15th Triennial World Congress, Barcelona, Spain

OPTIMAL CONTROL OF ROBOTIC MANIPULATOR FOR LASER CUTTING APPLICATIONS

A. Pashkevich[1,2], A. Dolgui[1] , O. Chumakov[2]

[1]*Industrial Systems Optimization Laboratory, University of Technology of Troyes*
12, rue Marie Curie B.P. 2060, Troyes, 10000, France
[2]*Robotic Laboratory, Belarusian State University, of Informatics and*
Radioelectronics, 6, P.Brovka St., Minsk, 220600, Belarus

Abstract**:** The paper focuses on the enhancement of automatic robot programming techniques for laser cutting application. Its particular contribution lies in the area of multiobjective optimization of robot motions via graph representation of the search space and the dynamic programming procedures. It have been developed algorithms that allow to generate smooth manipulator trajectories within acceptable time, simultaneously considering kinematic, collision and singularities constraints of the robotic system, as well as the limitations of the robot controller. The efficiency of the algorithms has been carefully investigated via computer simulation and verified for real-life applications in automotive industry. *Copyright © 2002 IFAC*

Keywords: robot off-line programming, path planning, multiple-criteria optimization.

1. INTRODUCTION

In last decades, laser machining has gained essential industrial acceptance as an alternative to mechanical processing. It has significant advantages over traditional production methods due to its high process quality combined with high speed, high precision and potential flexibility (Geiger and Otto, 2000). However, the *manual teaching* of robotic laser systems is a very tedious and time-consuming. In contrast to them, the *off-line programming* allows generating the control code by means of computer graphics and away from the factory floor. As the result, the down time for which a robot is out of production may be reduced by 80÷85%, enabling very small batch sizes to become economic.

At the moment, there are a number of robot off-line programming systems on the market. Some of the most common of them are RobCAD (Tecnomatix Technologies), IGRIP (Deneb Robotics), CimStation (Silma) and Workspace (Robot Simulations). They implement a number of sophisticated path-planning methods, however there still exists a considerable gap between their capabilities and requirements of a particular technology. And up to now, the robot programs for some cutting applications are constructed interactively.

For laser cutting applications, the main contribution in off-line programming has been done by M.Geiger and his co-workers (University of Erlangen-Nuremberg, Germany). However, the proposed techniques may be applied only to non-redundant kinematic structures, which are based on five-axis robots. This paper focuses on enhancement of the 3D off-line programming techniques for six-axis robots, which possess inherent redundancy with respect to the cutting. In contrast to the known methods, the proposed approach takes into account this *redundancy* in combination with kinematic, collision and singularities constraints of the robotic system, as well as *limitations of industrial control units.* It relies on simultaneous optimization of multiple criteria and allows generating smooth manipulator trajectories within acceptable for industrial application time.

2. PROBLEM STATEMENT

2.1. General Optimization Problem

Let us assume that input data for the motion planning system are presented by two vector functions

$$\left\{\boldsymbol{p}(t),\boldsymbol{n}(t):\ \ \left|\boldsymbol{n}(t)\right|=1;\ \ t\in[0;T]\right\} \qquad (1)$$

where t is a scalar argument (time); $\boldsymbol{p}(t)\in\boldsymbol{R}^3$ defines (x,y,z)-coordinated of the tool tip, and $\boldsymbol{n}(t)\in\boldsymbol{R}^3$ is the unit vector of the tool axis orientation, which must be normal to the processing surface (Fig. 1). To describe spatial location of the robotic tool, let us introduce another unit vector $\boldsymbol{a}(t)$ which is tangent to the workpiece surface and points the tool motion direction. Assuming that the vectors $\boldsymbol{a}(t)$ and $\boldsymbol{n}(t)$ are mutually orthogonal, an each point of the processing contour may be associated with the coordinate frame which *X-axis* is directed along the path, *Z-axis* is directed along the cutting tool, and *Y-axis* completes them to the right hand oriented triple.

