Spatial Planning and Geometric Optimization: Combining Configuration Space and Energy Methods

Dmytro Chibisov, Ernst W. Mayr, Sergey Pankratov

Institut für Informatik, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany; chibisov@in.tum.de, mayr@in.tum.de, pankrato@in.tum.de

Abstract. In this paper, we propose a symbolic-numerical algorithm for collision-free placement and motion of an object avoiding collisions with obstacles. The algorithm is based on the combination of configuration space and energy approaches. According to the configuration space approach, the position and orientation of the geometric object to be moved or placed is represented as an individual point in a configuration space, in which each coordinate represents a degree of freedom in the position or orientation of this object. The configurations which, due to the presence of obstacles, are forbidden to the object, can be characterized as regions in this configuration space called configuration space obstacles. As will be demonstrated, configuration space obstacles can be computed symbolically using quantifier elimination over the reals and represented by polynomial inequalities. We propose to use the functional representation of semi-algebraic point sets defined by such inequalities, so-called R-functions, to describe nonlinear geometric objects in the configuration space. The potential field defined by R-functions can be used to "move" objects in such a way as to avoid collisions. Introducing the additional function, which forces the object towards the goal position, we reduce the problem of finding collision free path to a solution of the Newton's equations, which describes the motion of a body in the field produced by the superposition of "attractive" and "repulsive" forces. These equations can be solved iteratively in a computationally efficient manner. Furthermore, we investigate the differential properties of R-functions in order to construct a suitable superposition of attractive and repulsive potentials.

1 Introduction

Many practical geometric problems for industrial applications deal with placing and moving an object without colliding with nearby objects. The intricate nature of such problems manifests itself in enhanced computational complexity, whereas in industrial applications these problems must often be solved in real time. In the present paper, we consider two main types of spatial planning problems in a common framework:

– **FindSpace**: optimal placement of geometric objects, for example, maximizing the number of objects of similar shape that can be cut out from a piece

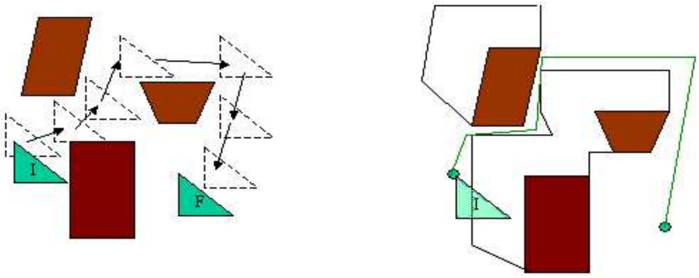

Fig. 1. Configuration space approach - enlarging obstacles: a problem of motion planning is reduced to finding a curve in the configuration space

of material, minimizing the quantity of material needed to produce certain shapes, various packing problems, etc. (see, for example, [5]);
- **FindPath**: finding a collision-free motion path of an object amidst some obstacles of a particular shape, for example, an automatic assembly using an industrial robot, which requires grasping objects, moving them without collisions, and ultimately bringing them together.

The position and orientation of the geometric object to be moved or placed in the real space may be manifested as an individual point in a configuration space, in which each coordinate represents a degree of freedom in the position or orientation of this object ([8]). The configurations which, due to the presence of obstacles, are forbidden to the object can be characterized as regions in the configuration space called *configuration space obstacles* (see Fig. 1). The algorithm which solves the translational and rotational collision-free motion or safe placement problem when the objects are polygons or polyhedra was first presented in [8]. This algorithm computes configuration space obstacles using the notion of the Minkowski sum. After the configuration space obstacles have been calculated, the problem of motion planning is reduced to finding a path in the so-called *visibility graph*. In the presence of rotational motion, the induced configuration space obstacles may be represented as nonlinear constraints, which can be approximated by linear constraints. As noted in [9], the fundamental difficulty is that an exponential number of linear constraints would be required to approximate even a quadratic surface within an accuracy of 2^{-n}, resulting in an exponential time algorithm.

