (order) is accomplished. Based on the received order, the Shop-Floor Holon will send *dispatch order* to the scheduling (expertise) holon; this latter one will reply *order done* if the order is finished. The Shop-Floor Holon sends *save order* information to a Data Warehouse to save all the order information. *Product Holons* $\mathrm{PH}_{\mathrm{o,mc,r}}$ are a priori created and stored in the Data Warehouse; they describe for the execution of each product the necessary operation types and precedence, material components, programs, technological data (calibration-, recognition-, grasping- and collision avoidance models), and types of material conditioning devices (sensors, robots, tools, machines).

Finally, to permanently estimate the current execution status both at single product and at batch level, *Work-In-Progress Holons* are defined to keep evidence of the production status, material flow and current capacities of the central and workstation component storages (AS/RS devices).

The interfacing holarchy messages of Scheduling and Order Holon, WIP Holon, Resource Holon, Data Warehouse and Material Handling (which includes AS/RS, Robot, Conveyor, Scanner and Vision) will be similarly defined.

Once the production schedule is computed, the output data – list of ordered production plans (order holons) is transferred as a binary file containing holons, operations and resources indexes to the execution mechanism – a PLC controlling the transportation system interconnecting all workstations.

In the event that the sequence of ordered production plans (order holons) is invalidated by changes in the system (e.g. a resource is not available), then the operating mode is switched from optimal to heterarchical, in which the remaining undone operations in the list of order holons are assigned to the valid resources according to an holonic auction procedure at which take part both the scheduling processor administrating the order holons and the resource holons which are still operational (Fig. 6).

No new palettes are inserted into the system until all the remaining operations of the products currently in execution assigned to valid resources and executed. Simultaneously with this auction a job re scheduling is initiated at p-product horizon, by considering the new working capabilities of the system. This new offline order scheduling will be used after the completion or evacuation of all the palettes that were in the system at resource breakdown time.

It is possible that after the breakdown of a resource some types of products would not be executed. In this case the related pallets will be evacuated and the products they carry marked as *unfinished*; from the remaining production plans some will be ignored (the ones requiring the failed resource) and marked as *non executable*.

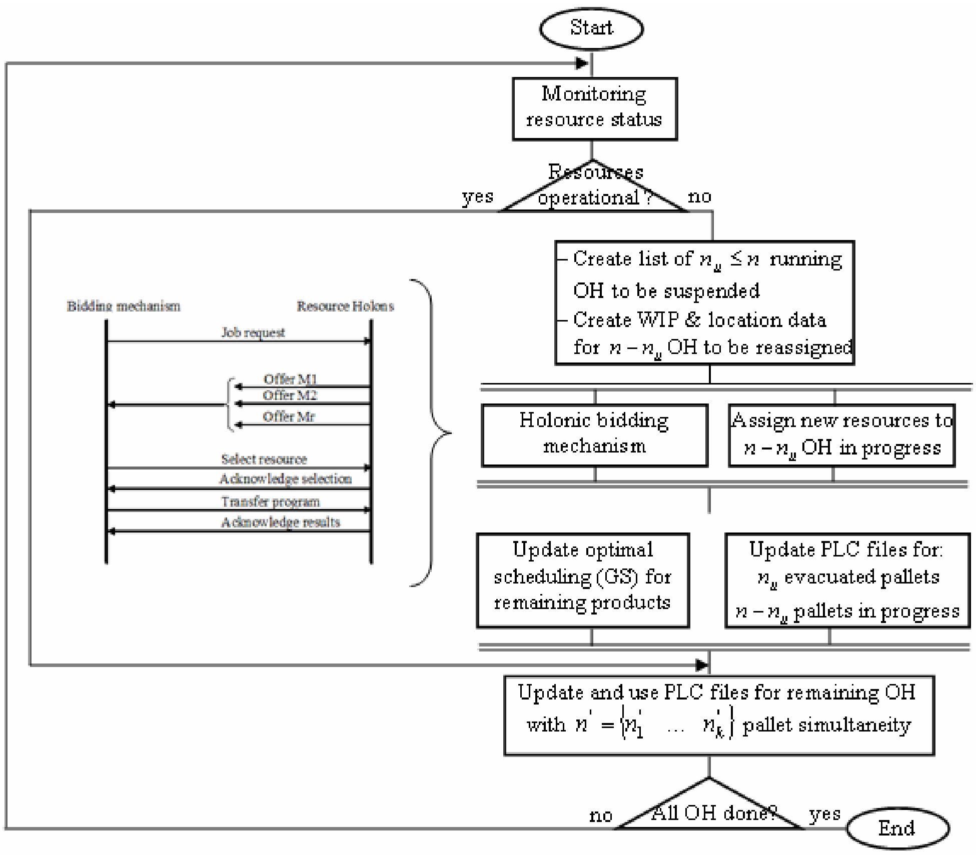


Fig. 6. HMES switching from the hierarchical operating mode to the heterarchical one by means of holonic bidding