The corresponding matrix of homogenous transformation $\boldsymbol{H}(t)$ can be used for defining complete pose of the robotic tool, which requires 6 independent parameters (3 Cartesian coordinates and 3 Euler angles). However, for the cutting technology, five parameters are sufficient because the tool axially symmetry. Hence, the cutting tool locations $\boldsymbol{L}$ can be defined accurate to rotation around the vector $\boldsymbol{n}$

$$\boldsymbol{L}(t,\gamma)=\left[\begin{array}{c:c}\boldsymbol{R}_n(\gamma)_{3\times3} & 0\\ \hdashline 0\ 0\ 0 & 1\end{array}\right]\cdot\boldsymbol{H}(t), \qquad (2)$$

where $\gamma\in(-\pi;\pi]$ is the rotation angle and $\boldsymbol{R}_n(\gamma)$ is the corresponding 3×3 orthogonal rotation matrix around the vector $\boldsymbol{n}$.

Therefore, the robotic task description includes one undetermined parameter γ (i.e. one redundant degree of freedom) that can be used for optimization purposes. Indeed, the technological tool can be rotated around the laser beam axis without any influence on the quality of processing, provided that this motion does not contradict to robot kinematic and collision constraints. The latter are defined by binary functions $\Psi_k(\boldsymbol{L})$ and $\Psi_c(\boldsymbol{L})$ which non-zero values correspond to the constraint violation. In addition, to insure singularity-free motion of the manipulator, let us define another binary function $\Psi_s(\boldsymbol{L})$ which zero value defines admissible distance to singularities. (Yoshikawa, 1985). So, the considered problem of the robot motion planing can be stated as follows:

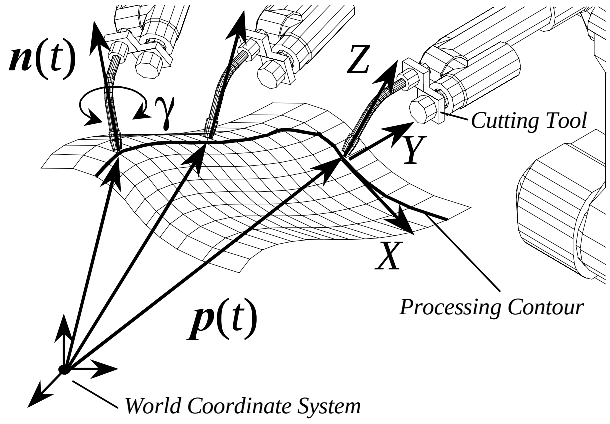

Fig. 1. Defining task frames

Design Problem. *For given manipulator task described by parameterized homogenous matrix-function* $\boldsymbol{L}(t,\gamma)$, $t\in[0;T]$, *find a scalar function* $\gamma(t)\in(-\pi;\pi]$ *which defines continuos sequence of feasible tool locations* $\boldsymbol{L}(t,\gamma(t))$ *and minimizes given performance measure*

$$J\left\{\boldsymbol{L}(t,\gamma(t));\ \ t\in[0;T]\right\}\rightarrow \min_{\gamma(t)} \qquad (3)$$

subject to kinematic, collision and singularities constraints

$$\Psi_k[L(t,\gamma(t))]=0;\ \ \Psi_c[L(t,\gamma(t))]=0;\ \ \Psi_s[L(t,\gamma(t))]=0.$$

Geometrical interpretation of this problem may be presented as searching for the best path in the plane that avoids prohibited regions indicating constraint violations. It should be noted however that in spite of the apparent similarity with mobile robot path planning (Latombe, 1991), the considered problem essentially differs by objective functions.

