15

A Manipulator Control in an Environment with Unknown Static Obstacles

Pavel Lopatin and Artyom Yegorov
Siberian state aerospace university named after academician M.F.Reshetnev
Russia

1. Introduction

In contemporary society robots and manipulators are used in different spheres of life. Robot should be as autonomous as possible and should effectively operate in a natural environment.

In the beginning of the robotic era, robots operated in a workspace which was free of obstacles. Later the works dedicated to the invention of algorithms for the control of robots in the presence of obstacles began to appear. There are algorithms which guarantee finding a trajectory in the presence of known obstacles, if such trajectory exists (Canny, 1988; Collins, 1975; Donald, 1985). Some authors use the artificial potential methods (see, for example, (Choset et al., 2005)). In this method a robot is represented by a point, regions in the workspace that are to be avoided are modeled by repulsive potentials and the region to which the robot is to move is modeled by an attractive potential. In a general situation there is no guarantee that a collision-free path will always be found, if one exists (Ahrikhencheikh & Sereig, 1994; Choset et al., 2005). There are different graph searching methods (Choset et al., 2005; La Valle, 1999-2004) which find a trajectory avoiding obstacles (even an optimal one), if it exists. It is easier to use such methods in the case where we have full information about free and forbidden points before the beginning of the movement. A computer may then calculate a preliminary trajectory and after that the manipulator may execute this trajectory. But in case of unknown obstacles the manipulator has to investigate its environment and plan its trajectory alternately. Then the difficulty arises that the graph searching algorithms demand to carry out in a certain volume a breadth-first search, otherwise the reaching of the target point q^T is not guaranteed (Ilyin,1995). But during the breadth-first search the following situation often arises: suppose we have just finished considering the vertices adjacent to a vertex q and we have to consider vertices adjacent to a vertex q' and the q and q' are not adjacent. In order to consider vertices adjacent to the q' the manipulator at first has to come to the q'. So we get a problem of the manipulator movement from q to q'. The necessity of searching and executing paths for multiple different q and q' makes the total sum of the manipulator movements very big (Ilyin,1995). In case we plan trajectory in known environment the computer simply switches its "attention" from q to q', which are stored in the computer's memory.

It is possible to outline the following representatives of the breadth-first approach: they are namely breadth-first searching algorithm, A* algorithm, best-first heuristic search, lazy PRM, dynamic programming (La Valle, 1999-2004). The methods based on randomized potential field, Ariadne's Clew algorithm, rapidly-exploring random trees (La Valle, 1999-2004) have such feature that new vertices are generated randomly and therefore using these methods for the unknown environment leads to the same difficulties. The approaches based on cell decomposition, visibility (bitangent) graphs, Voronoi diagrams (Choset et al., 2005; La Valle, 1999-2004) are reduced to alternate graph building and searching a path on it and have the above mentioned disadvantage connected with multiple mechanical movements.
In the algorithm presented in this article the vertices q and q' are always neighbor vertices and it reduces the number of movements.
For a solution of our problem it is possible to use the approach based on automatic theorem proving (Timofeev,1978), but this approach demands to consider large amount of variants and directions of the search and therefore the application of such approach is not effective (Yefimov, 1988).
It is also known that the "depth-first" algorithms do not guarantee reaching of a goal (Ilyin, 1995).
There is common difficulty for the methods for trajectory planning in the presence of known obstacles: it is very difficult to borrow full information about workspace of manipulator in advance and to represent this information in a form suitable for trajectory planning. Considering our algorithm one may see that there is no need for the control system to have full information about workspace in advance, manipulator will borrow necessary information by itself in limited quantities and in terms of generalized coordinates which is suitable for trajectory planning.
The attempts of creating algorithms for the robot control in presence of unknown obstacles were made. Most of them cover various two-dimensional cases (Lumelsky, 2006).
In (Chen et al., 2008; Ghosh et al., 2008; Masehian & Amin-Nasari, 2008; Rawlinson & Jarvis, 2008) different approaches for a robot control in a two-dimensional unknown environment are considered. In (Chen et al., 2008; Rawlinson & Jarvis, 2008) the approaches are based on Voronoi diagrams, in (Masehian & Amin-Nasari, 2008) a tabu search approach is presented. The approaches demand multiple robot movements. In (Masehian & Amin-Nasari, 2008) obstacles should have polygonal form. An application of methods proposed in (Chen et al., 2008; Ghosh et al., 2008; Masehian & Amin-Nasari, 2008; Rawlinson & Jarvis, 2008) to a n-link manipulator control in the unknown environment is not presented.
In (Lumelsky, 2006) an algorithm for the control of manipulators in the presence of unknown obstacles in three-dimensional space is given. Though this algorithm guarantees reaching of a target position it has such a limitation that the manipulator should not have more than three degrees of freedom.
In (Amosov et al., 1975; Kasatkin, 1979; Kussul & Fomenko, 1975) is described an application of semantic nets to the problem of robot control in unknown environment. The disadvantage of such approach is the preliminary teaching of the net, which simulates a planning system. The absence of formal algorithms for teaching makes impossible the teaching of a complex net, which should plan robot actions in an environment close to natural ((Ilyin, 1995).

In (Yegenoglu et al., 1988) the n-dimensional case is considered. The algorithm is based on the solution of the system of nonlinear equations using Newton method and therefore in cannot guarantee the reaching of a target position.
In (La Valle, 1999-2004) algorithms for moving a robot in the presence of uncertainty (including cases of unknown environment) are considered. The algorithms are based on the sequential decision theory. In general case the algorithms do not guarantee reaching the goal. In cases when the algorithms use searching on a graph the above mentioned difficulty arises connected with multiple mechanical movements.
In (Lopatin, 2001; Lopatin & Yegorov, 2007) algorithms for a n-link manipulator movement amidst arbitrary static unknown obstacles were presented. Algorithms guarantee reaching the q^T in a finite number of steps under condition that the q^T is reachable. It was supposed that the manipulator's sensor system may supply information about free and forbidden points either from a *r*-neighborhood of the manipulator's configuration space point where the manipulator is currently situated (Lopatin, 2001) or from *r*-neighborhoods of a finite number of points from the manipulator configuration space (Lopatin, 2006; Lopatin & Yegorov, 2007). In both cases it was supposed that the *r*-neighborhood has a form of a hyperball with a radius $r>0$. In this work we consider more general form of the *r*-neighborhood.