The exact computation of configuration space obstacles can be done with the aid of real quantifier elimination methods, as will be discussed in Section 2. The configuration space obstacles are semi-algebraic sets and the task of collision-free motion planning is then reduced to the problem of constructing a semi-algebraic curve between two points, such that the intersection of this curve with the interior of semi-algebraic set is empty. This purely geometric problem has been solved in [14] using Cylindrical Algebraic Decomposition ([4]) of semi-algebraic sets. The latter algorithm can be performed in time polynomial in the number of polynomials as well as their maximal degree and double exponential in the number of variables. More efficient algorithms for the path calculation are presented in [1], [2] and have single exponential bounds in the number of variables. One of disadvantages of the mentioned algorithms is that they follow the boundary of configuration space and may produce paths, which touch obstacles. To calculate paths with maximal clearance from obstacles several methods based on Voronoi diagrams have been proposed (see [2], [15]).

In contrast to all these approaches, the objective of the present work is the generalization of finding a geometric path in order to

– find paths, which guarantee a certain minimum clearance from obstacles,
– provide the possibility to incorporate nonholonomic motion constraints (velocities, acceleration, etc.).

For this purpose, we shall describe a family of analytic functions with the property to rise in the vicinity of obstacles of arbitrary shape in the direction towards them. Using such "obstacle functions" (sometimes called "distance function"), we shall show how a "goal function" (sometimes called "target function") can be constructed, which decreases monotonously along some path from the initial to the final position, if and only if the path does not intersect any obstacle. Combining obstacle and goal function, we shall obtain a scalar-valued "navigation function" such that the problem of motion planning can be reduced to the task of following the gradient of the navigation function.

To our knowledge, the idea of using scalar valued functions for the obstacle avoidance was pioneered in [6]. The author proposed the navigation functions for the case the obstacles are a parallelepiped, a finite cylinder, and a cone. However, these geometric primitives do not form a sufficient set to describe the images of obstacles in the configuration space. The first construction of a general analytic navigation function is due to [11]. The authors show how a smooth navigation function can be constructed for the case when obstacles are smooth manifolds. In the present paper, we describe the construction of a more general family of navigation functions for arbitrary semi-algebraic objects. For this purpose, we shall use the functional representation of semi-algebraic point sets defined by so-called R-functions ([12], [16]) and reduce the problem of path finding to the solution of the Newton's equations of motion in a field of forces that can be done numerically. The obstacle and goal functions play the role of repulsive and attractive forces that push the object away from obstacles and pull it towards the goal position. As will be shown, the R-functions exhibit a wide range of differential properties, which can be used for the purpose of nonholonomic motion control. The implementation of our approach and computational examples will be presented.

2 Description of Geometric Objects Using R-Functions

The theory of R-functions ([12],[16]) provides the methodology of constructing an implicit functional representation for any semi-algebraic set using logical (set-theoretical) operations. In this section we shall briefly introduce this concept and some results from the theory of R-functions, which shall be used in Section 3 for the purpose of the collision-free motion planning.

Let $F(X_1, ..., X_n)$ be a Boolean function with truth value 1 and false value 0 built using logical operations $\wedge$, $\vee$ and $\neg$. A real valued function $f(x_1, ..., x_n)$ is called an R-function if its sign is completely determined by the signs of its arguments. More precisely, f is an R-function if there exists a Boolean function F such that

$$sign(f(x_1, ..., x_n)) = F(sign(X_1), ..., sign(X_n)). \tag{1}$$

In other words, f works as a Boolean switching function, changing its sign only when its arguments change their signs. For example, logical operations on Boolean variables X_1, X_2 may be performed on real-valued variables x_1, x_2 such that (1) is satisfied using the following rules:

$$\begin{array}{l} x_1 \wedge \; x_2 \equiv x_1 + x_2 - \sqrt{x_1^2 + x_2^2} \\ x_1 \vee x_2 \equiv x_1 + x_2 + \sqrt{x_1^2 + x_2^2} \\ \neg x_1 \equiv -x_1. \end{array} \tag{2}$$

Consider, e.g., the Boolean function defined by

$$F(X_1, X_2, X_3, X_4) = X_1 \wedge X_2 \wedge X_3 \wedge X_4 \wedge X_5.$$

The corresponding real valued function f may be defined recursively according to (2):

$$\begin{array}{l} f_1(x_1, x_2) = x_1 + x_2 - \sqrt{x_1^2 + x_2^2} \\ f_2(x_3, x_4) = x_3 + x_4 - \sqrt{x_3^2 + x_4^2} \\ f(x_1, x_2, x_3, x_4, x_5) = f_1 + f_2 - \\ \sqrt{{f_1}^2 + {f_2}^2} + x_5 - \sqrt{\left(f_1 + f_2 - \sqrt{{f_1}^2 + {f_2}^2}\right)^2 + {x_5}^2} \end{array} \tag{3}$$