2.2. Performance Measures

For typical industrial robot, which possesses six degrees of freedom, the mapping from the task space $\{\boldsymbol{L}\}$ to the joint variable space $\{\boldsymbol{Q}\}$ is described by the inverse kinematic function

$$\boldsymbol{Q}=InvKin(\boldsymbol{L},M), \qquad (4)$$

which is parameterized by the configuration index M that allows to resolve a non-uniqueness problem. Therefore, the mapping from the task space to the joint variable space defines several self-motion manifolds

$$\boldsymbol{Q}(t,\gamma,M)=InvKin[\boldsymbol{L}(t,\gamma),M];\quad t\in[0,T], \qquad (5)$$

that must be considered separately during optimization. In addition, let us define the similar mapping

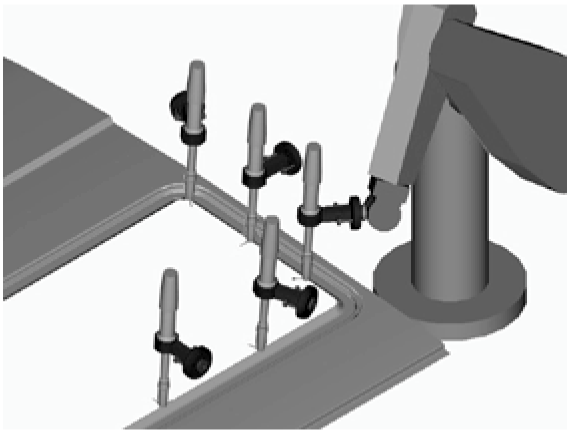

Fig. 2. Smooth control of the tool orientation

for the tool orientation angles

$$\boldsymbol{\Phi}(t,\gamma,M) = ToolAng[\boldsymbol{L}(t,\gamma),M]; \quad t \in [0,T], \quad (6)$$

which particular meaning depend on a convention adopted by a robot manufacturer. For similarity, the orientation angles are denoted as φ_1, φ_2, φ_3.

Thus, for given M, the function $\gamma(t)$ defines six joint trajectories $q_k(t)$ each of which may be evaluated by the following cost functionals:

- Joint coordinate range

$$J_\Delta^{(k)}[\boldsymbol{q}(t)] = \max_t [q_k(t)] - \min_t [q_k(t)], \quad (7)$$

- Joint coordinate deviation

$$J_d^{(k)}[\boldsymbol{q}(t)] = \max_t |q_k(t) - r_k|, \quad (8)$$

- Joint coordinate displacement

$$J_s^{(k)}[\boldsymbol{q}(t)] = \int_0^T |\dot{q}_k(t)|\, dt, \quad (9)$$

- Joint maximum speed

$$J_v^{(k)}[\boldsymbol{q}(t)] = \max_t [\dot{q}_k(t)], \quad (10)$$

It is obvious that mapping from the task space to the tool orientation space, which yields three trajectories $\varphi_1(t),\ldots\varphi_3(t)$ may be also evaluated applying the same performance measures: tool angle range, deviation, displacement, and maximum speed.

Geometrical meaning of these functionals is the following. The range evaluates width of the smallest tube that contains the corresponding function. The deviation shows the bias of this tube relative to the prescribed value. The displacement characterizes the total amount of joint motions (without regards to the motion direction). And, finally, the maximum speed estimates function smoothness.

2.3. Optimizing multiple objectives

As follows from the previous Section, it not possible to describe completely the considered design requirements by a single objective. Though in ideal (and obviously "utopian") case all of the introduced objectives tend to zero, minimizing one of the component may degrade performance in another. So, the designer must choose one of the techniques that are usually used to balance multiple criteria (Steuer, 1986; Pamanes and Zeghloul, 1991).

In this paper, instead of giving preference to a particular objective or optimization technique, it is proposed to leave the final decision for the design stage. It may be chosen from the following options: (i) defining priority of partial objectives or the primary objective; (ii) applying minimax technique, i.e. the worst-case optimization; (iii) assigning weights to combine multiple criteria in linear function. Independent of the chosen technique, the vector-optimization engine must include scalar-optimization routines that are developed in the following Sections.