2. Task formulation and algorithm

2.1. Preliminary Information

We will consider manipulators which consist of n rigid bodies (called links) connected in series by either revolute or sliding joints (Shahinpoor, 1987). We must take into account that because of manipulator's constructive limitations the resulting trajectory $q(t)$ must satisfy the set of inequalities

$$a^1 \le q(t) \le a^2 \tag{1}$$

for every time moment, where a^1 is the vector of lower limitations on the values of generalized coordinates comprising $q(t)$, and a^2 is the vector of higher limitations.
The points satisfying the inequalities (1) comprise a hyperparallelepiped X in the generalized coordinate space. We will consider all points in the generalized coordinate space which do not satisfy the inequalities (1) as forbidden.
We will have to move the manipulator from a start configuration $q^0=(q_1^0, q_2^0, \ldots, q_n^0)$ to a target configuration $q^T=(q_1^T, q_2^T, \ldots, q_n^T)$. In our case the manipulator will have to move amidst unknown obstacles. If in a configuration q the manipulator has at least one common point with any obstacle then the point q in the configuration space will be considered as forbidden. If the manipulator in q has no common points with any obstacle then the q will be considered as allowed.
So, in our problem a manipulator will be represented as a point which will have to move in the hyperparallelepiped (1) from q^0 to q^T and the trajectory of this point should not intersect with the forbidden points.

2.2. Preliminary Consideratons

Let us make the following considerations:

1) The disposition, shapes and dimensions of the obstacles do not change during the whole period of the manipulator movement. Their number may not increase.

2) It is known in advance, that the target configuration is allowed and is reachable (that is, we know that in the generalized coordinates space it is possible to find a line connecting q^0 and q^T, and this line will not intersect with any forbidden point).

3) The manipulator has a sensor system which may supply information about r-neighborhoods of points $q^i \in X$, i=0.1,…, N, where N – is a finite number, defined by the sensor system structure and methods of its using.

The r-neighborhood of a point q^i is a set $Y(q^i)$ of points near q^i (see Figure 1).

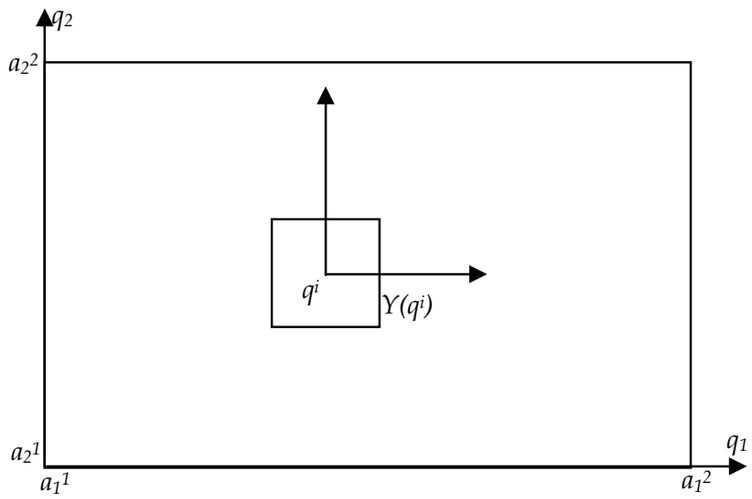

Fig. 1. An example of a r-neighborhhod $Y(q^i)$.

The $Y(q^i)$ has a shape of a hypercube. The q^i is situated in the centre of the hypercube. The length of the hypercube's edge may be arbitrary and is defined by the sensor system structure and methods of its using. All sets $Y(q^i)$, i=0,1,… should have the same size that is the length of an edge of a $Y(q^i)$ should be equal to the length of an edge of a $Y(q^j)$ for every i and j. The sides of a $Y(q^i)$, i=0,1,… should be parallel to the corresponding planes of the basic coordinate system of the generalized coordinate space. The sets $Y(q^i)$, i=0,1,… also should satisfy such condition that it should be possible to inscribe in the set $Y(q^i)$ a hyperball with the centre in the q^i and with a radius $r > 0$ (see Figure 2). The part of the $Y(q^i)$ comprising the hyperball should be compact that is the part should not have any hole. So, all points of the hyperball should belong to the inner part of the $Y(q^i)$.

The words "the sensor system supplies information about the r-neighborhood of a point q^i" mean that the sensor system tells about every point from the $Y(q^i)$ whether this point is allowed or forbidden. The sensor system writes all forbidden points from the $Y(q^i)$ into a set $Q(q^i)$, and writes into a set $Z(q^i)$ all allowed points from the $Y(q^i)$. The sets $Y(q^i)$, $Q(q^i)$, $Z(q^i)$ may be written using one of such methods like using formulas, lists, tables and so on, but we suppose that we have such method. We will not consider the sensor system structure.

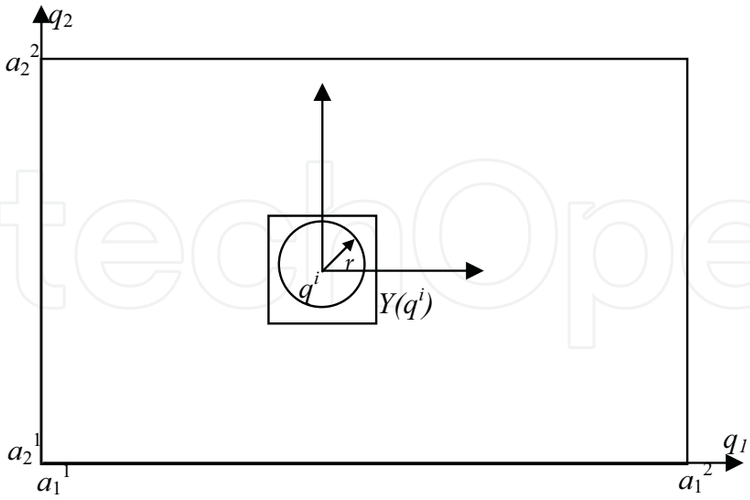

Fig. 2. It should be possible to inscribe in the set $Y(q^i)$ a hyperball with the centre in the $\boldsymbol{q}^i$ and with a radius $r > 0$

An r-neighborhood of q^i with the sets $Z(q^i)$ and $Q(q^i)$ may look as follows (Figure 3). Note that the sets $Z(q^i)$ and $Q(q^i)$ may be not continuous.

