PERIODICA POLYTECHNICA SER. EL. ENG. VOL. 43, NO. 3, PP. 189–198 (1999)

DESIGN AND EVALUATION ENVIRONMENT FOR COLLISION-FREE MOTION PLANNING OF COOPERATING REDUNDANT ROBOTS

Zsolt KEMÉNY

Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
H-1117 Budapest, Pázmány P. sétány 1/D, Hungary
e-mail: galaxy@impulzus.sch.bme.hu

Received: Nov. 9, 1999

Abstract

This paper deals with path planning methods suitable for use with closely cooperating kinematically redundant robots (primarily open-chain rigid-link manipulators) avoiding collision with segments and obstacles. A Matlab-based environment has been set up for designing such methods and evaluating already existing ones. Within this framework, several of the commonly used distance or intrusion criteria and corresponding path optimization methods have been examined for efficiency and reliability. Finally, proposals for further improvement of the methods are given.

Keywords: collision avoidance, cooperating robots, kinematic redundancy.

1. Introduction

With the spreading application of multi-agent systems, also problems related to cooperating robots receive more attention in research and development. One of the crucial issues hereof is collision avoidance, resulting from obstacles in the workspace and from cooperation itself.

A variety of systems can be considered cooperating open-chain robots, ranging from the fingers of a multi-fingered dextrous hand to a group of common industrial robots performing a task in cooperation. For the degrees of cooperation, following definitions can be given: *Loose cooperation* occurs when robots share the same workspace without further synchronization of their movement (e.g. different assembly tasks on the same object). *Close cooperation* (such as moving the same object by a number of robots), on the other hand, requires synchronous coordination of the end-effectors of the robots involved. Collision handling can range from actual collision avoidance according to an a priori model of the environment or sensor (e.g. vision) data to allowing surface contact up to a given force limit, but keeping the robot from further intrusion ('grazing' along object boundaries, requiring sensor-based reactive behavior). This paper is meant to focus primarily on collision *avoidance* for *closely cooperating* robots using preferably an *a priori model* of robot and obstacle boundaries. To complete the list of necessary definitions; kinematic redundancy occurs if the robot's joint space is of higher dimension than the workspace in which the task is defined. Since in this case, not only one solution of the inverse kinematic problem exists, an optimal solution can be chosen for given criteria.

A motion planner for collision-free close cooperation should return a path which keeps the robots from colliding with each other or the obstacles while performing dynamically or kinematically preferable motion, or, if this is not possible, it should give notice of its failure. The most important aspects for such algorithms are: energetic efficiency (within the limits of the model used), guarantee of collision-free motion (with regard to reliability of collision detection and avoiding 'false alarm') and computational cost (affecting for example real-time applicability). A number of algorithms are only suitable for robots with certain kinematic properties (such as kinematic redundancy, as dealt with in further parts of this paper), others may have mathematical properties which have to be taken into account in their practical application (e.g. susceptibility to local extrema).

2. Previous Work

A variety of approaches have been taken in the currently available methods. One of the most widespread concepts is to use a map related to significant configurations of the robot. This is either a search graph or a set of weighted areas. A graph-based method of this kind was for example presented by CHERIF and GUPTA [1] for a dextrous robot hand, the key issue, however, not being collision avoidance. SZCZERBA et al. [7] presented an efficient search algorithm for weighted regions, originally for mobile platforms, but not excluding possibilities of application for manipulators. Using a map-based approach, one has to be however conscious about the fact that even if not the entire search graph is constructed for a given task, the increasing dimensionality of the joint space requires much memory and computational effort.

The other major approach can be called 'objective-function-based.' This means that no specific search graph or map is constructed in the beginning; instead, an objective function for the measure of collision is defined which is then used either as a constraint or as a cost function during the actual calculation of the path with the latter ranging from a 'straightforward' planning of motion steps to an iterative modification of discrete path points. For this optimization procedure, also further cost functions are considered to provide the desired energetic efficiency. In this paper, some already existing methods will be shown and evaluated in the context of close cooperation and kinematic redundancy.

In recent years, also completely different ways of approach were taken, such as a method based on distributed artificial intelligence in the case of the hand-eye system JANUS (RICHTER et al. [4]).