3. SEARCH SPACE REPRESENTATION

Let us transform the search space into a directed graph. For the considered problem, the given path may be described by an evenly distributed sequence of nodes $\{\boldsymbol{p}_i, \boldsymbol{n}_i\}$ with equal sampling step ΔS. Similarly, the interval of the redundant parameter $\gamma \in (-\pi,\pi]$ may be divided in m segments. So, after extraction of only those locations $\boldsymbol{L}[t,\gamma(t)]$ that satisfy the *kinematic, collision and singularities constraints*, each node of the path (1) can be mapped into a set of tool locations and corresponding joint coordinates

$$\{\boldsymbol{p}_i, \boldsymbol{n}_i\} \quad \rightarrow \quad \left\{ \begin{array}{cccc} L_{i1}, & L_{i2,} & \ldots & L_{im} \\ \hdashline Q_{i1}, & Q_{i2}, & \ldots & Q_{im} \end{array} \right\}. \quad (11)$$

Therefore, the feasible search space can be represented by a multi-layer directed graph (Fig. 3) with vertexes $V = \{\boldsymbol{L}_{ij}\}$ and edges $E = \{(\boldsymbol{L}_{ij}, \boldsymbol{L}_{kl}) | i = k-1\}$. And the robotic path-planing task is reduced to the following network optimization problem.

Transformed Design Problem. *For given set of vertexes V and set of edges E, find the "best" path* of length n

$$\Pi(\gamma_0,\ldots\gamma_n) = \left\langle \boldsymbol{L}_{0j_1} \rightarrow \boldsymbol{L}_{1j_2} \rightarrow \ldots \boldsymbol{L}_{nj_n} \right\rangle \quad (12)$$

with initial state $V_0 \in \{L_{0j}\}$ *and final state* $V_0 \in \{L_{nj}\}$, *which minimizes the specified performance index*.

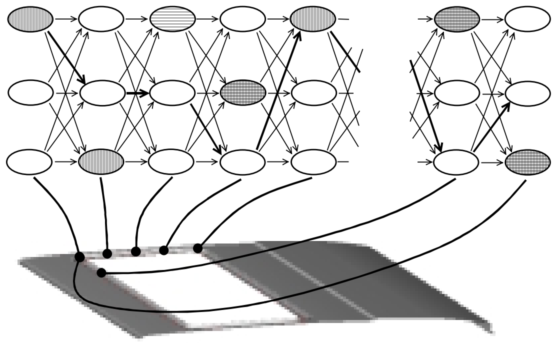

Fig. 3. Graph representation of the search space

It should be stressed that in this formulation both the initial and final states are not unique, but the problem can be transformed to the classical one by adding virtual start and end nodes (common for all layers).

4. GENERATION OF OPTIMAL PATH

Since the considered performance measures differ by their properties (additive, non-additive, etc.), the optimization technique should be also different. In this section, there are proposed several algorithms that minimize the performance measures (7)-(10) in acceptable time. To simplify description of the algorithms, the joint coordinates corresponding to the location $\boldsymbol{L}_{ij}$ are denoted as $q_k(i,j)$, and the trajectories corresponding to the solution vector Γ are denoted as $q_k(i,j_{\gamma_i})$. The algorithms are equally applicable for optimization in both the joint variable space $\boldsymbol{q}$ and the tool orientation space $\boldsymbol{\varphi}$, while the description below is given for the first case only.

4.1. Minimization of coordinate deviation

The optimization problem

$$J_d^{(k)}(\Gamma) = \max_i \left| q_k\left(i, j_{\gamma_i}\right) - r_k \right| \rightarrow \min_\Gamma , \qquad (13)$$

that minimizes deviation of the k-th joint variable with respect to the prescribed value r_k may be solved in a straightforward way, by selecting for each time instant t_i the value of $\gamma \in \Gamma_i$ that yields local minimum of the difference. It is obvious that such solution also ensures global optimum, though in general case it is not unique. However, using proposed multiobjective approach, the detected "*critical nodes*"

$$\left(i_r, j_r\right) = arg\left(\max_i \min_j \left| q_k\left(i, j\right) - r_k \right|\right) \qquad (14)$$

may be converted into constraints, which are taken into account on the next steps, while applying other optimization criteria. Within the proposed formulation, such transformation is performed by simple reduction of the set Γ_i up to a single element γ_{jo}.

