EMBEDDED NUMERIC-ADAPTIVE MACHINING CONTROL BASED ON DATA FUSION

Theodor Borangiu, Anamaria Dogar, Alexandru Dumitrache,
Florin Daniel Anton, Silvia Anton, Nick-Andrei Ivanescu

University Politehnica of Bucharest, Dept. of Automation and Applied Informatics,
313, Spl. Independentei, sector 6, 060032 Bucharest, E-mail: cimr@cimr.pub.ro

Abstract: The paper presents a design methodology and low-cost implementing solution for interlaced numeric and adaptive control of machining operations on vertical milling machines. The low-cost attribute of the NC contouring function resides in generating machining trajectories from successive grey level image processing of the part's model with an estimated cutting depth, feedrate and spindle speed. Two of these reference signals, the feedrate and spindle speed, are then periodically altered as control variables according to a low-cost implementing solution of a machining optimization algorithm. The approach used for real-time updating the feedrate and spindle speed is based on nonlinear, inequality constrained multivariable optimal control (ACO) and modelling of the cutting process. The image-based contouring and machine load-based optimization functions are embedded in a multitasking controller. A low cost implementing solution is proposed by extending the CNC unit with a signal acquisition and processing ACO unit performing the fusion of electric and mechanical data. Experimental results are included for the 3D milling of a car body mould.

Keywords: embedded systems, CNC, image interpolation, optimal control, non-linear control.

1. INTRODUCTION

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 map to 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 65536 discrete values, and this means, for a 100 millimetre tall work piece, 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 100x100 mm part, a 10000x10000 pixels image is needed. This makes the pixel model suitable only for low-precision machining applications, such as free-form artwork pieces.

The 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.

Short range laser scanners are used to capture the geometry of three-dimensional objects. They are used for reverse engineering manufactured objects, and for digitizing objects of scientific, artistic or historical importance for archiving and analysis. Most of the short range laser scanners are instruments using CCD cameras where distance measurement is based on the “triangulation” principle.

Laser range scanners have limitations, including noise and limits on resolution. Noise in the depth images from laser range scans comes from several sources, including quantization and noise in the video imaging system, laser speckle (caused by random reinforcement of the coherent light of the laser rejected from a rough surface), systematic error in peak detection (caused by surface curvature and colour), and the instability of the computation of point locations by triangulation.

Currently the interest is to find low-cost solutions replacing with sufficient accuracy the sophisticated mathematical CNC processors which describe the 3D complex surfaces or shapes, being then mapped to G-code by CNC post processors. The research reported in the paper describes a method for 3D complex shape modelling (e.g. negative mould shapes) by generating iso-level tool paths from grey level depth map images binarized with variable thresholds. Tool correction is then applied by simply gradient computing in the 2D greyscale cutter shape image.

For individual iso-level tool paths a low-cost control solution is proposed at G-phrase level by discretely altering the feedrate and main spindle speed (w, n), which optimizes cost functions like the machining throughput and balances the machine tool loading.

3. $2^{1/2}$ SHAPE COUNTOURING FROM DEPTH MAP IMAGES

The 3D surface of a model to be machined is scanned with a laser range finder device: 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. This brightness peak is detected at sub-pixel resolution. The relative positions of the laser and the video camera are used to find the three-dimensional coordinates of the brightest spot by triangulation. So, 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, and the depth value is computed from the brightness peak detected along the raster line in the video frame (see Fig. 1).

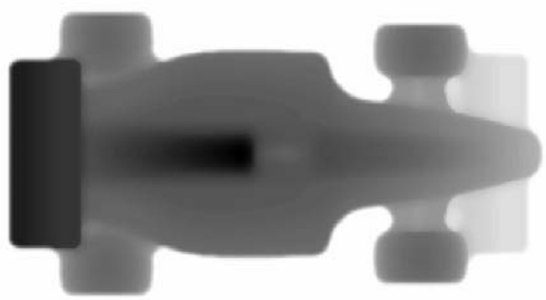

Fig. 1. Depth map image of a F1 car body model.