This R-function can be used to describe point sets bounded by four arbitrary polynomials:

$$\begin{array}{r} R(x, y) = \{(x, y) | \phi_1(x, y) \geq 0 \wedge \phi_2(x, y) \geq 0 \wedge \\ \phi_3(x, y) \geq 0 \wedge \phi_4(x, y) \geq 0 \wedge \phi_5(x, y) \geq 0\} \end{array} \tag{4}$$

For example, let four lines in the plane be given by the roots of the polynomials ϕ_i, $i = 1...4$,

$$\begin{array}{l} \phi_1(x, y) = x \\ \phi_2(x, y) = x - 4 \\ \phi_3(x, y) = y \\ \phi_4(x, y) = y - 4 \end{array}$$

and a circle be given by

$$\phi_5(x, y) = (x - 2)^2 + (y - 2)^2 - 1.$$

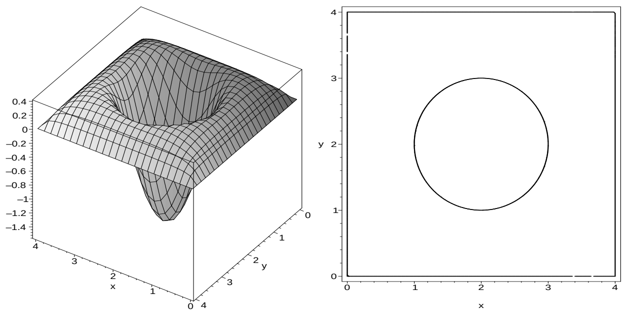

Fig. 2. O(x,y) and its roots.

The object shown in Fig. 2 can be described as semi-algebraic set (4) or, alternatively, with the help of analytic function $f(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5)$, that is equal to zero on the boundary of the object, positive inside and negative outside the object. In this way, any complicated semi-algebraic object can be constructed from primitive algebraic objects. Thus, R-functions enable one to write easily an equation for an object of arbitrary shape in the same way as one forms geometric objects by logical or set theoretic operations ([10]). Therefore R-functions are helpful in describing complicated semi-algebraic objects as an analytic function having the following sign property

$$\begin{array}{ll} f(\mathbf{x}) > 0 & \text{if } \mathbf{x} \text{ is inside the object} \\ f(\mathbf{x}) = 0 & \text{if } \mathbf{x} \text{ is on the boundary of the object} \\ f(\mathbf{x}) < 0 & \text{if } \mathbf{x} \text{ is outside the object} \end{array}$$

Alternatively to (2), the following rules described in [12] can be used to form union and intersection of geometric objects:

$$R_\alpha : \frac{1}{1+\alpha}(x_1 + x_2 \pm \sqrt{x_1^2 + x_2^2 - 2\alpha x_1 x_2}),$$

where $\alpha(x_1, x_2)$ is an arbitrary symmetric function such that $-1 < \alpha(x_1, x_2) < 1$.

$$R_0^m : (x_1 + x_2 \pm \sqrt{x_1^2 + x_2^2})(x_1^2 + x_2^2)^{\frac{m}{2}},$$

where m is any even positive integer.

$$R_p : x_1 + x_2 \pm (x_1^p + x_2^p)^{(\frac{1}{p})},$$

for any even positive integer p.

In each case above, choosing the $+/-$ sign determines the type of an R-function: $(+)$ corresponds to R-disjunction and $(-)$ sign gives the R-conjunction. The given families of R-functions exhibit a wide range of differential properties, which are studied in [17]. The change of parameters α, m and p leads to different characteristics of the navigation function, which will be described in Section 4 and allows to control the velocity or acceleration of the object.

The following theorem about the derivative of R_α-functions at the boundary has been proven in [12]. It states that the absolute value of the derivative of an R-function at the boundary point $\mathbf{p}$ in the given vector direction is equal to the absolute value of the derivative of the polynomial ϕ_i, which describes this part of boundary, provided the boundary part ϕ_i does not intersect with any other boundary boundary part ϕ_j in $\mathbf{p}$. The sign of the derivative is determined by the number of logical negations of x_i, called inversion degree.