4.2. Minimization of coordinate range

The optimization problem

$$J_\Delta^{(k)}(\Gamma) = \max_i \left[q_k\left(i, j_{\gamma_i}\right)\right] - \min_i \left[q_k\left(i, j_{\gamma_i}\right)\right] \rightarrow \min_\Gamma , \qquad (15)$$

may be solved by simultaneous application of the previous algorithm and non-linear optimization technique. The problem can be reduced to seeking for the best-prescribed value r_k that yields minimum of the corresponding deviation:

$$f^{(k)}(r_k) = \max_i \left[\min_j \left|q_k\left(i, j\right) - r_k\right|\right] \rightarrow \min_{r_k} . \qquad (16)$$

In this case, the value r_k is treated as the middle of the coordinate range, so the optimal solution r_k^{o} gives two "*critical nodes*" that correspond to the upper and the lower levels respectively. Similar to the previous case, the optimal solution is not unique, so the critical nodes may be also converted into constraints for the next optimization steps.

It should be stressed that because of non-smooth and poly-modal nature of the objective function, conventional nonlinear optimization methods (step descent or gradient search, for instance) can not be used here. Alternative approach is based on sophisticated random search techniques, *simulating annealing* in particular.

4.3. Minimization of coordinate increment

For the discrete representation of the search space, the coordinate velocity is estimated by the finite difference computed for the successive time instants. So, the related optimization problem is stated as

$$J_v^{(k)}(\Gamma) = \max_i \left| q_k\left(i, j_{\gamma_i}\right) - q_k\left(i-1, j_{\gamma_{i-1}}\right)\right| \rightarrow \min_\Gamma , \qquad (17)$$

and can be solved by means of the dynamic programming. To prove it, let us assume that at the p-th stage there have been found all optimal sequences $\Gamma^{o}(p,\chi) = \langle \gamma_0, \gamma_1, ..., \gamma_{p-1}, \chi \rangle$, with the last element $\chi \in \Gamma_p$ and the corresponding performance measures are denoted as $F_p(\gamma)$. Then, for the next stage, the sequence $\Gamma^{o}(p+1,\gamma) = \langle \gamma_0, ..., \gamma_{p-1}, \chi, \gamma \rangle$ with the last element $\gamma \in \Gamma_{p+1}$ may be found from the following recursion

$$F_{p+1}(\gamma) = \min_{\chi \in \Gamma_p} \max\left\{F_p(\chi), \left|q_k\left(p+1, j_\gamma\right) - q_k\left(p, j_\chi\right)\right|\right\}$$

Therefore, staring from p=1 and sequentially increasing length of the sequence, for each "end state" there can be find both the optimal path and the corresponding value of the performance measure. It is obvious that a similar approach can be also applied to minimization of the weighted sum and the "worst" component of the corresponding vector performance measure.

4.4. Minimization of coordinate displacement

Using discrete search space, this optimization problem is stated as follows:

$$J_s^{(k)}(\Gamma) = \sum_i \left| q_k\left(i, j_{\gamma_i}\right) - q_k\left(i-1, j_{\gamma_{i-1}}\right) \right| \to \min_{\Gamma} \quad (18)$$

In contrast to the previous case, it is an additive performance measure that is accumulated along the path. Therefore, it can be also minimized applying the dynamic programming. Using the notation adopted in the previous Section, the corresponding recursion can be written as

$$F_{p+1}(\gamma) = \min_{\chi \in \Gamma_p} \left\{ F_p(\chi), + \left| q_k\left(p+1, j_\gamma\right) - q_k\left(p, j_\chi\right) \right| \right\}$$

So, sequentially increasing length of the sequence $\Gamma^{o}(p,\gamma)$, for each "end state" there can be find both the optimal path and the corresponding performance measure. As in the previous case, the last step deals with the selection of the best "end state" from the set $\gamma \in \Gamma_n$. It can be easily proved that a similar recursion also yields an optimal solution for the weighted sum and the "worst" component of the corresponding vector performance measure.