A solution has been developed for $2^{1/2}$ milling of surfaces stored as greyscale height-map images. This method 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 negative image from which the $2^{1/2}$ object mould will be machined is obtained by complementing the brightness peaks stored in each point of the stored depth map image (Fig. 2).

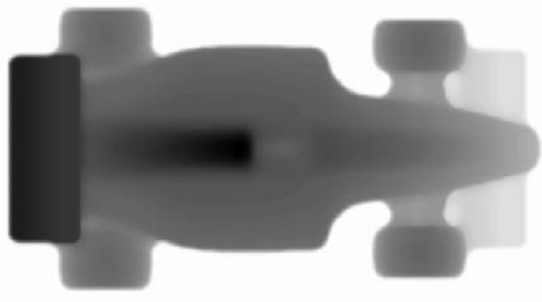

Fig. 2. Negative depth map image of the F1 car.

Once the negative depth map image created, the dimensions of the part to be machined can be set by specifying the pixel to mm ratio and the depth of cut. The tool shape is defined; based on that, the program computes the surface on which the tool tip is moving such that the tool is always tangent to the model.

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 is 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. 3.

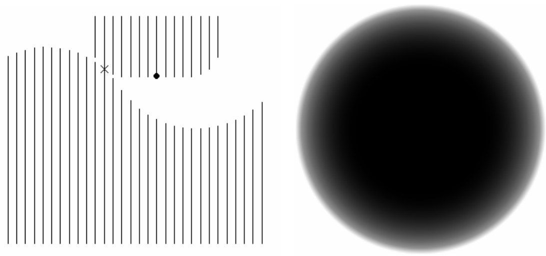

Fig. 3. Discrete cutter compensation (section). Greyscale representation of the cutter shape.

For roughing cuts only tool paths at constant *Z* level are used, and a flat end mill. First a binary image is obtained by thresholding the original greyscale surface at the desired *Z* level (Fig. 4). A contour detection algorithm is then applied for each binary image. To achieve a small offset between the cutter and the ideal model, the offsets are computed using a larger cutter radius, i.e. by adding the offset value to the actual cutter radius.

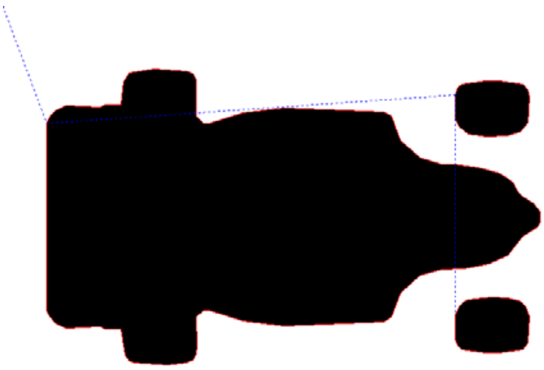

Fig. 4. Thresholded F1 image for Z = −5mm. The detected contour and the tool trajectory.

After the roughing cycle has been executed, new tool paths must be generated for the finishing stage. The finishing cycle can be optimally done by using a combination of three possible types of iso parametric tool paths. The cutting depth (distance between the cuts) can be defined by the user such as to provide a good quality of the final machined surface.

Iso parametric tool paths can be generated easily by basic image manipulation operations. Tool paths having the *Z* parameter constant (iso-level curves) can be computed as previously described. But for the finishing phase it is recommended to use a smaller distance between the iso-level curves.

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 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 (Park, 2005), and reading the grey values (heights) from these points (see Fig. 5).