3. The Issues of Objective-Function-Based Path Planning

As stated before, this approach consists of finding a measure equivalent to distance or intrusion of robot(s) and object(s) and using it either as a constraint or as a cost function to be minimized or maximized during the path calculation which is then a (constrained) optimization problem. Since multiple robots are involved in the cooperation, this also deserves special attention. One way of coping with this problem is to follow a master–slave principle (determining a motion step for a master robot while the others follow this motion) or considering all robots as one unified problem (apparently exploiting the presence of multiple robots more thoroughly).

3.1. The Distance Measures

The distance or intrusion measures can be subdivided into three major groups: point to point distance, point into body intrusion and body into body intrusion. An example for the first case is a (weighted) sum of distances between configuration control points (CCP) attached to the robot segments and obstacles, as proposed by SEZGIN [6] named MXDC (Maximum Distance Criterion):

$$\mathrm{MXDC} = \sum_{i=1}^{I} \sum_{l=1}^{L_i} \sum_{j=1}^{I} \sum_{k=1}^{K_j} \|\mathbf{p}_{i,l} - \mathbf{p}_{j,k}\| + \sum_{i=1}^{I} \sum_{l=1}^{L_i} \sum_{o=1}^{O} \|\mathbf{p}_{i,l} - \mathbf{p}_o\|, \tag{1}$$

where $\mathbf{p}_{i,l}$ denotes the coordinates of the lth CCP on the ith robot, while $\mathbf{p}_o$ stands for the oth CCP assigned to an obstacle. In this case, no actual collision is detected and therefore, the path optimization uses this sum of distances as an objective function to be maximized, and not as a constraint. *Fig.1* shows a pair of robot skeletons with configuration control points whose elementary distances are summed up. A simple example of a point-into-body intrusion measure was shown by SCHLEMMER and GRUEBEL [5]. In this case, a set of bounding ellipsoids and potential collision points are assigned to each body taken into consideration (*Fig.2*). For the calculation of one elementary distance between a collision point of one body and a bounding ellipsoid of another, a simple quadratic function derived from the ellipsoid's definition can be used which returns a negative value if the point is inside the ellipsoid, zero if it is on the boundary and a positive value if it is outside the ellipsoid, according to following scheme:

$$d = \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 - 1\,, \tag{2}$$

where a $(000)^T$-centered ellipsoid with main half-axis lengths a, b, c and a point $(xyz)^T$ are tested for the sake of simplicity. If a pair of objects is tested for distance/collision, a minimum of these elementary distances is calculated; for the case of more than two bodies, again the minimum of these is taken. In this case, the resulting function can already be included into a constraint (e.g. the requirement of it being greater than zero), and if a smooth gradient is required, a smooth approximation for the minimum can also be given.

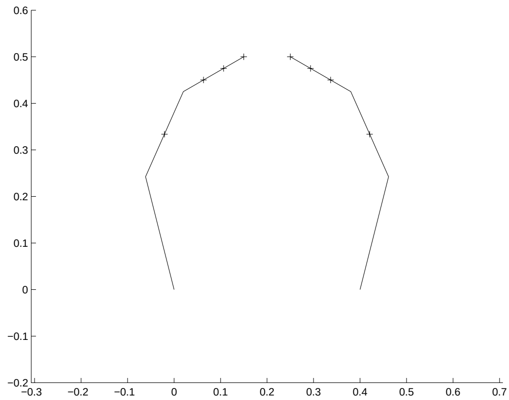

Fig. 1. CCPs on robot skeletons

The third case, a body-into-body intrusion is apparently the most demanding concerning computational cost. The approach presented by WANG et al. [8] approximates the boundaries of the objects with convex polyhedra (*Fig.* 3) and then calculates a body-into-body intrusion or distance for each selected pair of objects. First, a direction of projection is determined, then the minimal distance or maximal intrusion parallel to this direction is sought, as proposed by PREPARATA and SHAMOS [3]. Since this requires a large number of distance evaluations, the search space is kept as limited as possible. In a first step, the projected (planar) images of the polyhedra are checked for intersection, then the minimal distance is only searched for within this intersection, using an efficient two-level binary search technique. Also here, the minimum of distances for a given pair of bodies is taken if more than two are involved. Since these distances between polyhedra are not smooth and the search technique does not allow the straightforward application of a smooth approximation, robust optimization methods should be used with path planning.