5. SIMULATION RESULTS

To demonstrate the proposed technique, let us consider a three-link *RRR* planar manipulator with the parameters l_1=1.0, l_2=1.0 and l_3=0.25. The cutting contour is defined as a square with the side d=0.8, which angles are rounded with the radius r=0.10. The center of the contour is located at the point (1.0, 1,0) and is surrounded by an obstacle with the gap Δd=0.05. After the sampling, the contour is presented as a set of 60 uniformly distributed nodes. Using the inverse model and altering the tool orientation φ with the step of 10°, it has been generated a set of 1385 feasible tool locations $\{\boldsymbol{L}_{ij}\}$ and corresponding set of joint coordinates $\{\boldsymbol{Q}_{ij}\}$.

To investigate relative importance of the considered performance measures, firstly there were found optimal solutions for a single objective applied to a single coordinate q_1, q_2 or q_3 (see Tables 1 – 3). As follows from the results, the minimization of J_s (joint displacement) yields result that is also satisfactory for other objectives, so it may be chosen as the primary performance measure to present the engineering requirement of a "smooth" trajectory. However, further analysis shows that minimizing J_s for one joint may lead to very sharp profile for the remaining ones, especially for the third joint. Therefore, the competing objectives must be balanced by computing the weighted sum or the "worst" component of the vector criteria.

The simultaneous optimization of all joint trajectories shows that the weighted sum approach, as well as the "worst case" minimization, yield roughly the same results, which are also close to the result for minimization of $Js^{(3)}$. Therefore, in this particular case, the third joint may be considered as a "key" one and such solution may be chosen as the output of the multiobjective optimization process.

Table 1 Performance measures for optimization of q_1

Objective	$J_\Delta^{(1)}$	$J_v^{(1)}$	$J_s^{(1)}$
Minimum of range $J_\Delta^{(1)}$	19.09	2.69	56.56
Minimum of increment $J_v^{(1)}$	32.9	1.47	42.79
Minimum of displacement $J_s^{(1)}$	19.09	1.47	27.70

Table 2 Performance measures for optimization of q_2

Objective	$J_\Delta^{(2)}$	$J_v^{(2)}$	$J_s^{(2)}$
Minimum of range $J_\Delta^{(2)}$	41.32	3.31	98.78
Minimum of increment $J_v^{(2)}$	50.72	2.54	69.24
Minimum of displacement $J_s^{(2)}$	49.76	3.19	62.92

Table 3 Performance measures for optimization of q_3

Objective	$J_\Delta^{(3)}$	$J_v^{(3)}$	$J_s^{(3)}$
Minimum of range $J_\Delta^{(3)}$	11.50	10.69	161.8
Minimum of increment $J_v^{(3)}$	52.26	3.84	105.0
Minimum of displacement $J_s^{(3)}$	52.26	3.84	105.0

7. INDUSTRIAL IMPLEMENTATION

The algorithms developed here have been successfully implemented on the manufacturing floor, in ROBOMAX CAD package. It is already used in Russian automotive industry and has been successfully applied for design of manufacturing lines/cells for LADA cars, GAZEL lorries and ZIL mini-vans.

In application to the laser cutting technology, the Robomax/Laser (Fig. 4) allows to design the workcell layout and optimize robot motion using multiobjective optimization techniques. The main design procedure consists of three iteratively repeated steps. The first step is the selection of the proper manufacturing environment and locating them within the robot workspace. The second step deals with path planning using algorithms described in this paper. And at the third step, the obtained solution is verified using realistic simulation of the manufacturing environment. Recent application of the package is the off-line programming of a robotic cutting station for AMO ZIL (Moscow), which includes KUKA PR161 robot and corresponding positioning and clamping devices. It is used for a small-batch manufacturing, which requires frequent reprogramming.

SUMMARY

The developed technique allows generating optimal movements of robotic manipulators in 3D space taking into account its kinematic redundancy and particularities of the laser cutting technology. Incorporating these results in graphic simulation system leads to essential reduction of process planning time, enabling even very small batch sizes to become economic for the robotic processing.