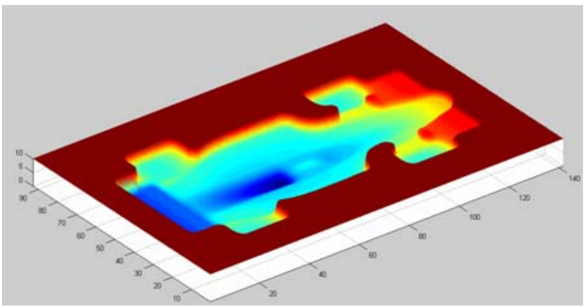

Fig. 5. Simulation of a finished part using only iso-level toolpaths.

After defining the operations for the roughing and finishing stages of part machining, the G-code is generated the TAP format.

Arcs can be in only one quadrant; if a generated arc spreads on more quadrants, the conversion program will decompose the big arc in several smaller arcs, each of them in a single quadrant. Arcs having the radius bigger than the superior limit of the CNC machine (for the EMCO machine used, the limit radius value is 200 mm) are approximated by lines.

3. THE INEQUALITY CONSTRAINED OPTIMAL CONTROL (ACO) OF MACHINING

The cutting productivity is considered as quality function for each one of the 2D closed *XY* roughing paths p_i approximating one locus of spatial points of same grey level (points of depth $Z_i, Z_i = Z_{i-1} - \Delta_j$ in the part model image). The first paths to be machined correspond to the loci of nearest points relative to the range sensor (grey level value Z_0), with cutting depth Δ_0. If the furthest image points relative to the sensor have a grey level Z_f, then, by piecewise estimating the grey level gradient in the point depths range $Z_0 ... Z_f$, the number of C cutting passes is computed to plan the $2^{1/2}$ approximation of 3D machining with cutting depths Δ_j, such that $\sum_j c_j \cdot \Delta_j = Z_f - Z_0$, and $\sum_j c_j = C$, and c_j roughing paths p_i have the same cutting depth Δ_j.

The computation of the C machining passes takes into account both *form accuracy* (piecewise change of the cutting depth Δ_j on grey level gradient basis) and *material characteristics* (type and hardness of material to be removed impose upper limits on Δ_j). The result is that each roughing path p_i should be realized at constant cutting depth Δ_j, feedrate $w_j = \mathrm{F}$ and spindle speed $n_j = \mathrm{S}$ to protect the tool and the machine's kinematical chain.

The ACO approach proposed maintains Δ_j constant at roughing path level, but *varies* in real time w, n when the tool moves along trajectory p_i – computed from image points of same grey level – in order to optimize a cost function.

The productivity of cutting is taken as cost function, being expressed as the ratio I between the mass of material removed in a time equal to the tool life, and the machining cost (Milner, 1974):

$$I = \frac{\mathrm{K} \cdot w \cdot \mathrm{B} \cdot \Delta \cdot \rho \cdot \mathrm{T}}{\mathrm{C_s} + \mathrm{k_s} \cdot \mathrm{C_s} \cdot \mathrm{P_s} \cdot \mathrm{t_s} + \mathrm{C_t}(\mathrm{t_r} + \mathrm{T}) + \mathrm{C_e} \cdot P_c \cdot \mathrm{T}} \quad (1)$$

where w [mm/min]: feedrate, B [mm]: cutting width, Δ [mm]: cutting depth, ρ [t/m³]: material density, $\mathrm{C_s}$ [$]: tool cost, $\mathrm{C_t}$ [$/min]: worker's wage rate, $\mathrm{C_e}$ [$/KWh]: energy cost, $\mathrm{P_s}$ [KW]:power consumed during tool change, P_c [KW]: power consumed when cutting, $\mathrm{t_s}$ [min]: time for tool change, $\mathrm{t_r}$ [min]: time for tool regrinding, T [min]:tool life, $\mathrm{K}, \mathrm{k_s}$:constants

The allowable working domain is defined as the area in the control space (w, n) bounded by the constraints corresponding to limit values taken by six parameters completely characterizing the machining process:

(1) Maximum torsion torque at the main spindle:

$$M_t \le \mathrm{M}_{\mathrm{t,max}}\text{, or } w\cdot n^{-1} \le \mathrm{K}_{\mathrm{t}} \quad (2)$$