Theorem 1 (Rvachev [12], [16]). *Let $f(x_1, ..., x_N)$ be such R_α-function that argument x_i appears in f only once and has the inversion degree m. Suppose the functions $\phi_1, ..., \phi_N$ and f are continuously differentiable and satisfy the following condition at point $\mathbf{p}$:*

$$\phi_i(\mathbf{p}) = 0; \phi_j(\mathbf{p}) \neq 0, i \neq j;$$

$$f(\phi_1, ..., \phi_N)\big|_{\mathbf{p}} = 0.$$

Then, for any vector direction l, the following equality holds

$$\left.\frac{\partial f(\phi_1, ..., \phi_N)}{\partial l}\right|_{\mathbf{p}} = (-1)^m \left(\frac{\partial \phi_i}{\partial l}\right)\bigg|_{\mathbf{p}}.$$

For example, for any point $\mathbf{p}$ on the boundary part ϕ_i, $i = 1...5$, shown in Fig. 2, the following condition is satisfied

$$\left.\frac{\partial f(\phi_1, ..., \phi_5)}{\partial l}\right|_{\mathbf{p}} = \left(\frac{\partial \phi_i}{\partial l}\right)\bigg|_{\mathbf{p}}.$$

This condition allows one to use the gradient of R-functions to predict the presence of obstacles and avoid collisions, as will be described in Section 4.

3 Computing Configuration Space Obstacles

As mentioned above, an important part in our approach to motion planning is a configuration space method ([8]). We propose to use the following two solutions:

- exact computation of configuration space obstacles based on quantifier elimination methods ([7]);
- approximation of configuration space obstacles by nonlinear constraints, which can be calculated in a more efficient manner ([13]).

In the following paragraphs we shall briefly describe both approaches.

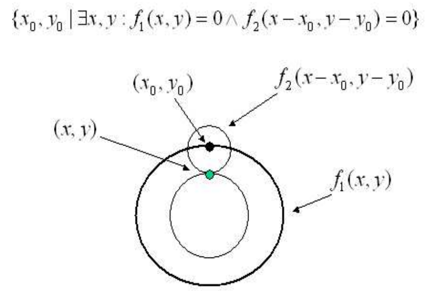

Fig. 3. Calculation of configuration space obstacles by quantifier elimination

Exact computation of configuration space obstacles This algorithmic problem can be formulated as a decision problem for the first-order theory of real fields. The real numbers constitute an ordered field, which is closed under addition and multiplication. The formulas in the first-order theory of reals, defined by A. Tarski in 1930 and called the Tarski formulas, are composed from equalities and inequalities. Such formulas may be constructed by introducing logical connectives (conjunction, disjunction and negation) and the universal and existential quantifiers to the atomic formulas.

For example, let some geometric object representing obstacle O be bounded by roots of finitely many polynomials $O_{i,j}(x, y, z)$. The inequalities $O_{i,j}(x, y, z) \geq 0$ and $O_{i,j}(x, y, z) \leq 0$ can be used to describe the exterior and interior of the object. Choosing the suitable sign of the polynomials we may use only "$\geq$" to describe any geometric object. The Tarski formula describing the set of points, which belong to the object, can be written as follows:

$$O(x, y, z) \equiv \bigvee_i \bigwedge_j O_{i,j}(x, y, z) \geq 0$$

The object P to be moved is given by roots of polynomials $P_i(x, y, z)$ and can be described with a Tarski formula in the same way. A shift of the object by x_0, y_0, z_0 units can be written as:

$$P(x, y, z) \equiv \bigvee_i \bigwedge_j P_{i,j}(x - x_0, y - y_0, z - z_0) \geq 0$$

As can be seen in Fig. 1 and 3, in the two-dimensional case the configuration space obstacle corresponding to O can be calculated for a particular orientation of P by contacting P with O and moving P along the boundary of O keeping them in contact. The resulting geometric object is the configuration space obstacle O_ϕ^{Conf} corresponding to the orientation ϕ of P.