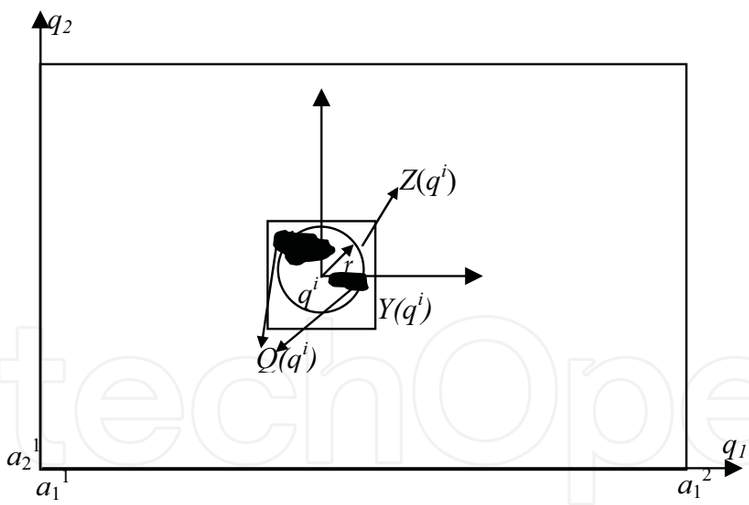

Fig. 3. An example of a r-neighborhood $Y(q^i)$ with the sets $Q(q^i)$ and $Z(q^i)$. The set $Q(q^i)$ is shown by a dark color. The points from the $Y(q^i)$ which do not constitute the $Q(q^i)$, belong to the $Z(q^i)$.

The considerations 1)-3) cover a wide range of manipulators' applications.

2.3. Algorithm for Manipulators' Control in the Unknown Environment

We will denote the points where generation of a new trajectory occurs as q^n, n=0, 1, 2,… We will call such points "trajectory changing points". Before the algorithm work n=0 and $q^n = q^0$.

Algorithm

STEP 1. The manipulator is in a point q^n, n=0, 1, 2,…, and its sensor system supplies information about the r-neighborhood of the q^n, and about the r-neighborhoods of points y^j, j=0, 1, …, Nn, $y^j \in X$ for every j=0, 1, …, Nn, Nn – is a known finite number. y^j, j=0, 1, …, Nn and the Nn are generally speaking different for every n and are told on every n to the sensor system before start of the sensor system functioning. So the sensor system supplies information about the $Q(q^n)$ and about the set

$$QS_n = \bigcup_{j=0}^{N_n} Q(y^j).$$

After that the manipulator generates in the configuration space a preliminary trajectory $L(q^n, q^T)$. The $L(q^n, q^T)$ should satisfy the following conditions: I) connect the q^n and the q^T; II) no point from the $L(q^n, q^T)$ coincides with any point from the sets $\bigcup_{i=0}^{n} Q(q^i)$ and $\bigcup_{i=0}^{n} QS_i$, in other words the preliminary trajectory should not intersect with any known forbidden point; III) satisfy the conditions (1).

The manipulator starts to follow the $L(q^n, q^T)$. The algorithm goes to STEP 2.

STEP 2. While following the $L(q^n, q^T)$ two results may happen:

a) the manipulator will not meet forbidden points unknown earlier and therefore will reach the q^T. Upon the reaching of the q^T the Algorithm terminates its work;

b) the manipulator will come to such a point (making at first operation n=n+1, let us define it q^n, n=1,2,…), that the next point of the preliminary trajectory is forbidden. The Algorithm goes to STEP 1. **End of Algorithm.**

2.4. Theorem and Sequences

Theorem. If the manipulator moves according to the Algorithm it will reach the target configuration in a finite number of steps.

Proof. Suppose that the manipulator being in q^n (n=0,1,…), generated a trajectory, leading to q^T and began to follow this trajectory. If the manipulator does not meet obstacles it will reach the target configuration in a finite number of steps (because the length of the trajectory is finite). Therefore, the endlessness of the manipulator wandering may be caused only by the endless repeating of the situation described in the item b) of STEP 2 of the Algorithm and therefore the endless generating of new trajectories may be caused by the two reasons:

1) the manipulator will infinitely return to the same point of the trajectory changing,
2) the number of points where it is necessary to change trajectory will be infinite.

Let us prove that all points where the manipulator changes its trajectory will be different. Suppose that the manipulator changed a trajectory being in a point q^s, and later it again changed a trajectory, being in a point q^p, that is $s<p$. Let us show that $q^s \neq q^p$. Suppose, at first, that, on the contrary, $q^s=q^p$. Then $Q(q^s)=Q(q^p)$. When the manipulator was in q^s, it generated a trajectory which did not intersect with the sets $Q(q^i)$, i=0, 1, ... , s. When the manipulator reached the point q^p, it discovered that it was necessary to change the trajectory that is this trajectory intersected with the set $Q(q^p)$. But $Q(q^p)=Q(q^s)$ and $Q(q^s)$ was taken into account