The methodology of negotiation and coordination among holons is one of the bases for effective: (i) reconstruction of order holons from the list of production plans and (ii) re scheduling of remaining order holons, management and execution control in the distributed manufacturing system. A version of the Contract Net Protocol (CNP) is usually adopted to implement in real time these two processing stages, as represented in the repeated sequence of bidding actions given below and shown in Fig. 7.

WHILE rescheduled packet of p production plans (i.e. order holons to be simultaneously executed) is not completed DO

Create a new order holon:

FOR *first_operation* on selected product TO *last_operation* (task) on selected product

1. **Task announcement**: the initiator agent (*order holon progressively created*) broadcasts an announcement to the participant agents to call for proposal (cfp).

2. **Bidding**: Participants (*resource holons*) receiving the announcement, being able to evaluate the task and having the appropriate capability to execute it, return then their weighted bids to the initiator agent (order holon).

3. **Awarding**: the initiator agent awards the task to the most appropriate agent upon comparing the costs of the received proposals.

4. **Implementing**: the awarded participant is granted the current task (operation).

5. Update time.

END

END

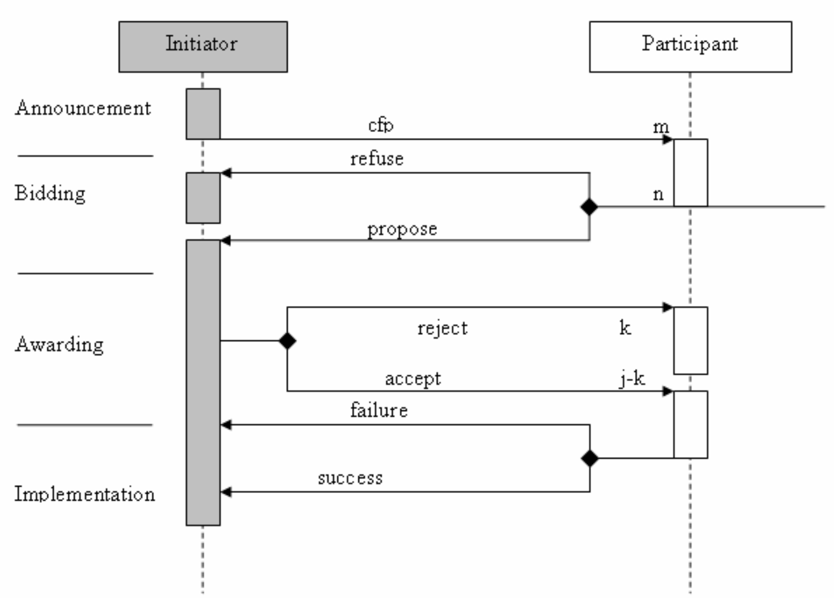


Fig. 7. FIPA Contract Net Protocol adopted for negotiation and coordination among holons at order replanning

As a current challenge in production control, the multi-agent heterarchical system will be approached with the new emergent Service-Oriented Architecture (SOA) concept, with the objective to reconcile the opposing principles of autonomy and interoperability. Thus, web services will be integrated with software agents to facilitate learning, management, re-configurability and evolution of the planning and control system.

SOA concepts are used to face the problems of interoperability in autonomous, team-based re-configurable multi-agent architecture implemented as a Holonic Manufacturing Execution System. As previously described, each smart device (robot, vision, laser scanner, conveyor, storage, etc.) controller will encapsulate functions and services that the physical device can perform, e.g. jogging the conveyor, moving the robot along a particular path pattern, etc. These services, that can be modified, added or removed (e.g. a new product can be handled by a robot after attaching a new gripper or manufactured on a milling machine after generating the G-code from depth map images produced by the laser scanner and range finder devices), are then exposed to be invoked by other smart device controllers.

Self-organization (the capability to dynamically re-organize itself in the presence of disturbances) and *learning* (the capability to acquire new knowledge supporting the dynamic behaviour evolution) mechanisms were considered to provide each HMES component with the capability to dynamically evolve during its life-cycle. SOA middleware is used to face the problems of interoperability in the form of service requester and service provider mechanisms, as suggested in Fig. 8.