(2) Maximum deflection torque at the main spindle:

$$M_d \le \mathrm{M}_{\mathrm{d,max}}\text{, or } w\cdot n^{-1} \le \mathrm{K}_{\mathrm{d}} \quad (3)$$

(3) Pitch feed *pf*:

$$\mathrm{pf}_{\min} \le pf \le \mathrm{pf}_{\max}\text{, or } \mathrm{K}_{\mathrm{fm}} \le w\cdot n^{-1} \le \mathrm{K}_{\mathrm{fM}} \quad (4)$$

(4) Kinematics of the feed drive:

$$\mathrm{w}_{\min} \le w \le \mathrm{w}_{\max} \quad (5)$$

(5) Protection of main spindle and tool:

$$\mathrm{n}_{\min} \le n \le \mathrm{n}_{\max} \quad (6)$$

(6) Main drive power limitation:

$$P \le \mathrm{P}_{\max}\text{, or } w^{yp}\cdot n^{1-yp} \le \mathrm{K}_{\mathrm{p}} \quad (7)$$

Considering all six constraints, the allowable cutting domain appears as the interior of the (w,n) area with bold frontier represented in Fig. x. The domain varies in time, as M_t, M_d, pf, and P are subject to changes due both to variations in material hardness or local non homogeneity and to periodic updates of the w,n controls.

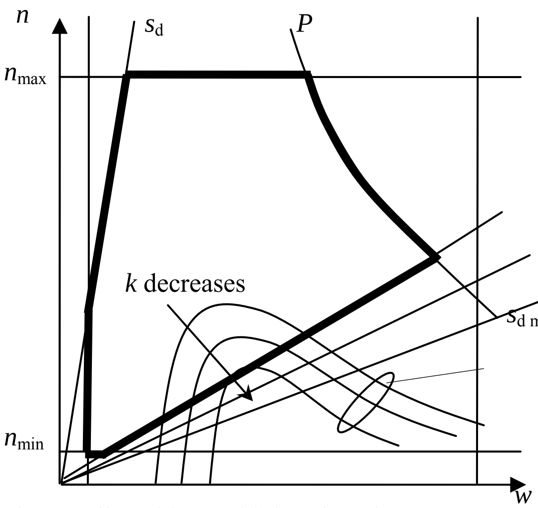

Fig. 6. Allowable machining domain.

By substituting in (1) the expression of the tool life T (T depends nonlinearly on *w* and *n*), maximizing the cost function *I* leads to minimizing $J = 1/I$:

$$J(w,n) = \mathrm{a}\cdot w^{\mathrm{q/m-1}}\cdot n^{(1-\mathrm{q})/\mathrm{m}} + \mathrm{b}\cdot w^{\mathrm{yp-1}}.n^{1-\mathrm{yp}} + \mathrm{c}\cdot w^{-1}$$

where $\mathrm{a, b, c, q, m, yp}$ are constants, the last three being given by Taylor's tool life formula (Shaw, 1986). The isocost curves of the function $J(w,n)$ are given in Fig. 6.

The optimization strategy is based on the localization of the minimum point of the cost function $J(w,n)$ in the convex allowable domain. The inequality constrained problem is solved using directional derivatives (Braswell, 1972). The problem is first converted to an equality constrained optimization problem with the usual assumptions concerning the differentiability of the objective function $J(w,n)$ and of the constraints (2) – (7).