when this trajectory was generated. It means that the manipulator can not come to a point of the trajectory changing q^p which will be equal to any other point of the trajectory changing and it means that all points where the manipulator changes its trajectory are different.
Now let us show that the number of such points is finite. Suppose that it is infinite. All points of a trajectory changing must satisfy the inequalities (1). It means, that the sequence of these points is bounded. According to the Boltsano-Weierstrass theorem it is possible to extract from this sequence a convergent subsequence q^i, i=1,2,… According to the Cauchy property of the convergent sequences it is possible for any ε to find such a number s that all points q^i, $i>s$ will lie in an ε-neighborhood of q^s. Let us take $\varepsilon<r$. Consider an arbitrary point q^i of the trajectory changing lying in the ε-neighborhood of q^s. As far as in the q^i the manipulator had to change the trajectory, it means that that trajectory intersected with $Q(q^s)$ (because q^i and its neighbor points belong to $Q(q^s)$). From this fact it is possible to make the conclusion that the set $Q(q^s)$ was not taken into account when that trajectory was generated. But such situation is impossible if we strictly follow the conditions of the algorithm. The situation when a trajectory changing point belongs to the ε-neighborhood of another trajectory changing point will necessarily appear if the number of the points where trajectory is changing is infinite. But we showed that such situation is impossible and it means that a number of the points where it is necessary to change trajectory will be finite.
The theorem is proved.
So, it was shown that the number of the preliminary trajectory changes is finite, that the number of calls of the STEP 1 will be finite. Therefore one may see that the problem of a n-link manipulator control in an unknown environment under the considerations mentioned above is reduced to a solution of a finite number of problems of a trajectory planning in the presence of the known forbidden states.
Sometimes it is necessary to explore the environment more intensively. The following sequences will be useful for such cases.
Sequence 1. In a point q^d, d=0,1,…, N_{dn} of the trajectory $L(q^n, q^T)$, n=0,1,…, where N_{dn} – a finite number not bigger than the number of points in the $L(q^n, q^T)$, the manipulator may do a retreat from the $L(q^n, q^T)$. The retreat is a sequence S of allowed points continiously following one after another. The S is different from the $L(q^n,q^T)$ and satisfies the (1). The retreats may be done for the environment investigation. The number of steps on every retreat should be finite. The manipulator may investigate the environment on any step of a retreat and get information about the r-neighborhoods of points y^i, i=0,1,2…, N_{s1}, where y^i, i=0,1,2…, N_{s1} – arbitrary points from X and N_{s1} – a finite number arbitrary for every step. The number N_{s1} of points y^i and their disposition is defined by the structure of the sensor system and methods of its using. After making a finite number of steps on a retreat the manipulator should return to the $L(q^n, q^T)$ exactly by the points made on the retreat. If on some step of a retreat it is discovered that further movement is impossible because of forbidden states' presence the manipulator should return to the $L(q^n, q^T)$ exactly by the points of the retreat. After return to the $L(q^n, q^T)$ the manipulator continues movement according to the Algorithm. As a result we get a finite number of steps on all retreats which is added to the finite number of steps according to the Algorithm and in sum we get that q^T will be reached in a finite number of steps.
Sequence 2. The manipulator also may, being in a point q of a preliminary trajectory or retreat, get information about $Q(q)$ and, maybe, about $Q(y^i)$ i=0,1,2,…, N_{s2} , where y^i,

i=0,1,2,…, N_{s2} – arbitrary points from X, and N_{s2} – a finite number. After that operations n:=n+1 and q^n=q should be done and the manipulator may generate a new trajectory $L(q^n, q^T)$ satisfying the conditions I-III of STEP 1 of the Algorithm and begin to move along the new $L(q^n,q^T)$ according to the Algorithm. Let us call such action "refuse from a preliminary trajectory" that is it is such action when despite the q^T has not been reached a new trajectory leading to the q^T is generated and the manipulator begins to follow the new trajectory. It is possible to make only a finite number of refuses. Then the finite number of steps on refuses will be added to the finite number of steps according to the Algorithm and in sum we get that the q^T will be reached in a finite number of steps.

3. Using the polynomial approximation algorithm as a subroutine in the exact algorithm

3.1 Reducing the Algorithm to a finite number of calls of a subroutine for a trajectory planning in known environment

Every time when the manipulator generates a new trajectory according to the STEP 1 of the Algorithm, two cases may happen: either the manipulator will not meet an obstacle and therefore it will reach the q^T in a finite number of steps (because the length of the trajectory is finite) or the manipulator will meet an unknown obstacle and will have to plan a new trajectory. Therefore, in the Algorithm the problem of a manipulator control in the presence of unknown obstacles is reduced to the solution of a finite number of tasks of trajectory planning in the presence of known forbidden states. In other words, the Algorithm will make a finite number of calls of a subroutine which will solve the problem stated in STEP 1. In the rest of the article we will call this subroutine the SUBROUTINE. We took the polynomial approximation algorithm as algorithm for the SUBROUTINE.

3.2 Polynomial approximation algorithm

Denote the components of vector-function $q(t)$ as q_1, q_2, …, q^n. Write down the restrictions using new variables:

$$q_j(0) = q^0{}_j,\ q_j(1) = q^T{}_j,\ j = 1, 2, \ldots, n. \tag{2}$$

$$q^L{}_j \le q_j(t) \le q^H{}_j,\ j = 1, 2, \ldots, n. \tag{3}$$

Specify the obstacles by hyperspheres with centers ($q^p{}_{m1}$, $q^p{}_{m2}$, …, $q^p{}_{mn}$) and radius $r = \Delta q / 2 \cdot 1.1$, where $q^p{}_{mi}$ correspond to the components of vectors p_m, m = 1, 2, …, M, Δq is step of configuration space discretization. The value of the radius r is chosen so that if two obstacles are located on neighbor nodes of the grid and their centers differ only in one component, the corresponding hyperspheres intersect. Otherwise the hyperspheres don't intersect. Then the requirement of obstacles avoiding trajectory can be written as follow:

$$\sum_{i=1}^{n} (q_i(t) - q^p_{mi})^2 \ge r^2 \quad \forall t \in [0;1],\ m = 1,2,\ldots,M. \tag{4}$$