3.2. Path Optimization

To plan a collision-free path which also optimizes other criteria, two approaches can be taken. In the first approach, a preliminary path is generated in the form of discretized points in configuration space. These points can then be modified to suit the constraints of the optimization problem (collision avoidance itself, velocity and acceleration limits etc.) and to optimize a given cost function (time, the norm of actuator forces etc.). The complete evaluation of the path is then performed taking the points themselves and interpolated path sections between them into account.

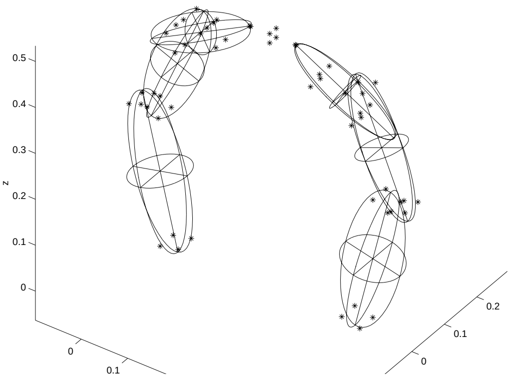

Fig. 2. Ellipsoidal representation

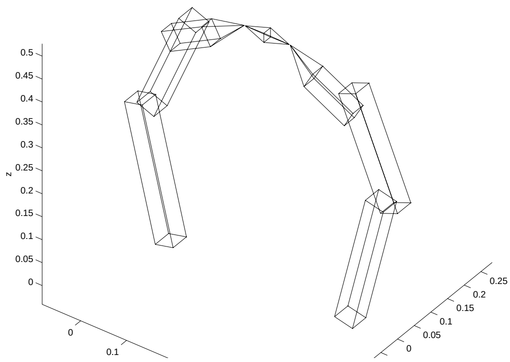

Fig. 3. Representation by convex polyhedra

An example for such optimization was presented by SCHLEMMER et al. [5] for use with an ellipsoid-based body/point intrusion measure as one of the constraints:

$$\min_{\mathbf{u}}\{t_f\} \text{ with:}$$

$$\begin{aligned}
\mathbf{u} &= \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}), && \\
\mathbf{q}(t_0) &= \mathbf{q}_0, & \dot{\mathbf{q}}(t_0) &= \dot{\mathbf{q}}_0, \\
\mathbf{q}(t_f) &= \mathbf{q}_f, & \dot{\mathbf{q}}(t_f) &= \dot{\mathbf{q}}_f, \\
d(\mathbf{q}) &\geq 0, && \\
\mathbf{q}_{\min} \leq \mathbf{q} &\leq \mathbf{q}_{\max}, & \dot{\mathbf{q}}_{\min} \leq \dot{\mathbf{q}} &\leq \dot{\mathbf{q}}_{\max}, \\
\ddot{\mathbf{q}}_{\min} \leq \ddot{\mathbf{q}} &\leq \ddot{\mathbf{q}}_{\max}, & \mathbf{u}_{\min} \leq \mathbf{u} &\leq \mathbf{u}_{\max},
\end{aligned} \tag{3}$$

where the required time t_f is minimized ($\mathbf{u}$ is the generalized torque vector, $\mathbf{H}$ and $\mathbf{h}$ denote the inertia and the Coriolis, centrifugal, gravitational and frictional parts of the dynamic model, respectively). A similar optimization problem is defined by WANG et al. [8], although not the required time of the motion, but a quadratic term of the generalized torque vector is minimized:

$$\min_{\mathbf{q},\dot{\mathbf{q}},\mathbf{u}} J(\mathbf{u}) = \int_0^T \frac{1}{2}\mathbf{u}^T\mathbf{R}\mathbf{u}\,\mathrm{d}t. \tag{4}$$

An interesting method of this group is the 'elastic band' as introduced by KHATIB et al. [10]. This concept uses a path in configuration space which consists of discretized points referred to as 'particles.' To each particle, a 'bubble' is assigned with its radius expressing the minimal distance to an obstacle, so that subsequent bubbles intersect and thereby ensure a collision-free path. The particles of a path can be moved by artificial forces (thus, they are applied to the path and not to the robot itself, as opposed to usual local motion planning methods), giving an optimal path when equilibrium of these forces is reached. The virtual forces are: i) internal contraction forces to ensure a minimum of motion for a given task, ii) external repulsive forces to keep the path away from obstacles and iii) constraint forces to prevent the particles from sliding along the path. This method can, if the forces are calculated during motion, be also applied in dynamically changing environments, and, since a shortest path in configuration space is sought, it is also advantageous for redundant robots.