Let X be an open set in $\mathbf{R}^{\mathrm{N}}$ and $f, \{g_i\}, i = 1,\ldots,p$ numerical real valued continuous and differentiable functions defined on X. The problem of interest is to find a point x_0 that locally minimizes $f(x)$, where $x \in X_1$:

$$\text{Problem A}: \begin{cases} X_1 = \{x \mid x \in X, g_i(x) \le 0, i = 1,2,\ldots,p\} \\ \text{or } \min\limits_{x\in X_1} f(x) \end{cases}$$

A point $x_0 \in X_1$ locally minimizes Problem A if there is a vector ε such that

$$f(x) - f(x_0) \ge 0 \text{ for all } x \in X_1 \text{ and } |\mathrm{x} - \mathrm{x}_0| < |\varepsilon|$$

Hence if an approximation of f in the neighbourhood of x_0 can be found for all $x \in X_1$ and $|x - x_0| < |\varepsilon|$, then Problem A can be treated as an unconstrained one, using directional derivatives.

Problem A is converted to an equality constrained optimization problem by adding slack variables:

$$\text{Problem B}: \begin{cases} \min\limits_{x\in X} f(x) \\ \text{together with } g_i(x) + x_{\mathrm{N}+i}^2 = 0, i = 1,\ldots,p \end{cases}$$

Problem B is an equality constrained optimization problem of $\mathrm{N}+p$ variables and p constraints, and can be solved using directional derivatives. Feasible direction vectors are defined as vectors tangent to the hyper planes resulting from the first approximation of the constraints. The feasible direction vectors $\mathbf{h}$ are determined from equations (8):

$$\mathbf{h}^{\mathrm{T}}\mathbf{G} = 0 \quad \mathbf{h}^{\mathrm{T}}\mathbf{h} = 1 \quad (8)$$

where $\mathbf{h}$ is a $(\mathrm{N}+p)\times 1$ vector and $\mathbf{G}$ is a $(\mathrm{N}+p)\times p$ matrix defined as

$$\mathbf{G} = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \ldots & \frac{\partial g_p}{\partial x_1} \\ \ldots & \ldots & \ldots \\ \frac{\partial g_1}{\partial x_{\mathrm{N}}} & \ldots & \frac{\partial g_p}{\partial x_{\mathrm{N}}} \\ 2x_{\mathrm{N}+1} & 0 & 0 \\ 0 & 2x_{\mathrm{N}+2} & 0 \\ \ldots & \ldots & \ldots \\ 0 & 0 & 2x_{\mathrm{N}+p} \end{bmatrix} \quad (9)$$

with $x_{\mathrm{N}+i}^2 = -g_i(x_0), i = 1,\ldots,p$

The system (8) determines N linear independent feasible direction vectors provided the implicit function theorem holds in an extreme point x_0. From all solutions only N are independent; two cases are analysed: (1) $p < \mathrm{N}$ ($\mathrm{N} = 2$ as $x = [w,n]^{\mathrm{T}}$); (2) $p \ge \mathrm{N}$.

Let $J(w,n)$ be the cost function for the machining process with $x = (w,n) \in X = \{x \mid w \ge 0, n \ge 0\}$ and one single constraint from (2), (3) or (4), having the form

$$g(w,n) = C_1 w - n \le 0, C_1 = \text{constant} \quad (9)$$

Then, Problem A is: find point $x_0 = (w_0, n_0)$ such that $J(x_0) = \min_{(w,n)} J(w,n), w \geq 0, n \geq 0, g(w,n) \leq 0$.

By adding a slack variable z, $g(w,n) \leq 0$ of Problem A is converted to $g^*(x) = g(w,n) + z^2 = 0$ of a Problem of type B, where $x = (w,n,z)$ and $\mathbf{G} = [\mathrm{C}_1 \quad \text{-1} \quad 2z]$.