The left part of this inequality is squared distance between the trajectory point at moment t and the center of the m-th obstacle. We will search the trajectory in the form of polynomials of some order s:

$$q_j(t) = \sum_{i=0}^{s} c_{ji} t^i, \quad j = \overline{1,n} \tag{5}$$

Here c_{ji} are unknown coefficients. Substitute $t = 0$ and $t = 1$ in (5):

$$q_j(0) = c_{j0} + c_{j1} \cdot 0 + c_{j2} \cdot 0^2 + \ldots + c_{js} \cdot 0^s = c_{j0}$$

$$q_j(1) = c_{j0} + c_{j1} \cdot 1 + c_{j2} \cdot 1^2 + \ldots + c_{js} \cdot 1^s = c_{j0} + \sum_{i=1}^{s} c_{ji}, \text{ j} = 1, 2, \ldots, n.$$

Taking into account requirements (2):

$$c_{j0} = q^0{}_{j\,(5)} \tag{6}$$

$$\sum_{i=1}^{s} c_{ji} = q^T{}_j - q^0{}_j, \text{ j} = 1, 2, \ldots, n \tag{7}$$

Divide duration [0; 1] into K+1 pieces by K intermediate points t_1, t_2, …, t_K. Then the requirements (3) and (4) will be as follow:

$$\begin{gathered} \sum_{i=0}^{s} c_{ji} t_k^i \geq q_j^L, \quad j = \overline{1,n} \\ \sum_{i=0}^{s} c_{ji} t_k^i \leq q_j^H, \quad j = \overline{1,n} \\ \sum_{j=1}^{n} \left(\sum_{i=0}^{s} c_{ji} t_k^i - q_{mj}^p \right)^2 \geq r^2, \\ m = 1,2,\ldots,M, \quad k = 1,2,\ldots,K. \end{gathered} \tag{8}$$

Thus, it is necessary to find such coefficients c_{ji} (j = 1, 2, …, n, i = 1, 2, …, s) which satisfy the system of equations (6), (7) and inequalities (8). Obviously, the coefficients c_{j0} are easily found from the equations (6). Express the coefficients c_{js} from the equations (7):

$$c_{js} = q_j^T - q_j^0 - \sum_{i=1}^{s-1} c_{ji}, \quad j = \overline{1,n}.$$

Substitute c_{j0} and c_{js} in (8):

$$q_j^0 + \sum_{i=1}^{s-1} c_{ji} t_k^i + \left(q_j^T - q_j^0 - \sum_{i=1}^{s-1} c_{ji} \right) t_k^s \geq q_j^L, \tag{9}$$

$$q_j^0 + \sum_{i=1}^{s-1} c_{ji} t_k^i + (q_j^T - q_j^0 - \sum_{i=1}^{s-1} c_{ji}) t_k^s \le q_j^H, \quad j = 1,2,\ldots,n,$$

$$\sum_{j=1}^{n} \left[q_j^0 + \sum_{i=1}^{s-1} c_{ji} t_k^i + (q_j^T - q_j^0 - \sum_{i=1}^{s-1} c_{ji}) t_k^s - q_{mj}^p \right]^2 \ge r^2, m = 1,2,...,M, \quad k = 1,2,...,K.$$

Thus, it is necessary to solve the system of $(M + 2n) \cdot K$ nonlinear inequalities. This problem can be replaced by the problem of optimization of some function $F(C)$:

$$F(C) = \begin{cases} E(C), if\ E(C) < 0 \\ P(C), if\ E(C) = 0, \end{cases}$$

where C is vector of coefficients $(c_{1,1}, c_{1,2}, \ldots, c_{1,s-1}, c_{2,1}, c_{2,2}, \ldots, c_{2,s-1}, \ldots, c_{n,1}, c_{n,2}, \ldots, c_{n,s-1})$, $E(C)$ is measure of restrictions violation, $P(C)$ is estimated probability that trajectory not intersects with unknown obstacles. Let's define function $F(C)$. First, introduce function defining the trajectory point for the vector of coefficients C at the moment t:

$$L_j(C,t) = q_j^0 + \sum_{i=1}^{s-1} c_{ji} t^i + (q_j^T - q_j^0 - \sum_{i=1}^{s-1} c_{ji}) t^s.$$

The following functions correspond to the inequalities of the system (8), they show the violation of upper and lower bounds and the intersection with the obstacles of trajectory at the moment t:

$$E_L(C,t) = \sum_{j=1}^{n} I(L_j(C,t) - q_j^L),$$

$$E_H(C,t) = \sum_{j=1}^{n} I(q_j^H - L_j(C,t)),$$

$$E_P(C,t) = \sum_{m=1}^{M} I\left[\sum_{j=1}^{n} (L_j(C,t) - q_{mj}^p)^2 - r^2 \right],$$

$$I(z) = \begin{cases} z, & if\ z < 0, \\ 0, & if\ z \ge 0. \end{cases}$$

Because of using of the operator I, the items in the above functions are negative only if corresponding restrictions are violated. The more violation of the restriction, the more the value of the function. If particular restriction is not violated, the corresponding function element is zero. In any case, values of functions can't be positive. Join all restrictions into single function (taking into account all discrete moments except $t = 0$ and $t = 1$):

$$E(C) = \sum_{k=1}^{K} \left[E_L(C, t_k) + E_H(C, t_k) + E_P(C, t_k) \right].$$

Thus, $E(C)$ takes negative values if at least one restriction is violated and becomes zero otherwise, that is if the vector C satisfies all the inequalities of the system (9). The function $P(C)$ was introduced to make possible comparison of trajectories which not violate the restrictions. Assume $p(d) = e^{-2d}$ is probability that there is an unknown obstacle in some point, where d is distance between this point and the nearest known obstacle. Then $P(C) = \prod_{m=1}^{M^*} p(D(C, O_m))$, where $\{O_1, O_2, \ldots, O_{M^*}\}$ is set of obstacles along which the trajectory lies, $D(C,O) = \min_k \sqrt{\sum_{j=1}^{n} (L_j(C, t_k) - O_j)^2 - r^2}$ is distance between the trajectory C and the obstacle O. A trajectory *lies along* an obstacle O if such point of trajectory $L(t_k)$ exists as an obstacle O is nearest among all obstacles for this point. Usage of function P(C) would promote the algorithm to produce trajectories which lie far from unknown obstacles. Since function $F(C)$ is multiextremal, the genetic algorithm is used to find the desired vector.