The contact of O and P can be expressed in terms of common roots of bounding polynomials. Thus, O_ϕ^{Conf} corresponds to such shifts x_0, y_0, z_0 of P where some of polynomials P_i and O_i have common roots. This can be formalized with a Tarski sentence as follows:

$$\{(x_0, y_0, z_0) | \exists x, y, z : P(x - x_0, y - y_0, z - z_0) = 0 \wedge O(x, y, z) = 0\}$$

Eliminating $\exists$-quantifiers with existing methods ([3]) produces the semi-algebraic set that corresponds to O_{ϕ}^{Conf}. The latter quantity can also be described with the aid of R-functions, as explained in Section 2. In Section 4 we shall show how such description can be used to predict collisions.

Approximate computation of configuration space obstacles The configuration space obstacles can be calculated using the notion of the Minkowski sum. An algorithm for the approximation of the Minkowski sum with the help of R-functions has been proposed in [13]. Suppose P and O are defined by R-functions $P(x_1, ..., x_d) \geq 0$ and $O(x_1, ..., x_d) \geq 0$, respectively. The intersection of P shifted by $s_1, ..., s_d$ units and O can be written as

$$F(x_1, ..., x_d, s_1, ..., s_d) = P(x_1 - s_1, ..., x_d - s_d) \wedge O(x_1, ..., x_d)$$

As explained above, O_{ϕ}^{Conf} consists exactly of such shifts $s_1, ..., s_d$, which produce the contact between P and O. The contact of P and O means that their intersection is not empty. In this case $F \geq 0$, otherwise, if P does not touch O, $F < 0$.

Thus, the projection of $F(x_1, ..., x_d, s_1, ..., s_d)$ must be calculated: find such $s_1, ..., s_d$ so that there exist some $x_1, ..., x_d$ with $F(x_1, ..., x_d, s_1, ..., s_d) \geq 0$. As shown in [13], this projection can be computed by solving the following maximization problem:

$$O_{\phi}^{Conf}(s_1, ..., s_d) = \mathbf{max}\{F_3(x_1, ..., x_d, s_1, ..., s_d)\}.$$

The necessary condition for a point $(x_1, ..., x_{2d})$ where the maximum is attained:

$$\frac{\partial F_3}{\partial s_i} = 0, i = 1...d;$$

These equations can be solved numerically, for example, with the help of the Newton's method. In this manner, the configuration space obstacles can be represented as R-functions and used to predict collisions with obstacles.

4 Navigation in the Configuration Space

As mentioned above, the calculated configuration space obstacles can be represented with the help of R-functions. It follows from Theorem 1 that in the vicinity of obstacles the R-function increases towards them (see Fig. 4). Such "obstacle function" is therefore useful in order to predict collisions and determine the direction of the motion in order to avoid obstacles.Apart from the "obstacle function" O, we introduce the "goal function" G, which is decreasing monotonously along the path π that connects the initial position $\mathbf{s} = (s_1, ..., s_N)$

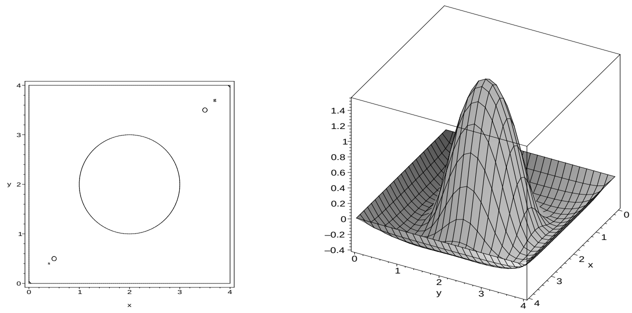

Fig. 4. The region with obstacles (colored black) and its obstacle function. The path from s to g must be calculated.

and the target position $\mathbf{g} = (g_1, ..., g_N)$. The goal function is required to have only one minimum value in $\mathbf{g}$. As we shall describe below, the sum of both functions defines the potential field U, which is used for motion planning (see Fig. 5):

$$U(x_1, ..., x_N) = O(x_1, ..., x_N) + G(x_1, ..., x_N). \tag{5}$$

Different functions with only one minimum value in the goal position and different differential properties can be used. In general, the following conditions must be satisfied:

- G is decreasing monotonously along the shortest path π connecting $\mathbf{s}$ with $\mathbf{g}$, e.g. the sign of the derivatives should be constant:

$$sign\left(\left.\frac{\partial G(x_1, ..., x_N)}{\partial x_i}\right|_{\pi}\right) = const. \tag{6}$$