Particular contribution of this paper deals with the multiobjective optimization of robot motions that is based on simultaneous optimization of performance measures for all joint coordinates. To generate smooth motion, each joint trajectory is evaluated by a set of performance indices such as the coordinate range, the deviation, the maximum increment, and the total displacement. The search space is converted into a directed graph and the problem is reformulated in terms of the combinatorial optimization theory. The optimal solution is obtained via the dynamic programming procedures that minimize weighted sum of the objectives (or the "worst" of them) and yield result within acceptable for industrial application time. During optimization, the weights are altered to generate a set of Pareto-optimal solutions.

ACKNOWLEDGEMENTS

The authors appreciate the financial support of the INTAS through the project 2000-757.

REFERENCES

Geiger, M. and Otto, A. (2000). *Laser in der Elektronikproduktion & Feinwerktechnik, Tagungsband des 3*. Erlanger Seminars LEF 2000, Bamberg: Meisenbach 2000.

Latombe, J.C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Dordrecht, Netherlands.

Pamanes, J.A. & Zeghloul, S. (1991). Optimal Placement of Robotic Manipulators Using Multiple Kinematic Criteria. *Proc. of the IEEE Int. Conference on Robotics and Automation*, Sacramento, California (pp. 933-938).

Sendler. U. (1994). Offline-programmiertes Laserschneiden im Automobilbau, *VDI-Z* 136, No. 1/2, (pp. 28-30).

Steuer, R. (1986). *Multiple criteria optimization: theory, computation and application*. John Wiley, New York.

Yoshikawa, T. (1985). Manipulability of Robotic Mechanisms. *Int. Journal of Robotics Research*, 4, No.2, (pp. 3-9).

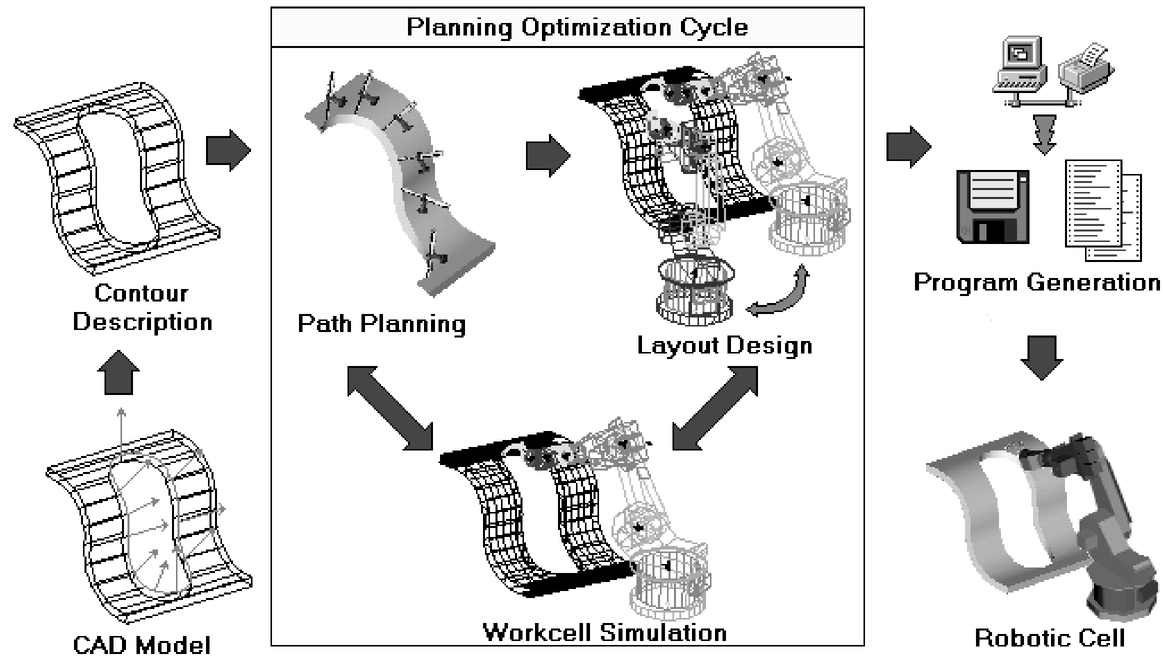

Fig. 4. Robotic cell design and programming using Robomax/Laser