3.3 Optimization of the restriction function using genetic algorithm

Genetic algorithm is based on collective training process inside the population of individuals, each representing search space point. In our case search space is space of vectors C. Encoding of vector in the individual is made as follows. Each vector component is assigned the interval of values possessed by this component. The interval is divided into a quantity of discrete points. Thus, each vector component value is associated with corresponding point index. The sequence of these indexes made up the individual. The algorithm scheme:

1. Generate an initial population randomly. The size of population is N.
2. Calculate fitness values of the individuals.
3. Select individuals to the intermediate population.
4. With probability P_C perform crossover of two individuals randomly chosen from the intermediate population and put it into new population; and with probability 1 – P_C perform reproduction – copy the individual randomly chosen from intermediate population to new population.
5. If size of new population is less than N, go to Step 4.
6. With given mutation probability P_M invert each bit of each individual from new population.
7. If required number of generations is not reached go to Step 2.

The fitness function matches $F(C)$. On the third step the tournament selection is used: during the selection the N tournaments are carried out among m randomly chosen individuals (m is called tournament size). In every tournament the best individual is chosen to be put into the new population. On the fourth step the arithmetical crossover is used. The components of offspring are calculated as arithmetic mean of corresponding components of two parents.

3.4 Quantization of the path

After the polynomials coefficients specifying the route are found, it's necessary to get the sequence of route discrete points q^0, q^1, …, q^T. The first point (q^0) is obtained from the initial values. The rest of points can be found using the following algorithm:

1. $t = 0$; $i = 0$.
2. $t_H = 1$.
3. $t^* = \frac{t + t_H}{2}$.
4. Find the point q^* with coordinates (q^*_1, q^*_2, …, q^*_n) in whose neighborhood the trajectory is lying at the moment t^*: $q_j^* - \frac{\Delta q}{2} \le q_j(t^*) < q_j^* + \frac{\Delta q}{2} \quad \forall j = \overline{1,n}$
5. If q^* equals to q^i, then $t = t^*$, go to Step 3.
6. If q^* is not a neighbor of q^i, then $t_H = t^*$, go to Step 3.
7. If q^* is not forbidden, then go to Step 9.
8. If $q_j^i - \Delta q \le q_j(t^*) \le q_j^i + \Delta q \quad \forall j = \overline{1,n}$, where q^i_j are the coordinates of q^i, then $t = t^*$, otherwise $t_H = t^*$, go to Step 3.
9. $i = i + 1$; $q^i = q^*$.
10. If $q^i = q^T$, then the algorithm is finished, otherwise go to Step 2.

4. Experimental results

Consider the following experimental set (Figure 5). It is necessary to move a seven-link manipulator (Figure 4) from the start configuration q^0 = (1.57; 1.57; 0; 4.71; 0; 4.71; 0) to the target configuration q^T = (4.71; 1.57; 0; 0; 0; 0; 0). There are the following limitations on the generalized coordinates: $0 \le q_i(t) \le 6.28$, i = 1,2,…,7. The lengths of links are 10. There are four cuboid obstacles in the working area. Each obstacle is described by six values: the length, width, height and the coordinates of the origins attached to the obstacles in the basic coordinate system (Table 1).

The parameters of algorithms are as follow:

1. Polynomial order s: 10.
2. Number of time pieces K: 100.
3. Polynomials coefficients precision: 10^{-5}.
4. Population size N: 20.
5. Number of generations: 20.
6. Probability of crossover P_C: 0,5.
7. Probability of mutation P_M: 0,1.
8. Tournament size m: 5.

The working time of the Algorithm depending on different *number_of_discretes* is given in the Table 2. The Δq is calculated as the difference between upper and lower bounds of $q(t)$ (that is 6.28) divided on the *number_of_discretes*. The working time is a sum of three elements: trajectory search time, time to check whether trajectory intersects with unknown obstacles, manipulator moving time (12° per second).

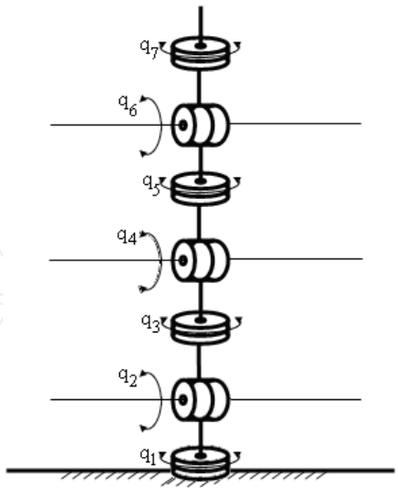

Fig. 4. The manipulator kinematic scheme

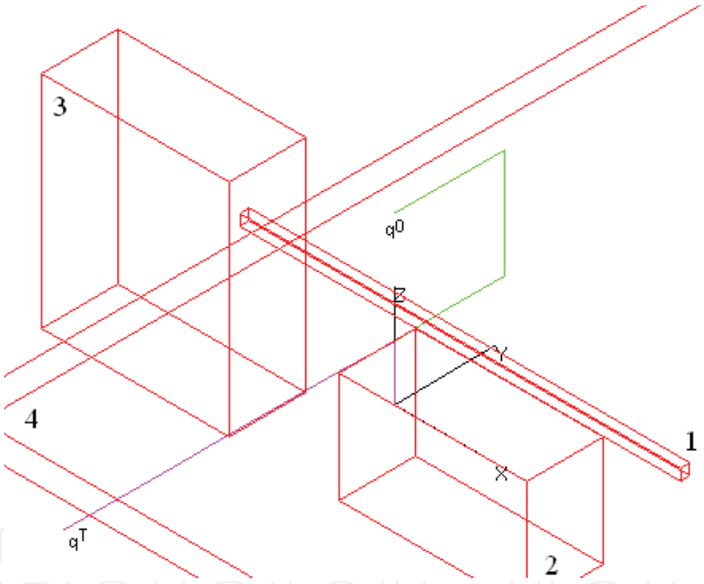

Fig 5. Experimental set

№	*x*	*y*	*z*	length	width	height
1	-30	2	12	80	1.6	2
2	10	-20	0	34	14	20
3	-44	-20	0	34	14	40
4	-40	-40	-10	200	200	10