- U is decreasing monotonously in all points $\mathbf{p} \in \pi$, which lie not too close to any obstacle ($|O(\mathbf{p})| < \epsilon$). From (5), (6) and from the fact that $sign(\frac{\partial O}{\partial x_i}) \neq const$, it follows that the derivatives of G should be greater than those of O:

$$|O(\mathbf{p})| < \epsilon \Leftrightarrow \left|\left.\frac{\partial G(x_1, ..., x_N)}{\partial x_i}\right|_{\mathbf{p}}\right| > \left|\left.\frac{\partial O(x_1, ..., x_N)}{\partial x_i}\right|_{\mathbf{p}}\right| \tag{7}$$

- U has a minimum value in some point $\mathbf{p} \in \pi$ in the vicinity of an obstacle ($|O(\mathbf{p})| \geq \epsilon$) and increases towards the obstacle:

$$|O(\mathbf{p})| \geq \epsilon \Leftrightarrow \left|\left.\frac{\partial G(x_1, ..., x_N)}{\partial x_i}\right|_{\mathbf{p}}\right| \leq \left|\left.\frac{\partial O(x_1, ..., x_N)}{\partial x_i}\right|_{\mathbf{p}}\right| \tag{8}$$

The following functions, which have only one minimum value in the goal position $\mathbf{g}$, can be used as goal functions:

$$G_0(x_1, ..., x_N) = \alpha(\epsilon)\sqrt{|g_1 - x_1| + ... + |g_N - x_N|},$$

$$G_d(x_1, ..., x_N) = \alpha(\epsilon)((g_1 - x_1)^{2d} + ... + (g_N - x_N)^{2d})^{\frac{1}{2d}},$$

where d and $\alpha(\epsilon)$ are the parameters to be chosen in order to satisfy the conditions (6)-(8). Using this function, the potential field U can be constructed according to (5). The collision-free path from the initial to the final position corresponds to the direction of the gradient of U. In other words, we must simply follow the gradient of U. In this way, the purely geometric problem of path calculation can be reduced to the physical problem formulated with the help of the Newton's equations, which describe the motion of an object in the field of some forces $\mathbf{F}$:

$$m\mathbf{a} + \lambda\mathbf{v} = \mathbf{F},$$

where m is a mass of the object to be moved, $\mathbf{a}$ and $\mathbf{v}$ are acceleration and velocity, respectively, and λ is a so-called dissipation coefficient. Large values of λ correspond to the motion in a highly viscous environment. To describe the motion we may write the following differential equation:

$$m\frac{d^2x_i(t)}{dt^2} + \lambda\frac{dx_i(t)}{dt} = -\frac{\partial U(x_1, ..., x_d)}{\partial x_i} \tag{9}$$

(for simplicity, we do not consider curvilinear coordinates here). The force due to the environment "resistance" in our model is taken to be $\mathbf{R} = -\lambda\mathbf{v}$. However, other models, in particular those that account for the resistance increasing with velocity, can also be formulated, e.g. $\mathbf{R} = -C|\mathbf{v}|\mathbf{v}$ or, in the component form, $R^j = -(Cg_{ik}\dot{x}^i\dot{x}^k)^{\frac{1}{2}}\dot{x}^j$. Here g_{ik} is a metric tensor and C is the drag coefficient, which in general depends on the object's geometry and on the Reynolds number. The first term in (9) corresponds to the inertial motion. In our primary example, we assume this term to be small as compared to the dissipative term, which impedes the object's when the object approaches the obstacle. This is justifiable when the inertia coefficient m is small compared to $\lambda\tau_0$ where τ_0 is the characteristic time of object motion. The equations

$$\lambda\frac{dx_i(t)}{dt} = -\frac{\partial U(x_1, ..., x_d)}{\partial x_i} \tag{10}$$