The other major group of path optimization methods rather show a 'continuous' approach to the problem, even if the calculations themselves are, as usual, carried out in discrete steps (although with a finer discretization than customary in the previous group). Since this paper focuses on kinematically redundant robots, this class of methods will be also specific to them. As opposed to the previously mentioned discretizing path planners, in this case not only a workspace goal is defined, but the entire path in the workspace is prescribed. Given the end-effector motion to be performed, the kinematic redundancy can be utilized for a desired purpose; in this case for collision avoidance. An example for such an algorithm is presented by SEZGIN [6] where a sum of distances has to be maximized while moving the robot's end point along the desired path:

$$\begin{aligned} &\max_{\mathbf{q}} \mathrm{MXDC} \\ &\mathbf{F}_{\mathrm{kin}}(\mathbf{q}) = \mathbf{0}. \end{aligned} \tag{5}$$

One way of achieving this is solving the following set of equations simultaneously for each of the robots involved:

$$\begin{aligned} \left[(\mathbf{J}^{n-m})^T(\mathbf{J}^{m^T})^{-1} - \mathbf{I}_{n-m}\right]\cdot\mathbf{h} &= \mathbf{0}, \\ \mathbf{F}_{\mathrm{kin}}(\mathbf{q}) &= \mathbf{0}, \end{aligned} \tag{6}$$

where m is the workspace dimension, n is the joint space dimension, $\mathbf{J}^{n-m}$ represents the last $n-m$ rows and $\mathbf{J}^m$ the first m rows of the Jacobian, $\mathbf{h} = \partial MXDC/\partial\mathbf{q}$, and $\mathbf{F}_{\text{kin}}(\mathbf{q})$ stands for the robot's direct kinematics. Another way of solving the problem is to take the resolved motion approach:

$$\dot{\mathbf{q}} = \mathbf{J}^{+}\mathbf{v} + \alpha(\mathbf{J}^{+}\mathbf{J} - \mathbf{I}_n)\mathbf{h}, \tag{7}$$

where $\mathbf{J}^{+}\mathbf{v}$ with the Jacobian's Moore-Penrose pseudoinverse $\mathbf{J}^{+}$ ensures, as a particular part, the correct end-effector motion with minimal transients in the *kinematic* sense (it must be noted, that also a dynamically consistent pseudoinverse exists (see ŽLAJPAH [11]) which gives an *energetically* more favorable minimization of transients), while the rest of the solution is a linear combination of motions within the nullspace of $\mathbf{J}$ (homogeneous part) which can be used for collision avoidance. A reactive scheme with collision handling for a single redundant arm was presented by ŽLAJPAH [11] where null space motion is utilized for collision avoidance in a similar sense. A completely different approach is taken for redundancy resolution in the FSP (Full Space Parametrization) method by MORGANSEN et al. [9]. In this case, the entire solution space given by $\mathbf{J}$ is determined in form of base vectors and an additional linear constraint for weighting them. The base vectors can be acquired by selecting invertible square submatrices from the columns of $\mathbf{J}$ and multiplying their inverse by the prescribed task space velocity vector. To rebuild a vector matching the dimension of the joint space, zeros are inserted in the place of the columns not selected for a given square submatrix. Given a set of linearly independent base vectors of the solution space, the gradient of an objective function can be easily determined and the base vectors can be weighted according to its components, giving a solution which satisfies the kinematic requirements and optimizes a given cost function.

4. Test Results

To design new methods and to evaluate already existing ones, a set of Matlab functions, planned to be augmented to a toolbox complying with the Robotics Toolbox by CORKE [2], was produced (see also [12]). So far, the complete MXDC-method by SEZGIN [6] was realized and extensively examined in this framework, and two other distance/collision models (SCHLEMMER et al. [5], WANG et al. [8]) were implemented. The experimental setup for testing the resolved motion method by SEZGIN [6] and comparing the various distance measures, a pair of 3-DOF planar manipulators was chosen; the task to be performed was moving the end points of the manipulators to follow the ends of a rigid stick, simultaneously rotated and translated. *Fig. 4* shows a successful motion series planned by the MXDC method, and *Fig. 5* shows the objective function during the motion.