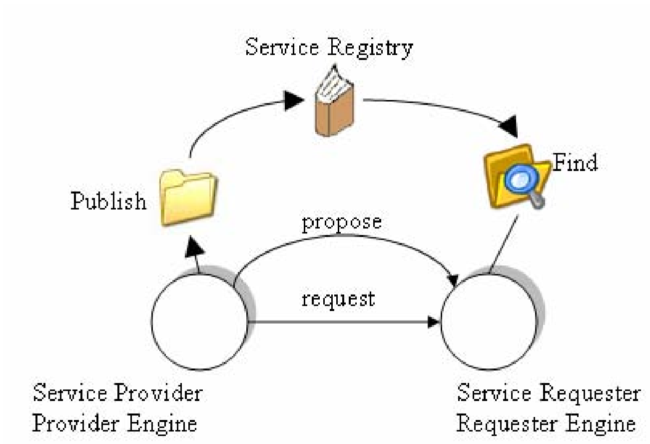


Fig. 8. The *service provider – service requester* mechanism used in the Service-Oriented Manufacturing Architecture

A provider hides its internal structure and shows only the necessary functionalities to the outside world, in the form of services. The list of provided services must be published, so they can be found by the service requester. A service discovery facility will act like a director in which services can be added, removed and located.

It is estimated that the integration of web services with manufacturing software agents brings benefits for the developed multi-agent, holonic control technology. The adoption of web services in the multi-agent, HMES holarchy will satisfy the requirements:

- Resources can be encapsulated with a service provider that acts like a bridge between the internal structure and the exposed interface.
- Some services can be composed by other services, creating a levelled structure of services (e.g. task-and product-oriented learning of virtual cameras).
- Interoperability in the HMES can be addressed by using common communication semantics based on the use of open protocols, namely web technologies (services).

- Fault-tolerant attributes are provided (anomalies that may occur during production processes, and identification in advance of possible future disturbances are handled).

3. Generating CNC toolpaths from depth map image processing

A theoretical approach and implementing issue are presented for automatic CNC tool path generation using morphological image processing operators applied to depth map images. This method allows computing cutter compensation for various end mill shapes. Roughing is made using zigzag cuts with a flat end mill, and finishing is done with isoparametric toolpaths using various rounded end mills. An AI-based algorithm and image processing software creates CNC machining instructions from depth map image models obtained with the laser scanner-robot system in Fig. 3.

Fig. 9 shows the diagram of the global software system developed for CNC part machining from digital 3D model surface data reconstructed from depth map images acquired with the laser scanning device.

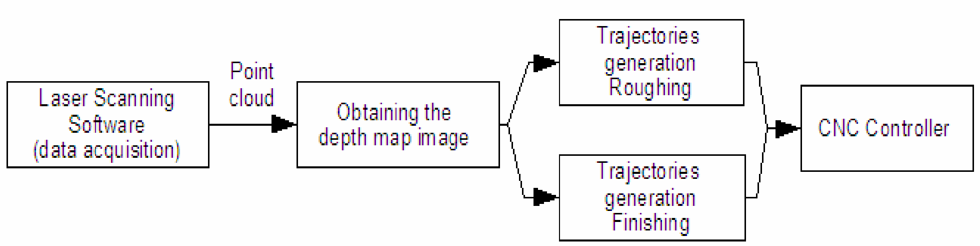


Fig. 9. Software diagram and data flow of CNC part machining from G-code extracted from depth map images

In part machining process, a phase-based approach is very common to shape a work piece into its final form. A typical sequence follows the template: roughing, semi-finishing and finishing. The most important objectives for the finishing phase are: meeting the imposed tolerances and surface finish. These objectives are opposed to the objectives of the roughing stage. Rough machining consists in fast and efficient material removal; the tool path generation for roughing operations is almost always done in 2.5D mode. This means that the geometry of the volume to be machined (called the "delta volume") is sliced by a set of parallel planes and that in each plane a 2D tool path is calculated (contour-offset, zig, zig-zag, etc.). The intermediate step to move between two successive parallel planes (0.5 dimension) determines the toolpaths to be disconnected in these planes which results in the generation of the "stairs" shape [12]. Four general types of toolpaths were considered: point-to-point, profiling, pocketing and roughing, and surface machining [13].

The type of object representation determines what geometric information is readily available and what must be computed. It also determines the information that can realistically be computed. For example, some representations support estimation of tool load by giving an estimate of the width and depth of cut for a given tool move. Other representations support the generation of partial or cleanup paths that direct the cutter only into the areas of the model containing material not removed. The choice of object representation therefore determines the types of possible path generation algorithms.

There are four general types of object representation: point-based, curve-based, surface- based and solid-based.

Point-based representations include both pixel based representations and formulations based on non-uniform point sampling. *Curve-based* representations involve planar profile geometry for pocketing and roughing. *Surface-based* representations include common surface schemes such as tensor product, B-spline surfaces, trimmed NURBS surfaces as well as surface approximations such as piecewise linear triangular mesh approximations. *Solid* representations include Constructive Solid Geometry (CSG) and B-Rep solids octree models, and even tensor product solids [14].

Surfaces that can be described by an arbitrary implicit function $z = f(x, y)$ may be stored as greyscale images, where the pixel's coordinates in the image are the real variables x and y, and the grey level encodes the z value. Usually the black colour maps to the lowest z level and the white colour maps to the highest. Because pixel coordinates are integers, a pixel-to-mm ratio must be used.

Most computer graphics software use 8 bit greyscale images and high end ones can use 16 bit greyscale images. For 16 bit images, the z level can have 65,536 discrete values, and this means, for a part of 100 millimetre height, a maximum precision of 0.0015 mm. The precision in XY plane is given by the image resolution. In order to get a precision of 0.01 mm for a 100×100 mm part, a 10,000×10,000 pixels image is needed. This makes the pixel model suitable only for low-precision applications, such as free-form artwork pieces.

Depth map images can be obtained through passive acquisition techniques like laser scanners and active acquisition techniques like structured light or generated with 3D modelling software packages.

3D objects were modelled using the freeware software POV-Ray. To create a depth map image, an orthographic camera is used, the texture and light sources are removed, and a gradient pigment is applied. The gradient is along the camera orientation axis. The gradient pigment is scaled and translated such that the farthest visible point from the camera maps to pure black and the closest point maps to pure white.

The 3D surface of a model to be machined is scanned with a laser range finder device held by a robot's gripper and moved along a predefined path pattern:

- A vertical stripe of laser light is moved across the model object surface, and captured by a video camera. Along each horizontal scan line of the video frame, the *brightest* spot is taken to be the point at which the laser stripe "hits" the surface (detection at sub-pixel resolution).
- The relative positions of the laser and the video camera are used to find the 3D coordinates of the brightest spot by triangulation.

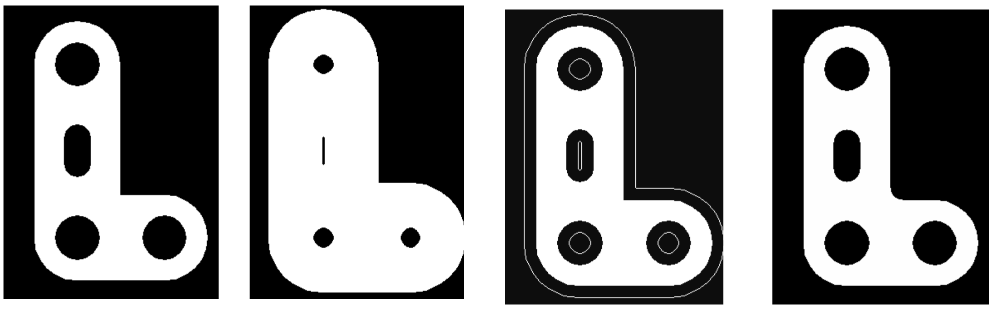
• The *x*-coordinate of each point in the output depth image is determined by the position of the laser stripe for a particular video frame;

• The *y*-coordinate corresponds to a raster line in the video frame;

• The depth value is computed from the brightness peak detected along the raster line in the video frame.

A 2D profile can be stored as a binary image, where white pixels are portions of material to be kept, and black pixels are the portions of material to be removed. Using this model, it is possible to generate gouge-free toolpaths for 2D profile milling by morphological dilation with a round-shaped structural element.

After dilation, the contour extraction algorithm (Moore-Neighbour) is used and the tool path is generated. By eroding the dilated image with the same structural element, one gets the actual machined part, where the corners that cannot be milled with the given tool are rounded (Fig. 10).



a b c d

Fig. 10. Generating 2D toolpaths by morphological erosion: a) the work piece model; b) dilated image; c) tool path; d) work piece with rounded corners

A speed increase for the morphological dilation is possible by processing only contour pixels of the original work piece. The advantage of this approach is that it can be used for any free-form 2D profile shapes, and it is not necessary to compute intersections between geometrical primitives. The method can test whether a given tool radius is appropriate for machining the 2D profile, by comparing the original image with the eroded one. However, the precision of the method is limited to the image resolution, and the processing speed is much lower than geometrical-based approaches.

The method can be extended for milling 2.5D surfaces stored as greyscale height-map images. It allows one to compute cutter compensation for different cutter shapes, including but not limited to flat end mills, ball end mills, and flat end mills with corner radius, known as bull end mills.

The algorithm is based on the following idea: at every location in the *XY* plane (i.e. any pixel in the image) one has to compute the depth which should be reached by the milling cutter, in order to be tangent to the surface. The shape of the milling cutter was modelled as a greyscale image, using the same scale factors as for the surface to be milled. The principle of discrete cutter compensation based on gradient computing in the 2D greyscale cutter shape image is given in Fig. 11.

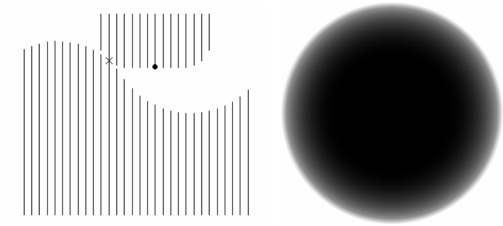


Fig. 11. Discrete cutter compensation from grey scale representation of the cutter shape

Suppose the work piece image is stored in a matrix of $M \times N$ pixels named A and the cutter image is a matrix of $m \times n$ pixels named B. We also use a $m \times n$ Boolean mask array which is true at the positions where the cutter image model contains valid values (i.e. the inside of the circle). The surface on which the tip of the milling cutter can move tangent to the work piece is computed in the image matrix C, which is also $M \times N$ pixels. The algorithm is:

```
for ia = 1 to M
  for ja = 1 to N
   h = +inf
   for ib = 1 to m
    for jb = 1 to n
     if mask[ib,jb] = true
      hc = A[ia+ib-int(m/2),
           ja+jb-int(n/2)] - B[ib,jb];
      h = min(h, hc)
     end
    end
   end
   C[ia,ja] = h;
  end
end
```

In a real implementation one must take care of image boundaries, but for simplicity these tests are omitted here. The algorithm has the complexity of $O(MNmn)$, or roughly $O(N^4)$. This significantly increases the computational time; however for some types of isoparametric toolpaths there is no need to compute the compensation for the whole surface. The computation process is equivalent to a grey scale morphological dilation, by using a non-flat structural element which is the negative of the grey scale cutter shape image.

Once the offset surface has been computed, isoparametric toolpaths can be generated easily by basic image manipulation operations. A tool path having the X parameter constant can be obtained by extracting a column from the image. In the same way, a tool path with the Y parameter constant is obtained by extracting a line from the image.

Toolpaths having the Z parameter constant (iso-level curves) can be computed by binarizing the image with a threshold corresponding with the desired Z level, followed by applying the contour detection algorithm on the binary image.

Toolpaths that follow a constant direction in *XY* plane can be obtained by first computing the points of the 2D line along that direction using the Bresenham algorithm, and reading the grey values (heights) from these points.

The toolpaths generated in this way can be used for finishing the work piece, only after the roughing cycle has been executed. To generate roughing toolpaths, one should consider the following:

- The roughing cutter should not touch the model's surface; instead, it should keep a small constant distance from the model, e.g. 1 mm. This is because roughing cutters are less precise than finishing ones, and we want to make sure roughing cuts won't be visible on the finished product.
- The roughing cutter must not plunge into the material more than a maximum allowed depth. For example, if we want to mill a pocket having 15 mm depth, we should first plunge the cutters at 5 mm, then at 10 mm and lastly at 15 mm.

For roughing cuts only toolpaths at constant *Z* level and flat end mills were used. This allows one to use the 2D tool path compensation algorithm, which is much faster. First we get a binary image by thresholding the original greyscale surface at the desired *Z* level. To achieve a small offset between the cutter and the ideal model, we compute the offsets using a larger cutter radius, i.e. we add the offset value to the actual cutter radius. That is, if we use a cutter having a radius of 10 mm, we will compute the cutter compensation with a radius of 11 mm. Also, we will subtract the offset from the plunging depth, i.e. if the image is thresholded at $Z = -5$ mm, we will plunge the cutter at $Z = -4$ mm. This leads to a distance in the range [offset ... offset × 1.41], depending on the local surface derivative.

For precise offsetting, the tool profile should be modified using an algorithm similar to 2D cutter compensation, and we should use the 3D compensation algorithm which is slower. The algorithm for generation roughing cuts, in pseudo code, is given below:

```
for Z = –dz to –MZ step  –dz
        B = threshold(model, Z)
        se=round_structural_element(tool_radius + offset)
        Boff = dilate(B, se)
        make_roughing_cut(Boff, Z + offset)
end
```

4. Experimental results. Conclusion and future work

A design methodology and cost-effective implementing solution for reverse engineering of complex-shaped objects using interlaced numeric and adaptive machining control has been reported. Scanning of object surfaces is performed using a short-range laser device moved by the robot gripper along configurable patterns. Image enhancement techniques are developed for noise removal, data thinning and multiple point clouds registration eliminating data overlapping. The 3D model of the object of interest is then reconstructed using depth map images.

An innovative, low computation effort solution is proposed for generating the toolpaths for complex 3D surface processing on milling machines. The generated adaptive toolpaths are optimized in order to increase accuracy, reduce tool wear and machining time. Tool compensation is obtained using an original method, which consists in generating the height map of the active tool profile; by applying image morphology techniques one obtains the surface to be machined.

A graphical user interface (GUI) for tool paths generation using depth map images has been developed. The program allows defining the milling tool shape and performs cutter compensation. It is possible to view the part in different stages of the machining process. A simulation of the milling operations can also be displayed. After loading a depth map image, the dimensions of the final product can be set by specifying the pixel to mm ratio and the depth of cut (Fig. 12).

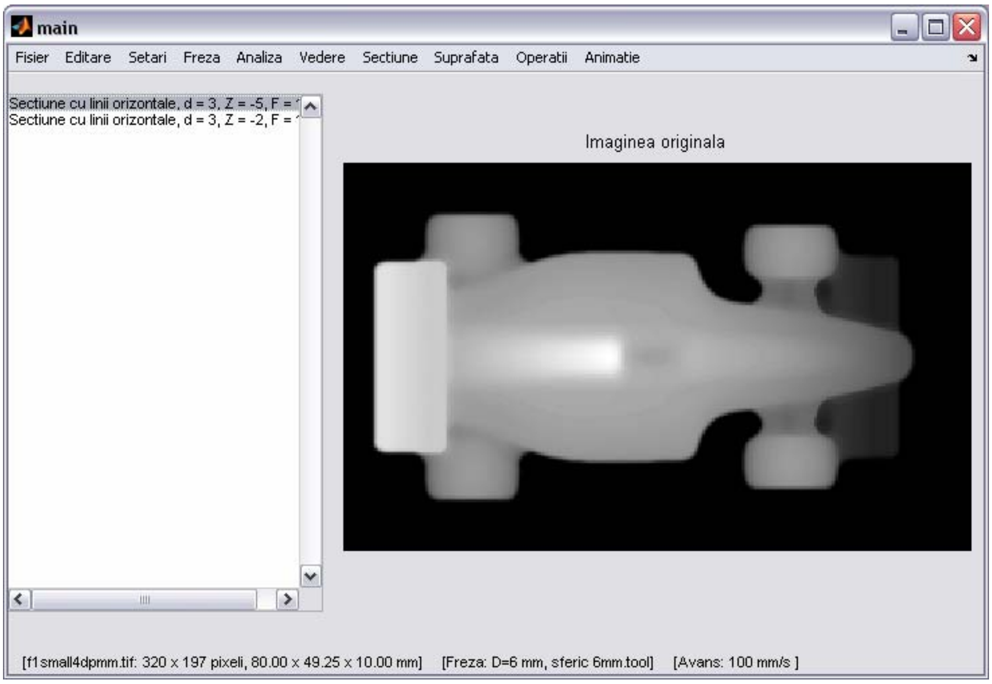


Fig. 12. Graphical user interface main window (sample model: F1 car)

The user can define the milling tool shape (Fig. 13). Based on these definitions, the program computes the surface on which the tool tip is moving, to always keep the tool tangent to the model.

The set of milling tools includes three predefined types (ball end mill, flat end mill and conic end mill) or any shape defined by the user, by means of two arrays *X* and *Y* representing the milling tool section. The user defined milling tool can be saved in a **.**tool extension type file.

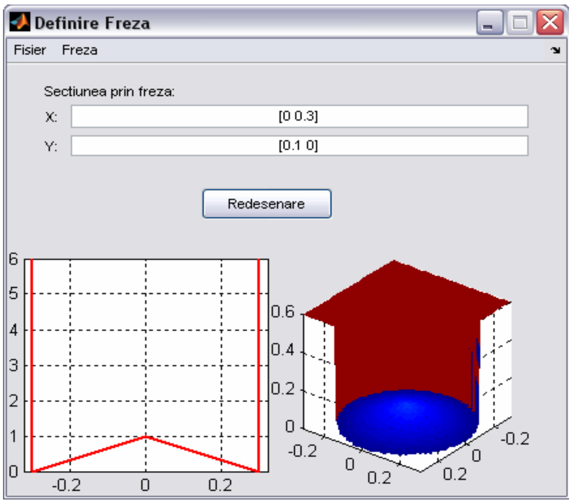
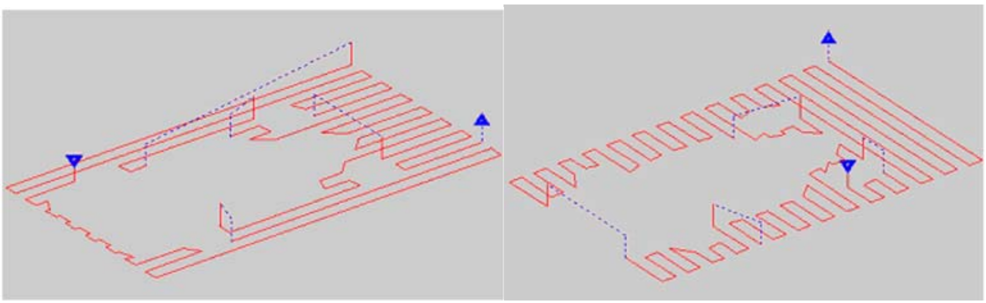


Fig. 13. Defining a milling tool shape (*Tool Shape* editor in the GUI)

Parts were executed in two stages: roughing and finishing. The rough machining is realised using trajectories parallel to the *OX* or *OY* axis as shown respectively in the graphical windows in Fig. 14 a and b (sample part: F1 car reconstructed from the height map image in Fig. 12).

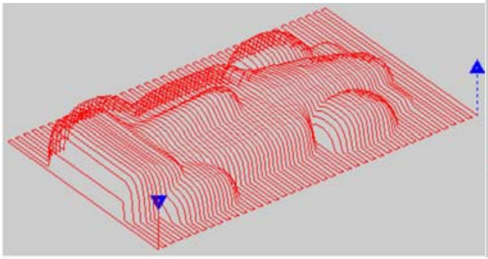


a b

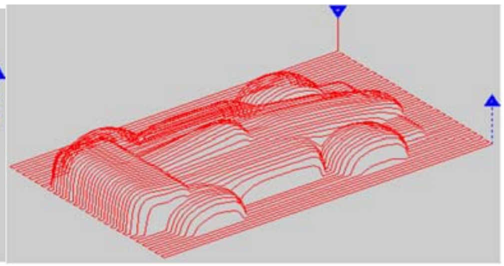
Fig. 14. Generating roughing toolpaths using trajectories parallel to the *OX* axis (a) and *OY* axis (b)

Each roughing pass is performed at constant *Z* level. At a given *Z* level, selecting the region where the cutter should remove material is done by an image thresholding operation. For flat end mill cutters 2D offset compensation was used.

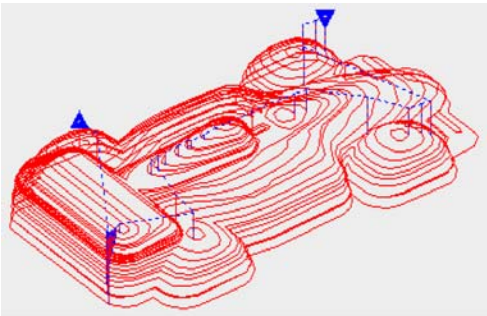
Finishing toolpaths were experimented in several ways: paths in *XZ* plane, in *YZ* plane and iso-level paths (Fig. 15 a, b, and c). The distance between the isoparametric toolpaths can be defined by the user, based on the type of the surface to be machined. The finishing method and the distance between isoparametric toolpaths can be redefined at any time after partial machining.



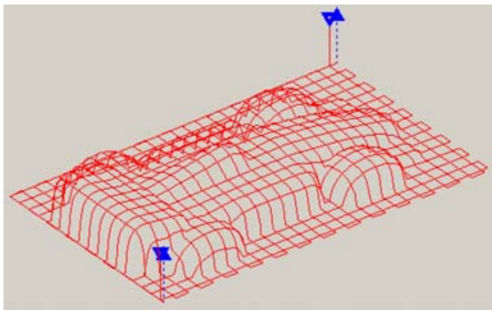
a



b



c



d

Fig. 15. Generating finishing toolpaths using trajectories in the *XZ* plane (a), *YZ* plane (b), isolevel (c) and combined (d) trajectories

In any of these machining strategies, the tool was modelled as a greyscale image, the part image is eroded using the tool model, and finally the eroded image is compared with the original model. Toolpaths were interpolated with linear lines and circular arcs.

Future developments will consider approximation of successive contour segments with circular arcs. Collision detection will be also checked by modelling the complete tool shape including the tool holder. An embedded numerical-adaptive machining control will be tested by varying the feedrate and speed at roughing to save time, increase throughput, and efficiently use the tool during its lifetime.

R e f e r e n c e s

1. B o r a n g i u, T. Intelligent Image Processing in Robotics and Manufacturing, Bucharest, Publishing House of Romanian Academy, 2004.
2. R a h i m i f a r d, S. Semi-Heterarchical Production Planning Structures in the Support of Team-Based Manufacturing. – International Journal of Production Research, Vol. **42**, September 2004, No 17, 3369-3382.
3. B a b i c e a n u, R. F., F. C h e n. Development and Applications of Holonic Manufacturing Systems: a Survey. – Journal of Intelligent Manufacturing, Vol. **17**, 2006, 111-131.
4. R a t c h e v a, V., S. V y a k a r n a m. Exploring team formation processes in virtual partnership. – Journal of Integrated Manufacturing Systems, Vol. **12**, 2001, 512-523.
5. V a l c k e n a e r s, P., H. V a n B r u s s e l, J. W y n s, L. B o n g a e r t s, P. P e e t e r s. Designing Holonic Manufacturing Systems. – Robotics and Computer-Integrated Manufacturing, Vol. **14**, 1998, 455-464.

6. V a n B r u s s e l, H., J. W y n s, P. V a l c k e n a e r s, L. B o n g a e r t s, P. P e e t e r s. Reference Architecture for Holonic Manufacturing Systems: PROSA. – Computers in Industry, **37**, 1998, 255-274.
7. L e i t a o, P., F. R e s t i v o. ADACOR: A Holonic Architecture for Agile and Adaptive Manufacturing Control. – Computers in Industry, **57**, 2006, No 2, 121-130.
8. C h e n g, F. -T., C. -F. C h a n g, S. -L. W u. Development of Holonic Manufacturing Execution Systems. – In: Advanced Robotic Systems International, Pro Literatur Verlag, Mammendorf, 2007, 77-100.
9. B a b i c e a n u, R. F., F. F. C h e n, R. H. S t u r g e s. Framework for the Control of Automated Material-Handling Systems Using the Holonic Manufacturing Approach. – International Journal of Production Research, **42**, September 2004, No 17, 3551-3564.
10. L e i t a o, P., J. M e n d e s, A. W. C o l o m b o, F. R e s t i v o. Reconfigurable Production Control Beyond ADACOR. – In: Preprints of the IMS'07 IFAC Workshop on Intelligent Manufacturing Systems, Alicante, Spain, 2007, 138-143.
11. B a r a t a, J. Coalition Based Approach for Shop Floor Agility. Lisbon, Orion Edition, 2005.
12. L e f e b v r e, P., B. L a u w e r s. 3D Morphing for Generating Intermediate Roughing Levels in Multi-Axis Machining. – Computer-Aided Design & Applications, Vol. **2**, 2005, No 1-4, 115-123.
13. E l b e r, G. Freeform Surface Region Optimization for Three- and Five Axis Milling. – Computer-Aided Design, **27**, 1995, No 6, 465-470.
14. D r a g o m a t z, D., S. M a n n. A Classified Bibliography of Literature on NC Tool Path Generation. – Computer-Aided Design, **29**, 1997, No 3.

Направления в SOA архитектурах для промышленного управления CNC роботом

Теодор Борангиу

Политехнический университет, Букурещ, Факультет автоматического управления и компьютер, 313, Spl. Independentei, Sector 6, RO-060032, Румыния
E-mail: borangiu@cimr.pub.ro

(Р е з ю м е)

Работа анализирует и дискутирует подход SOA архитектур для управления производством, интегрируя лазерные сканирующие устройства. Обсуждается также движение робота по выбранной траектории, обработку полутоновых изображений и генерирование адаптивных CNC траекторий инструмента в системах, основанных на знаний, в самообучающихся системах, динамически доступных в четырехстепенной архитектуре: наблюдение, предсказание, стратегия и управление.