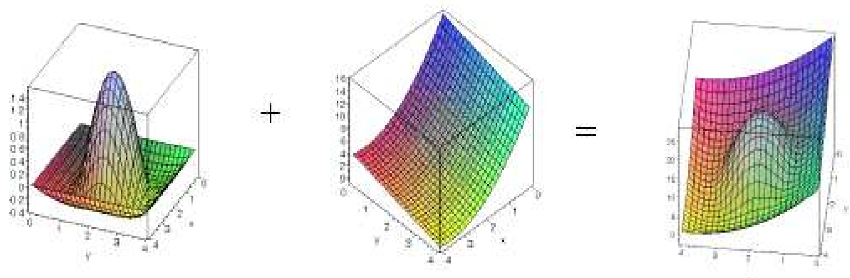

Fig. 5. Addition of the obstacle and the goal functions

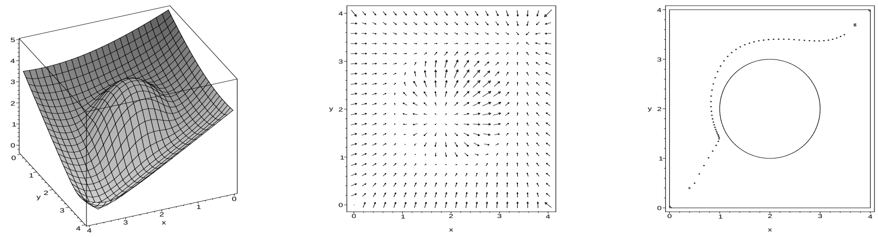

Fig. 6. The potential field (left), its gradient field (in the middle) and the path (right) calculated by following the gradient according to (12). $\frac{\Delta t}{\Delta x} = 0.1$; number of time steps: 54; computational time (using Maple): 0.121 sec

can be solved numerically, e.g. using the finite difference techniques. Numerical methods of solution of the motion equations are mostly based on evaluating the fist derivatives as

$$\frac{df(x)}{dx} = \frac{f(x + \Delta x) - x(x)}{\Delta x} + O(\Delta x).$$

Such discretization of (10) leads to

$$\lambda x_i(t_{j+1}) = \lambda x_i(t_j) - \frac{\Delta t}{\Delta x_i}(U(x_1, ..., x_i + \Delta x_i, ..., x_n) - U(x_1, ..., x_i, ..., x_n)). \quad (11)$$

According to (11), the object position $x(t_{j+1})$ at the time step t_{j+1} can be calculated from the previous position x_i at the time step t_j and the approximation of the gradient of U at $x(t_j)$. Initial values $x_i(0)$ designate the initial positions of an object. Solving the equations

$$\begin{aligned} x_{j+1} &= \lambda x_j - \frac{\Delta t}{\Delta x}(U(x_j + \Delta x, y_j) - U(x_j, y_j)) \\ y_{j+1} &= \lambda y_j - \frac{\Delta t}{\Delta y}(U(x_j, y_j + \Delta y) - U(x_j, y_j)) \end{aligned} \quad (12)$$

leads to the motion shown in Fig. 6, 7. An example with three degrees of freedom demonstrated in Fig. 8 can be produced by solving

$$x_{j+1} = \lambda x_j - \frac{\Delta t}{\Delta x}(U(x_j + \Delta x, y_j, \phi_j) - U(x_j, y_j, \phi_j))$$

$$y_{j+1} = \lambda y_j - \frac{\Delta t}{\Delta y}(U(x_j, y_j + \Delta y, \phi_j) - U(x_j, y_j, \phi_j))$$

$$\phi_{j+1} = \lambda \phi_j - \frac{\Delta t}{\Delta \phi}(U(x_j, y_j, \phi_j + \Delta \phi) - U(x_j, y_j, \phi_j)).$$

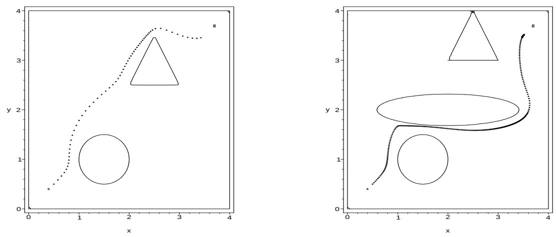

Fig. 7. Examples of calculated paths from the initial to final positions. Left: $\frac{\Delta t}{\Delta x} = 0.1$; number of time steps: 60; computational time (using Maple): 0.251 sec. Right: $\frac{\Delta t}{\Delta x} = 0.035$; number of steps: 300; computational time (using Maple): 1.562 sec.

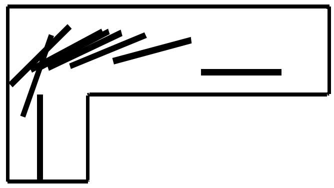

Fig. 8. An example of calculated path with three degrees of freedom: x and y translations and rotation.

5 Conclusion

Our approach to spatial planning and associated geometrical problems is based on the object motion representation in a configuration space, which has a dimensionality equal to the number of independent coordinates describing the object position and orientation in the real space. The advantage of such a method is due to the fact that in the configuration space an object's motion corresponds to that of a fictitious material point moving in a potential field combined with viscous (dissipative) forces. This allows one to employ powerful numerical algorithms to compute collision-free trajectories. The potential field configuration is defined with the help of R-function techniques, which seems to be a convenient method for the functional (analytical) representation of complex geometries. The potential force field defined by R-functions has an attractive and a repulsive parts whose competition determines the goal function and the obstacle function, respectively. As it is typical of such situations, certain extremal properties arise defining the optimal path. The future work will be devoted to the extremal properties of the obstacle and goal functions.

Possible applications of presented techniques, apart from robot motion planning, may include medical kinesiology, biomechanics of human motion, rendering of human body positions, velocities and accelerations, joint simulations - all being modeled with the help of motion equations.

6 Acknowledgments

The authors are grateful to Prof. Hoon Hong and Prof. Christopher Brown for helpful suggestions concerning the presentation of this work as well as for valuable recommendations on further development of the described method.

References

1. Basu, S., Pollack, R., Roy, M.-F.: Computing Roadmaps of Semi-algebraic Sets on a Variety, In Foundations of Computational Mathematics, F. Cucker and M. Shub (eds.), 1-15, Springer-Verlag, 1997
2. Canny, J.: The Complexity of Robot Motion Planning, MIT Press, 1987
3. Caviness B.F., Johnson J.R. (eds.): Quantifier Elimination and Cylindrical Algebraic Decomposition, Springer-Verlag, 1998
4. Collins, G.E.: Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition, Lect. Notes in Comp. Sci. (33), 515-532, Springer-Verlag, 1975
5. Croft T.H., Falconer K. J., Guy R. K.: Unsolved Problems in Geometry, Springer-Verlag, 1991
6. Khatib, O.: Real time obstacle avoidance for manipulators and mobile robots, Internat. J. Robotics, 5, 90-99, 1986
7. Latombe, J.-C.: Robot Motion Planning, Kluwer Academic Publishers, 1991
8. Lozano-Perez, T.: Spatial Planning: A Configuration Space Approach, IEEE Transactions on Computers, C-32 (2), 108-120, 1983
9. Reif, J. H. : Complexity of the Generalized Mover's Problem; In Planing, Geometry and Complexity of Robot Motion, T. Schwartz, M. Sharir, J. Hopcroft (eds.), 267-281, Ablex Publishing Corporation, 1987
10. Requicha, A.: Representations for Rigid Solids: Theory, Methods, and Systems, ACM Computing Surveys (CSUR) Archive, 12 (4), 437-464, ACM Press, 1980
11. Rimon, E., Koditschek, D.E.: The Construction of Analytic Diffeomorphisms for Exact Robot Navigation on Star Worlds, Transactions of the AMS, 327 (1), 71-116, 1991
12. Rvachov, V.L.: Methods of Logic Algebra in Mathematical Physics, Naukova Dumka, Kiev, 1974 (in Russian)
13. Pasko, A., Okunev O., Savchenko V.: Minkowski sum of point sets defined by inequalities, Computers and Mathematics with Applications, Elsevier Science, 45 (10/11), 1479-1487, 2003
14. Schwartz, J., Sharir, M.: On the Piano Movers Problem II. General Techniques to Computing Topological Properties of Real Algebraic Manifolds, Advances in Applied Mathematics, 4, 298-351, 1983
15. Schwartz, J., Yap C. K.: Advances in Robotics , Lawrence Erlbaum associates, Hillside New Jersey, 1986
16. Shapiro, V.: Theory and Applications of R-Functions: A primer, Technical Report, Cornel University, 1991
17. Shapiro, V., Tsukanov, I.: Implicit Functions With Guaranteed Differential Properties; In Proceedings of the Fifth Symposium On Solid Modeling "SOLID MODELING'99", 258-269, Ann Abor, Michigan, ACM Press, 1999