Clearly, the implicit function theorem holds for w. The sub matrices $\mathbf{G}^{(i)}, i = 1,2$ of $\mathbf{G}$ that generate the feasible direction vectors $\mathbf{h}_1$ and $\mathbf{h}_2$ have the form $\mathbf{G}^{(1)} = [\mathrm{C}_1 \quad \text{-1}]$, $\mathbf{G}^{(2)} = [\mathrm{C}_1 \quad 2z]$, and the vectors of feasible directions are:

$$\mathbf{h}_1 = \frac{1}{\sqrt{1+\mathrm{C}_1^2}} \cdot \begin{bmatrix} -1 \\ -\mathrm{C}_1 \\ 0 \end{bmatrix}, \quad \mathbf{h}_2 = \frac{1}{\sqrt{\mathrm{C}_1^2 + 4z^2}} \cdot \begin{bmatrix} 2z \\ 0 \\ -\mathrm{C}_1 \end{bmatrix}$$

The necessary condition for a point x_0 to be a local minimum of Problem B when $p < \mathrm{N}$ ($p = 1, \mathrm{N} = 2$) imposes that the first order directional derivative of J vanishes, which gives:

$$D_{\mathrm{h1}} J(w_0, n_0, z_0) = \frac{1}{|\mathbf{h}_1|} \cdot \mathbf{h}_1 \cdot \nabla J(w_0, n_0, z_0) = 0$$

$$D_{\mathrm{h2}} J(w_0, n_0, z_0) = \frac{1}{|\mathbf{h}_2|} \cdot \mathbf{h}_2 \cdot \nabla J(w_0, n_0, z_0) = 0$$

From these equations, the following stationary points $x_0 = (w_0, n_0, z_0)$ are obtained:

$$w_0 = \mathrm{C}_1^{\text{q-1}} \cdot \left[\frac{\mathrm{mc}}{(1\text{-}\mathrm{m})\mathrm{a}}\right]^{\mathrm{m}}, \quad n_0 = \mathrm{C}_1^{\mathrm{q}} \cdot \left[\frac{\mathrm{mc}}{(1\text{-}\mathrm{m})\mathrm{a}}\right]^{\mathrm{m}},$$

$$z_0 = 0 \tag{10}$$

To check the type of stationary points (10), one must further investigate, by means of 2nd order directional derivatives, the second order conditions. It was demonstrated that the matrix $\mathbf{D}$ of the quadratic form $D_{\mathrm{h}}^2 J(x_0)$ of 2nd order directional derivative in (11):

$$\mathbf{D} = \begin{bmatrix} \frac{1}{1+\mathrm{C}_1^2} \cdot \frac{\mathrm{c}}{\mathrm{m}(1\text{-}\mathrm{m})} \cdot \frac{1}{w_0^2} & 0 \\ 0 & 0 \end{bmatrix} \tag{11}$$

is positive semi definite, which requires a set of stronger condition. These new conditions are derived from the *corrected* 2nd order directional derivatives. For Problem B, the matrix $\mathbf{D}^* = \lfloor D_{ik}^* \rfloor, i = 1,2; k = 1,2$ with $D_{ik}^* = \mathbf{h}_i^{\mathrm{T}} \nabla D_{hk} J(x)$ is positive definite, which indicates that the stationary point (10) computed from necessary conditions is a local minimum.

The computing results are also confirmed by the form of the isocost curves of the function $J(w,n)$. The optimum point will be placed on the boundary of the allowable operating domain, at the intersection of the constraint $\mathrm{C}_1 \cdot w - n = 0$ with the computed curve

$$w^{\mathrm{q}} \cdot n^{1-\mathrm{q}} = \left[\frac{\mathrm{mc}}{(1\text{-}\mathrm{m})\mathrm{a}}\right]^{\mathrm{m}}, \tag{12}$$

as represented in Fig. 7.