In the MXDC method, additionally to the original concept of SEZGIN [6], a scalar weighting factor can be assigned to each CCP, and the corresponding elementary distance is then weighted by the product of the two factors involved. This way, it is not only possible to apply points of attraction to regions known to be obstacle-free (as already suggested by SEZGIN [6]), but it is also possible to reduce the weights of less critical distances. During the test runs, the suggestion of SEZGIN [6] concerning CCP placement was found true; CCP's should advantageously be assigned to segments more prone to collision while CCP's on less relevant segments can unnecessarily restrict the motion. This also suggests that the weights of elementary distances could be changed during motion according to their relevance. Other tests, however, made the drawbacks of this computationally simple method clear. One major disadvantage is that no actual collision is detected. Furthermore, since only the first derivative of the objective function is used, local minima may cause serious problems, as do singular configurations as well. Since only null space motion is used for collision avoidance, the particular part using $\mathbf{J}^{+}$ cannot be 'switched off', even in critical situations where collision avoidance would have the highest priority. This drawback can be overcome by using the FSP method which allows a more elaborate (and even time-varying) weighting of different criteria. Altogether, the MXDC method, as presented by SEZGIN [6], should be used with much caution in its original form, since, although it is computationally not very demanding and thus suitable for real-time applications, it does not *guarantee* failure-free results.

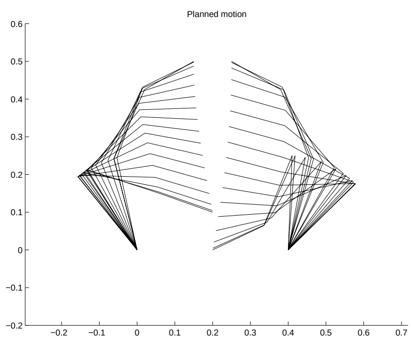

Fig. 4. Motion planned with MXDC

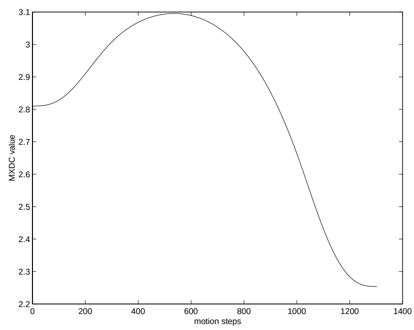

Fig. 5. MXDC value during motion

Figs. 6 and 7 show the two other distance measures implemented; the ellipsoidal approximation by SCHLEMMER et al. [5] and the polyhedral model by WANG et al. [8], respectively. Solid lines represent the minimal distance regarding the segments only, while dotted lines show the minimal distance including the object to be moved. The small values along the dotted line in *Fig.* 6 are caused by the proximity of the selected points of the object to the distal segments of the manipulators. This is not detected immediately in the case of polyhedra, since there, the distance is taken along a direction of projection and is thus infinite for objects whose projected images do not intersect. As it can be seen, the last motion steps signalize collision for the case of ellipsoids which, however, turns out to be false alarm, comparing it with the polyhedral distance values. From this we may conclude that an ellipsoidal approximation can give a wider safety margin; this is, however, not always true. In

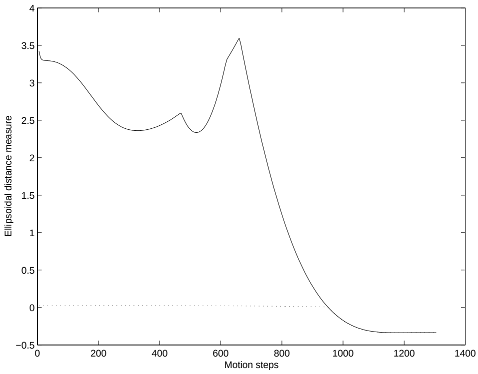

Fig. 6. Evaluation with ellipsoids

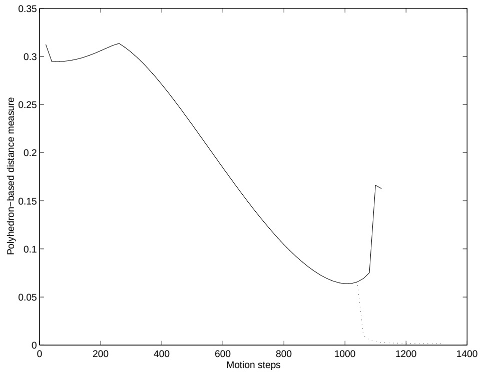

Fig. 7. Evaluation with polyhedra

fact, a point/body intrusion check may very well fail for non-planar arrangements if the potential collision points are selected too sparsely. A rule of thumb for the general case may be that the minimal density of selected collision points should be determined by the smallest main axis length of all ellipsoids involved.