Table 1. The characteristics of obstacles

Obstacles	*number_of_ discretes*	Working time, seconds
1, 2	40	58
	60	82
	120	153
	240	150
	360	125
1, 2, 3	40	83
	60	96
	120	154
	240	233
	360	251
1, 2, 3, 4	40	233
	60	340
	120	761
	240	1142
	360	1169

Table 2. Experimental results

The tests were made on the processor AMD Athlon XP 1800+ (1533 MHz). During experiments the value N_n in the Algorithm was equal to 0 for every n. The key steps of manipulator trajectory for the last test case (with four obstacles) are shown on the Figure 6.
One may see that the software based on our Algorithm moved the 7-link manipulator to the target configuration in 58-1169 seconds. In case of using breadth-first or A* algorithms (La Valle, 1999-2004) we have to discretize the configuration space and therefore we get m^n points of the configuration space, where m is a number of points for certain level of discretization, n is the configuration space dimensionality. The working time of such algorithms may approximate to billions of seconds.
It is possible to outline the following criteria which should be satisfied by an algorithm for SUBROUTINE: a)it should be applicable to the n-dimensional case; b) it should guarantee finding a path in the presence of known forbidden states; c) in case of new call of STEP 1 minimum work for finding path in the presence of known forbidden states should be done.
Using of cell decomposition and bitangent graphs algorithms (La Valle, 1999-2004) as algorithms for SUBROUTINE of our Algorithm looks promising.

5. Conclusion

An algorithm for a n-link manipulator movement amidst arbitrary unknown static obstacles was presented. Given a theorem stating that if the manipulator moves according to the algorithm it will reach a target configuration q^T in a finite number of steps under condition that the q^T is reachable. Given proof of the theorem. It was shown that the problem of the manipulator control in the unknown environment may be reduced to a solution of a finite number of problems for the manipulator control in known environment. Given the sequences from the theorem which may facilitate reaching the q^T. We also introduced the results of computer simulation of a 7- link manipulator movement in the unknown environment based on our Algorithm and compared results with some other algorithms.

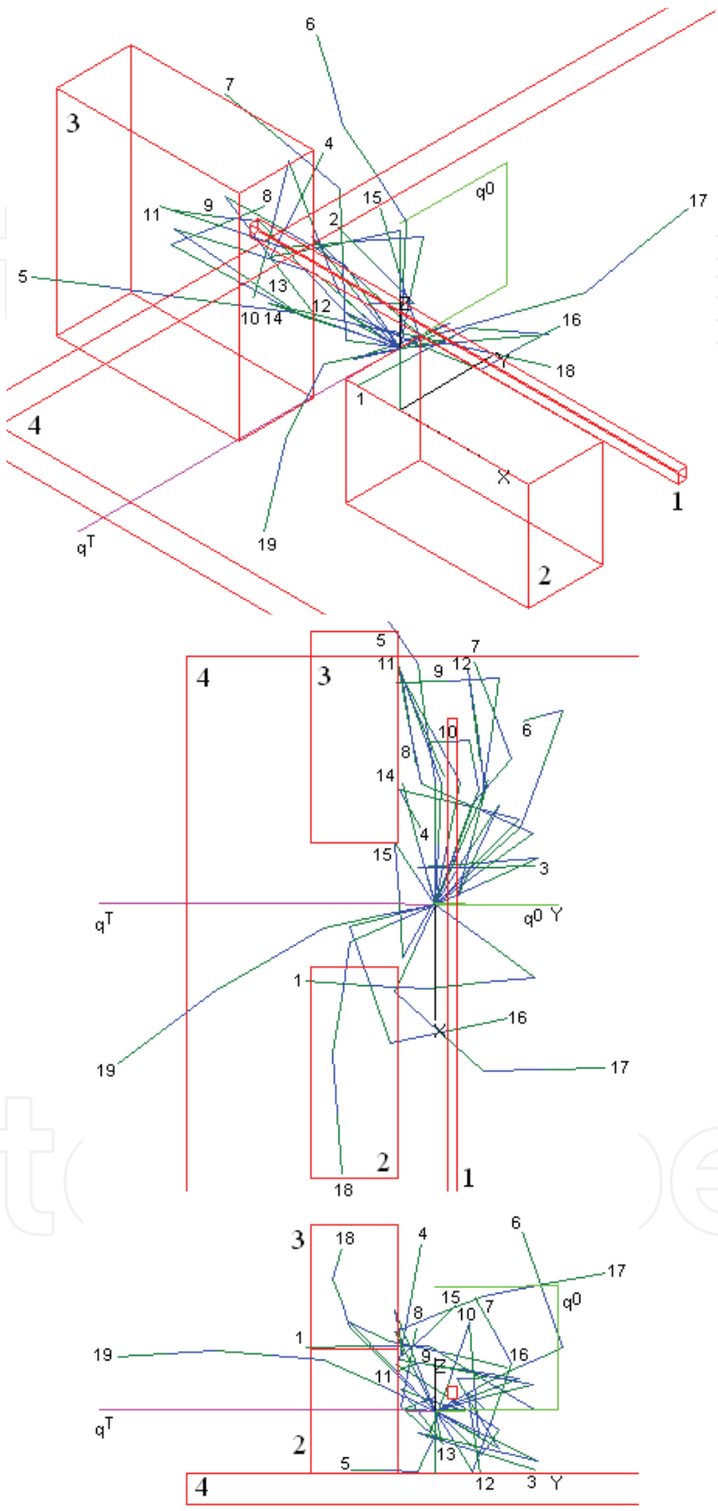

Fig. 6. The key steps of manipulator movement trajectory

6. References

C.Ahrikhencheikh, A. Seireg, *Optimized-Motion Planning: Theory And Implementation*. John Wiley & Sons, Inc, 1994.