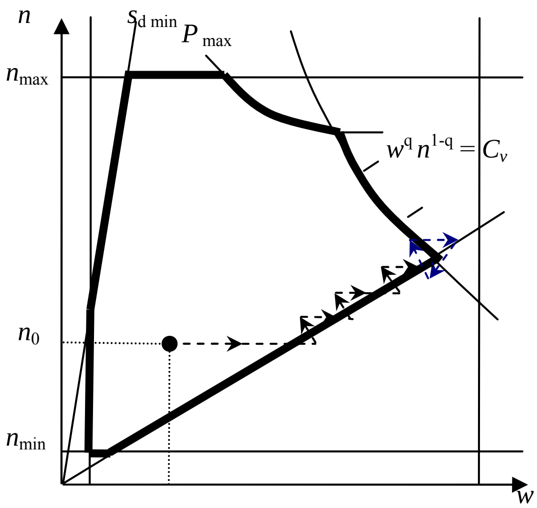

Fig. 7. Locating the optimum point at the intersection of a "mechanical load" constraint $\mathrm{C} \cdot w - n = 0$ (torsion, deflection or pitch feed) with one of the type: electric power, feedrate or computed curve.

For case 2 ($p \geq \mathrm{N}$), all six constraints (2) − (7) must be considered. Thus, one of the *main drive power* or *maximum feedrate* constraints may replace the computed curve above defined, for intersection with the *maximum mechanical load* constraint (one of torsion, deflection or pitch load curves $\mathrm{C} \cdot w - n = 0$, depending on their current position during cutting).

4. EMBEDDED OPTIMAL CONTOURING

The optimization strategy for the milling process is based on two types of control actions:

1. Driving the working point $x = (w,n)$ towards the stationary one x_0 and maintaining it as close as possible to it (x_0 moves due to load disturbances i.e. to variations of material hardness or depth of material to be removed).
2. Bringing back the current working point into the allowable domain bordered by constraints (2)–(7) and the computed curve (12), whenever these seven constraints are violated.

Table 1 Constraint grouping in response to violation of the allowable working domain boundaries

Constraint class	Components	Controls actions at frontier violation
R_1: "max. chip load" type	(2), (3) and (4) right	$w \leftarrow w - \Delta w$ $n \leftarrow n + \Delta n$
R_2: "max.power" type	(7), (12) and (5) right	$w \leftarrow w - \Delta w$ $n \leftarrow n - \Delta n$
R_3: "unloaded" type	(4) left	$w \leftarrow w + \Delta w$ $n \leftarrow n - \Delta n$
R_4: "speed" type	(6)	$w = \mathrm{ct}$ and $n \leftarrow n + \Delta n$ or $n \leftarrow n - \Delta n$
R_5: "min. feed" type	(5) left	$w \leftarrow w + \Delta w$ $n = \mathrm{ct}$

The seven constraints are grouped in Table 1 in five classes R_i for corrective actions at frontier violation.

If more than one constraint is violated, the algebraic sum of the changes in w and n is taken to maintain the total change of the controls at the value of one increment Δw or Δn. If the working point is inside the allowable area, the control strategy increases the feedrate: $w \leftarrow w + \Delta w$, $n = \text{ct}$., which optimizes the throughput cost function and reduces simultaneously the cutting time.

The nonlinear optimization algorithm with multiple inequality-type constraints is implemented in discrete version at sample period time T_s *within each tool path* and uses four variables WINC, WDEC, NINC, and NDEC (increase/decrease w, n) to periodically update the references w, n of the control signals – feedrate and spindle speed starting from their initial values F, S *specified for each CNC-program phrase.*

The interlaced numeric-adaptive (CNC-ACO) cutting control for one tool path is realized in the sequence:

1. Set up new cutting depth for next tool path p_i.
2. Start tool path p_i; set up $w = F, n = S$ as initial control references from g-code.
3. Reset WINC, WDEC, NINC, and NDEC
4. Loop:
   - Continuously measure machine & process signals from internal sensors (w, n) and external ones: mechanical – M_d (or M_t) and electrical – U, I, $\cos\varphi$ over $T_{s,k}$ time intervals ($T_{s,k}$ corresponds to the variable time interval $16.(6)/n_k$ milliseconds during which one complete rotation of the cutting tool (mill) occurs, where n_k [rot/sec] is the updated spindle speed during the k^{th} sampling period within the current tool path).
   - Hold $M_d(t_k) = \max\{M_d\}|T_{s,k}$ and the values $w = w(t_k), n = n(t_k), U = U(t_k), I = I(t_k)$ with $t_k \in T_{s,k}$ the time moment when the maximum value of M_d was registered.
   - Compute pf, P and the w, n dependency (12)
   - Check the violation of constraints (2)-(6) and modify WINC, WDEC, NINC, and NDEC according to the corrective actions in Table 1.
   - Change control references w, n as follows:

$$w_{k+1} = \begin{cases} w_k + (\text{WINC} - \text{WDEC}) \cdot \Delta w, & \text{WINC} \neq \text{WDEC} \\ w_k + \Delta w, & \text{WINC} = \text{WDEC} \end{cases}$$

$$n_{k+1} = n_k + (\text{NINC} - \text{NDEC}) \cdot \Delta n \,,$$

5. … Until tool path p_i completely generated.

The embedded CNC-ACO milling control updates g-code data *F,S* by sensor data fusion and nonlinear constrained optimization. From the resource point of view, external sensors are added to measure the mechanical and electric load of the machine, and a processor-based hardware module computes $M_{t,\max}$, $w^{\alpha} n^{\beta}$, pf, P and updates the w, n optimization controls from initial CNC values F, N. n is directly fed to the main spindle motor drive, and w is decomposed by the CNC trajectory generator into the interpolated Cartesian components w_x, w_y.

5. CONCLUSIONS. EXPERIMENTAL RESULTS

Experiments have been carried out in an information processing chain starting with the creation of a depth map image of 3D complex model surface (a F1 car body) using a FARO laser scanner – short range finder device, then generating iso-level tool paths with equal cutting depth as closed contours in the binarized negative depth map image (Fig. 8), and finally creating the finishing tool paths as iso-parametric *XY* trajectories (Fig. 9).

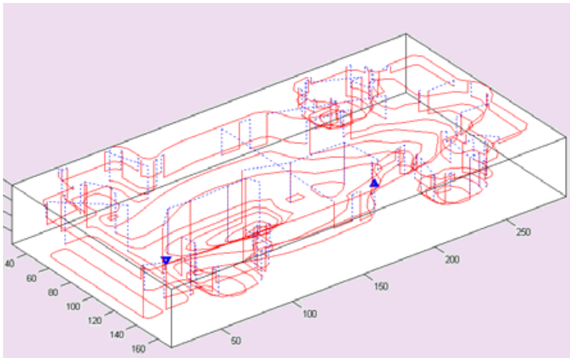

Fig. 8. Roughing stage with iso-level toolpaths.

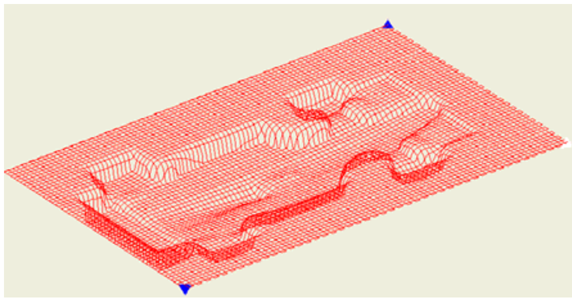
Fig. 9. Finishing stage using iso parametric tool paths in the XY plane.

The g-code was executed on an EMCO vertical mill, with torsion torque measurement and ACO control, increasing by 21% throughput relative to fixed *F,S*.

REFERENCES

Braswell, R. (1972). Inequality Constrained Optimal Milling, *Int. J. System Science*, **3**, *3*, 43-50.

Milner, D.A. (1974). Adaptive Control of Feedrate in the Milling Process, *Int. J. Mach. Tool Des. Res.*, **14**, *2*, pp. 65-72.

Park, S. (2005). Associating a Design Model and its Machining Process Plan, *Computer-Aided Design & Applications*, **2**, *5*, pp. 665-673.

Shaw, M.C. (1986). *Metal Cutting Principles*, Chapter 19, Clarendon Press, Oxford.