As for the 'discretizing' optimization methods, it can be said that they are directly not suitable for closely cooperating robots. This results from the fact that even if it may be easy to keep the discrete points of joint space satisfying the constraints of close cooperation, the interpolated motion between them still needs to be compliant with the requirement of synchronized motion as well. This could be achieved either by a master–slave solution (choosing an 'arbitrary' interpolation for a master robot and following it with the other robots) or a 'global' solution. Finding such methods still remains a task for further research.

5. Conclusions

Several already existing motion planning methods for collision avoidance were shown, focusing on closely cooperating redundant manipulators, along with test results of the implemented algorithms. It could be seen that a guarantee of collision avoidance is necessary for failure-free operation. Furthermore, conclusions concerning setup and parameterization of collision measures could be drawn or verified. As for path optimization, it could be seen that roughly discretizing methods need further additions (mainly in terms of improved interpolation). Quasi-continuous approaches not using a preliminary path can also show different qualities of path planning, considering their treatment of the solution space. For path optimization methods, some suggestions for further improvement could be made.

Acknowledgement

Support for the research was provided by the Hungarian National Research Programs under grant No. OTKA T 029072 and FKFP 0417/1997.

References

[1] CHERIF, M. – GUPTA, K. K.: Planning Quasistatic Motions for Re-configuring Objects with a Multi-fingered Robotic Hand, subm. to *IEEE Transactions on Robotics and Automation*, 1998. (available at http://www.ensc.sfu.ca/people/faculty/kamal/journals.html).

[2] CORKE, P. I.: *Robotics Toolbox (Release 5) for Use with Matlab*, CSIRO, Division of Manufacturing Technology, Preston, Australia, 1999.

[3] PREPARATA, F. P. – SHAMOS, M. I.: *Computational Geometry*, Springer Verlag, New York Inc., 1985.

[4] RICHTER, G. – SMIEJA, F. – BEYER, U.: Integrative Architecture of the Autonomous Hand-Eye Robot JANUS, *Proceedings of the CIRA'97*, July 1997, Monterey, CA.

[5] SCHLEMMER, M. – GRUEBEL, G.: Real-Time Collision-Free Trajectory Optimization of Robot Manipulators with Semi-Infinite Parameter Optimization, *The International Journal of Robotics Research*, **17**, No. 9, September 1998, pp. 1013–1021.

[6] SEZGIN, U.: Utilization of Manipulator Redundancies for Collision Avoidance Strategies. PhD Thesis, King's College London, Mechanical Engineering Department, University of London, 1995. pp. 1–364.

[7] SZCZERBA, R. J. – CHEN, D. Z. – UHRAN, J. J., Jr.: Planning Shortest Paths among 2D and 3D Weighted Regions Using Framed-Subspaces, *The International Journal of Robotics Research*, **17**, No. 5, May 1998, pp. 531–546.

[8] WANG, D. – HAMAM, Y.: Optimal Trajectory Planning of Manipulators with Collision Detection and Avoidance, *The International Journal of Robotics Research*, **11**, No. 5, October 1992, pp. 460–468.

[9] MORGANSEN, K. A. – PIN, F. G.: Enhanced Code for the Full Space Parametrization Approach to Solving Underspecified Systems of Algebraic Equations, Version 1.0, Oak Ridge National Laboratory, Technical Report ORNL/TM-12816, March 1995.

[10] KHATIB, O. – QUINLAN, S. – WILLIAMS, D.: Robot Planning and Control, *Robotics and Autonomous Systems*, **21**, No. 3, September 1997, pp. 249–261.

[11] ŽLAJPAH, L.: On Obstacle Avoidance for Redundant Robots Using Tactile Sensors, *Proceedings of the 1999 IEEE International Conference on Intelligent Engineering Systems*, Nov. 1999, pp. 415–420.

[12] KEMÉNY, ZS. – LANTOS, B.: Design and Evaluation Environment for Collision-free Motion Planning of Cooperating Robots, *Proceedings of the 1999 IEEE International Conference on Intelligent Engineering Systems*, Nov. 1999, pp. 81–86.