N.M. Amosov, A.M. Kasatkina, L.M. Kasatkina, "Aktivnye semanticheskiye seti v robotakh s avtonomnym upravleniyem (Active semantic nets in robots with autonomous control)", Trudy IV Mezhdunarodnoy obyedinennoy konferencii po iskustvennomu intellectu. M.: AN SSSR, 1975, v.9, pp. 11-20 (in Russian).

J. Canny, "The Complexity of Robot Motion Planning", The MIT Press, Cambridge, Massachusetts, 1988.

C. Chen, H.-X. Li, D. Dong, "Hybrid Control for Robot Navigation. A hierarchical Q-learning algorithm", IEEE Robotics & Automation Magazine, Vol.15, № 2, June 2008, pp. 37-47.

H. Choset et al., "Principles of Robot Motion. Theory, Algorithms and Implementations", A Bradford Book. The MIT Press. 2005.

G.E.Collins, "Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition", Lecture Notes in Computer Science, Vol.33, pp.135-183, Springer-Verlag, New York, 1975.

B.R.Donald, "On Motion Planning with Six Degrees of Freedom: Solving the Intrsection Problems in Configuration Space", Proceedings of the IEEE International Conference on Robotics and Automation (1985).

S.K. Ghosh, J.W. Burdick, A. Bhattacharya, S. Sarkar, "Online Algorithms with Discrete Visibility. Exploring Unknown Polygonal Environments", IEEE Robotics & Automotion Magazine. Vol.15, №2, June 2008. pp.67-76.

V.A. Ilyin, "Intellektualnye Roboty, Teoriya I Algoritmy (Intelligent Robots: Theory and Algorithms)", Krasnoyarsk, SAA Press, 1995 (in Russian).

A.M. Kasatkin, "O predstavlenii znaniy v sistemah iskustvennogo intellecta robotov (On knowledge representation in artificial intelligence systems of robots)", Kibernetika, 1979, №2, pp.57-65 (in Russian).

E.M.Kussul, V.D. Fomenko, "Maket avtonomnogo transportnogo robota TAIR (A Model of an autonomous transport robot TAIR)", Voprosy teorii avtomatov, robotov I CVM. Kiyev, 1975. pp.60-68 (in Russian).

S.M. LaValle, "Planning Algorithms", 1999-2004. Available: http:// msl.es.uiuc.edu/ planning

P. K. Lopatin, "Algorithm of a manipulator movement amidst unknown", *Proceedings of the 10th International Conference on Advanced Robotics (ICAR 2001*), August 22-25, 2001, Hotel Mercure Buda, Budapest, Hungary – pp.327-331.

P. K. Lopatin, "Algorithm 2 upravlenia dinamicheskimi sistemami v neizvestnoy staticheskoy srede (Algorithm2 for Dynamic Systems' Control in an Unknown Static Environment)". *Herald of The Siberian state aerospace university named after academician M.F.Reshetnev / ed. prof. G.P.Belyakov; SibSAU.* № 4(11). Krasnoyarsk. pp.28-32, 2006. (in Russian).

P. K. Lopatin, A. S. Yegorov, "Using the Polynomial Approximation Algorithm in the Algorithm2 for Manipulator's Control in an Unknown Environment", *International Journal of Applied Mathematics and Computer Sciences,* Volume 4, Number 4, pp.190-195. http://www.waset.org/ijamcs/v4/v4-4-32.pdf

V. Lumelsky, "Sensing, Intelligence, Motion: How Robots and Humans Move in an Unstructured World", John Wiley & Sons, 2006.

E. Masehian, M.R. Amin-Nasari, "Sensor-Based Robot Motion Planning. A Tabu Search Approach", IEEE Robotics & Automation Magazine. Vol. 15, №2, June 2008, pp.48-57.

D. Rawlinson, R. Jarvis, "Ways to Tell Robots Where to Go. Directing autonomous robots using topological instructions", IEEE Robotics & Automation Magazine. Vol. 15, №2, June 2008, pp.27-36.

M. Shahinpoor, "A Robot Engineering Textbook", Harper & Row Publishers, New York, 1987.

A.V. Timofeev, "Roboty i Iskusstvennyi Intellekt (Robots and Artificial Intelligence)", M.: Nauka, 1978 (in Russian).

Ye. I. Yefimov, "Problema perebora v iskusstvennom intellekte (The search problem in artificial intelligence)", Izv. AN SSSR. Tehnicheskaya Kibernetika. 1988. №2, pp.127-128.

F. Yegenoglu, A. M. Erkmen, H.E. Stephanou, "On-line Path Planning Under Uncertainty," Proc. 27th IEEE Conf. Decis. And Contr., Austin, Austin, Tex., Dec.7-9, 1988. Vol.2, pp.1075-1079, New York (N.Y.), 1988.

IntechOpen

Advanced Technologies

Edited by Kankesu Jayanthakumaran

ISBN 978-953-307-009-4

Hard cover, 698 pages

Publisher InTech

Published online 01, October, 2009

Published in print edition October, 2009

This book, edited by the Intech committee, combines several hotly debated topics in science, engineering, medicine, information technology, environment, economics and management, and provides a scholarly contribution to its further development. In view of the topical importance of, and the great emphasis placed by the emerging needs of the changing world, it was decided to have this special book publication comprise thirty six chapters which focus on multi-disciplinary and inter-disciplinary topics. The inter-disciplinary works were limited in their capacity so a more coherent and constructive alternative was needed. Our expectation is that this book will help fill this gap because it has crossed the disciplinary divide to incorporate contributions from scientists and other specialists. The Intech committee hopes that its book chapters, journal articles, and other activities will help increase knowledge across disciplines and around the world. To that end the committee invites readers to contribute ideas on how best this objective could be accomplished.

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen