Path Planning for Multi-Robot Inspection Task Considering Acceleration Limits

Jan Faigl[1], Gregor Klančar[2], Drago Matko[2], Miroslav Kulich[3]

[1] The Gerstner Laboratory for Intelligent Decision and Control
[3]Center of Applied Cybernetics
Faculty of Electrical Engineering
Technická 2, 166 27 Prague 6, Czech Republic
xfaigl@labe.felk.cvut.cz

[2]Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Ljubljana, Slovenia
gregor.klancar@fe.uni-lj.si

Abstract

This paper deals with an inspection task of a known environment for a multi-robot team considering limited visibility range and acceleration limits of the robots. The problem is to find routes such that a group of robots surveys the whole working space while moving along the routes. Having a working environment represented as a polygon with holes, the problem can be decomposed into three sub-problems: generation of sensing locations (Art Gallery Problem - AGP), connecting the found locations by a set of paths (Multiple Travelling Salesmen Problem - MTSP), and final optimization of the found paths with respect to dynamic and kinematic capabilities of the robots. First two problems are NP-hard, therefore algorithms finding approximate solutions are used. The AGP solver is based on randomized sampling of the environment, while a self-organizing neural network solves the MTSP. Moreover, the problem of limited visibility range as in real applications is addressed. Routes obtained from the MTSP part consists of points e.g. cities that robots should visit. Among those points a well-planned path considering robot capabilities should be found. The limitation is a grip of tires that results in the acceleration limits. The generated paths are represented as spline curves with maximal allowable velocity profiles that mobile robots could still afford due to acceleration limits.

01. Introduction

An inspection task is one of interesting problems of robot path planning. The problem is to find a route such that a robot surveys the whole working space while it is moved along the route. Variations of this problem are also known as the *searching in polygon* and the *pursuit-evader problem*. The searching in polygon deals with finding a static object in an environment represented as a polygonal map (i.e. polygon with holes), while the object can move arbitrary fast in the pursuit-evader problem.

The inspection problem can be formulated as the Multiple Watchmen Routes Problem (MWRP) in case of a team consisting of multiple robots (guards). The aim is to find an optimal set of routes (one for each of m mobile guards) such that each point of the polygon P is visible from at least one route.

Two criteria defining optimality are mentioned in the literature for the MWRP - *MinSum* and *MinMax*. While the first one defines a cost of a solution as a total length of the routes, solutions with a minimal longest route are preferred by the later criterion. A proof that the problem is NP-hard for both criteria can be found in [1]. Due to this, algorithms do not generate an optimal (best) solution but find some feasible and good enough solution. The algorithm described in the following chapters divides the Multiple Watchmen Route Problem into two sub-problems that can be solved separately.

Sensing locations (i.e. places from which all points of the environment are visible) are found in the first step, which is followed by specification which locations will be visited by which robot and in what order. The first problem is well known as *Art Gallery Problem* (AGP), while the second problem is *Multiple Traveling Salesman Problem* (MTSP). Both problems are known NP-hard.

The output of the MTSP solver consists of a set of routes in the form of poly-line connecting points to be visited by the particular robots. This representation is not appropriate for real applications because it is not easy to control a robot along thereby generated trajectory. Moreover, although this trajectory is the shortest path connecting sensing locations and avoiding obstacles it need not be time-optimal. Therefore a well-planned path considering robot capabilities should be found among those points. Smooth paths - splines composed of cubic polynomials are found and maximal allowable velocity profiles which mobile robots could still afford due to acceleration limits are determined.

The rest of this paper is organized as follows. Section 2 contains description of the used algorithm for determination of sensing locations. A self-organizing neural network algorithm for the MTSP problem is introduced in section 3. A problem of trajectory generation is studied in section 4. Experimental results are shown in section 5. Finally, conclusions and a discussion about achieved results are discussed in section 6.

02. Determination of sensing locations

The Art Gallery Problem for a polygon P is a problem of finding a minimal set of points G (guards) in P such that every point in P is visible from some point of G. Two points in a polygon P are visible, if a straight line segment connecting them entirely lies in P. A polygon with n vertices and h holes can always be guarded with $\left\lfloor \frac{n+h}{3} \right\rfloor$ point guards [2].

If visibility range is restricted to a distance d, two points in a polygon P are d-visible if they are visible and their distance is less than d. The theoretically achieved bound for a polygon with holes is too high in many practical situations. An algorithm based on randomized dual sampling of the environment [3] typically finds less number of sensing locations than the theoretical bound.

Inputs for the sensing locations placement algorithm are a *map* of an environment represented as a polygon with holes and robot's visibility distance d. As a result, a set C contains sensing locations, such that all points of the environment are d-visible from at least one point of C.

The algorithm incrementally adds points to C (which is set initially empty) while a volume of a not covered region S is larger than 0. At first, S is set to contain whole free-space of the environment. In each cycle of the placement procedure, a point p_i is selected randomly at a border of the region S and its d-visibility region V_i (i.e. set of points of the map that are d-visible from p_i) is computed. A given number of points is then randomly placed in V_i and their d-visibility regions are computed as well. A new sensing location l_i is selected from these points so that intersection of its d-visibility region with S is maximal. Finally, the d-visibility region of l_i is subtracted from S.

03. Route finding

A set of sensing locations C found by the placement algorithm acts as an input for the MTSP part of the inspection task. Used self-organizing neural network algorithm is based on work [4]. This algorithm finds an approximate solution of Euclidean instances of the MTSP with MinMax criterion for m salesmen. A route of each salesman starts and finishes at a same city, that is called *depot*. An idea of the algorithm is to represent a path of each particular salesman by a ring of neurons, where neighboring neurons are connected. Initially, a ring of connected neurons is generated for each salesman around depot. Number of neurons in each ring is $\frac{5n_c}{2m}$, where n_c is a number of cities and m is a number of salesmen. An unsupervised learning procedure consists of two main steps: *adaptation* of the closest neuron from each ring to the depot and *adaptation* of the closest neuron to each city. The adaptation procedure lies in moving a neuron with its defined neighborhood closer to the city. A distance by which neurons are moved depends on two factors. The smaller is neuron's topological distance to the winner the more is the neuron moved. Moreover, movement distances are decreased in each learning iteration, which leads to very small changes in final iterations. The learning procedure is repeated until maximal neuron-city distance is smaller than a defined constant.

Detail description of the algorithm can be found in [11]. Example of found routes for three robots is in Fig. 1.

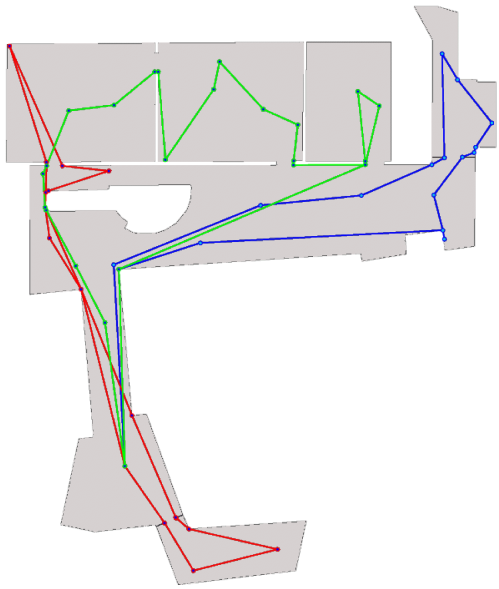


Fig. 2: Solution of MTSP for 3 robots

04. Trajectory generation

The above part of the paper defines routes consisting of points which each robot should visit. Among those points a smooth paths - spline composed of cubic polynomials should be found. The obtained path for each robot must cross all the sensing points defined by the route and must not collide with obstacles (boundaries of the polygon and other obstacles). To assure collision free paths the obtained curves should be close to planned routes (see Fig 3).

The two-dimensional curve is obtained by combining two splines, $x(u)$ and $y(u)$, where u is the parameter along the curve. Each spline consists of more segments - cubic polynomials. Knots are points of tangency of two neighbour segments with continuous (the same) derivatives. When the knots are set, the spline parameters can be obtained by solving a linear equation system [5]. If the spline consists of m polynomial segments of order p, than the number of parameters to determine is $m \cdot (p+1)$ which requires $n = m \cdot (p+1)$

conditions (linear equations) to completely define spline curve.

The spline curve is therefore fully determined by a start point of the route (*SP*), an end point of the route (*EP*), derivatives in *SP* and *EP* and knots. Knots are selected to fit in the remaining points of the route (all the points accept *SP* and *EP*) and additional points obtained as geometric average of two neighbouring points of the route. An example of obtained spline paths for three robots is given in Fig. 4.

The robots will drive on the obtained paths if pure rolling condition of the robot wheels is supposed. This condition is achieved by limitation of the allowed overall acceleration $a = \sqrt{(a_{tang}^2 + a_{rad}^2)}$ which is limited with the friction force and can be decomposed to the tangential acceleration a_{tang} and to the radial acceleration a_{rad}.

$$a_{tang} = \frac{dv}{dt}$$
$$a_{rad} = v \times \omega = v^2 \kappa \quad (1)$$

where κ is curvature defined as

$$\kappa(u) = \frac{x'(u)y''(u) - x''(u)y'(u)}{\left(x'(u)^2 + y'(u)^2\right)^{3/2}} \quad (2)$$

and derivatives in Eq. (3) are with respect to parameter u.

Because the gravity centre of the robot with differential drive is on certain height above ground level, the limit of the tangential acceleration differs from the limit of the radial acceleration. When accelerating in linear direction, a part of the robot weight is carried by rear or front slider which results in a smaller limit for the tangential acceleration. Measured acceleration limits are shown in Fig. 5. (for more details see [6]). The overall acceleration should be somewhere inside or on the ellipse if robots are driven time optimally.

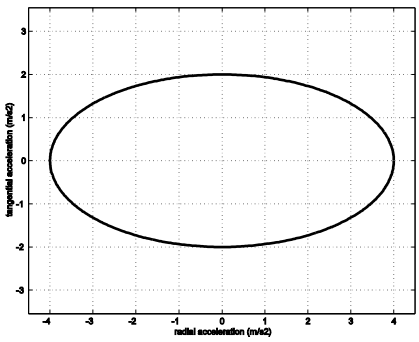


Fig. 6: Acceleration limits

For a given path the velocity profile for each robot is calculated as follows. The local extreme (local maximum of absolute value) of the curvature are determined and named turning points (TP). In these points the robot has to move with maximum allowed speed due to radial acceleration limit. Its tangential acceleration a_{tang} must be 0. Before and after a TP, the robot can move faster, because the curve radius gets bigger than in the TP. Before and after the TP the robot can tangentially decelerate and accelerate respectively as max. allowed by (de)acceleration constraint. In this way the maximum velocity profile is determined for each TP and has the shape of "U" (or "V"). Similarly the maximum velocity profile (due to tangential acceleration/deceleration) is determined for initial point (SP) and final point (EP) velocity respectively. The highest allowable overall velocity profile is determined as the minimum of all the velocity profiles. An example of determined velocity profile is given in Fig. 7.

05. Experiments

Experiments were performed on two polygonal maps representing real environments. The maps were obtained from CAD model of buildings by a semiautomatic procedure. Example of generated trajectories for 3 robots is shown in Fig. 8

The main problem is possible collision of a generated trajectory with obstacles. A partial solution to avoid collision with obstacles is based on map growing by a certain distance and path planning on modified map. It is necessary, that the "growing" distance has to be smaller than visibility distance of the robot in order to guarantee inspection of the whole environment. Points on the shortest path between sensing locations are then placed in certain distance from real obstacles (walls) and therefore a found spline can be traversable by a robot without collision, because there is a free-space around control points.

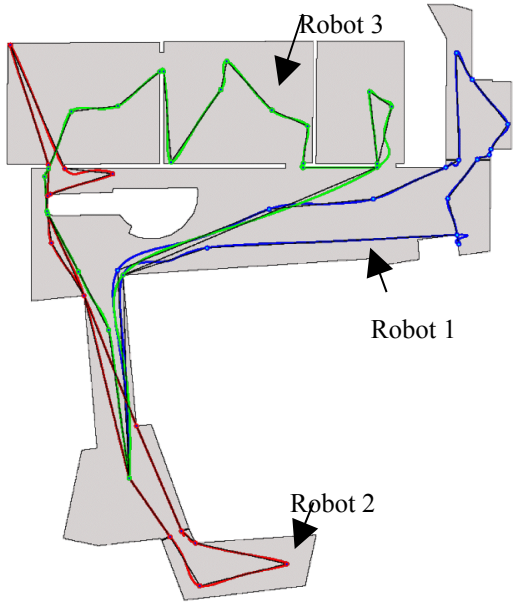
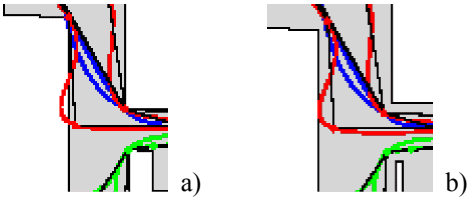


Fig. 9: Spline paths for 3 robots

An example of a collision of a spline curve with a grown map is in Fig. 10a. The same spline curve in original map is shown in Fig. 11b.



a) b)

Fig. 12: Collision of spline curve with wall in the grown map a) and spline curve in the original map b)

An example of the highest allowable velocity profile of the first robot for the spline defined in Fig. 13 is given in Fig. 14.

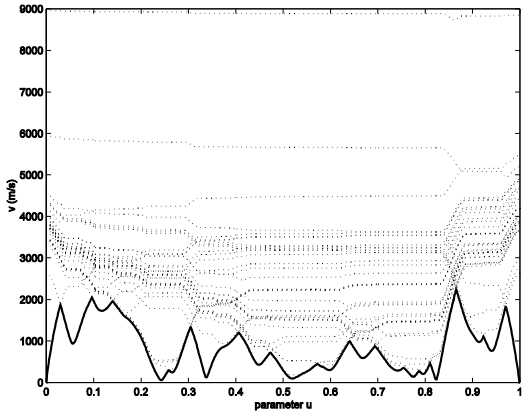


Fig. 15: Highest velocity profile (thicker line) for the first robot

Dotted lines in Fig. 16 are velocity profiles determined for each TPs and the highest allowable overall velocity profile (solid line) is determined as the minimum of all the velocity profiles. A time to complete the mission could be determined for each robot by integrating velocity profile along the path in Fig. 17. The first robot needs 12.60 *s* to travel 11.83 *m* long path, the second robot needs 14.93 *s* to travel 11.39 *m* long path and the third robot needs 16.71 *s* to travel 12.74 *m* long path.

06. Conclusion

Generation of paths for an inspection task for group of mobile robots with respect to traversability was proposed. A path is represented as a spline curve. The problem is decomposed into three sub-problems that are solved separately. One of the main issues is spline curve generation according to obstacles. This curve allows mobile robots to drive through all sensing locations efficiently. For the obtained spline curve the maximum allowable velocity profile due to acceleration limits is determined.

Future improvements to this problem could be based on specifying additional criterion in trajectory optimization procedure. Such criterion has to integrate both possible free-space of the environment and allowable acceleration limits at the same time. Future improvements could also introduce a feed-back from the trajectory generation to the sensing locations placement algorithm. Basically, some of sensing locations should be moved from the original positions to new positions of a designed spline curve with respect to time-optimality. However, the whole environment must still be completely covered from sensing locations.

A0cknowledgement

The work has been supported within Czech-Slovenian intergovernmental S&T cooperation programme for 2005-2006 under project no. 10 ”Cooperative mobile robots for industry and service”. The first author acknowledges the support of the Czech Technical University in Prague, grant CTU 0505413. The support of the Ministry of Education of the Czech Republic, under the Project No. 1M6840770004 to Miroslav Kulich is also gratefully acknowledged.

References

[7] B. J. Nilsson. *Guarding art galleries: Methods for mobile guards*, Ph.D. dissertation, Department of Computer Science, Lund University, 1994.

[8] I. Bjorling-Sachs, D. Souvaine. *A tight bound for guarding general polygons with holes*, Laboratory for Computer Science Research, Hill Centre for the Mathematical Sciences, Busch Campus, Rutgers University, New Brunswick, New Jersey, 1991.

[9] H. H. González-Baños, J. C. Latombe. Planning robot motions for range-image acquisition and automatic 3d model construction, in *AAAI Fall Symposium*, 1998.

[10] M. Kulich, J. Faigl, J. Kléma, J. Kubalík. Rescue operation planning by soft computing techniques, in *IEEE 4th International Conference on Intelligent Systems Design and Application*, Piscataway: IEEE 2004, pp. 103–108, 2004.

[11] M. Kulich, M., J. Faigl, , L. Přeučil. Cooperative Planning for Heterogenous Teams in Rescue Operations, *IEEE International Workshop on Safety, Security and Rescue Robotics*, International Rescue System Institute, Kobe, Japan, June 6-9, 2005.

[12] M. Lepetič, G. Klančar, I. Škrjanc, D. Matko, B. Potočnik. Time optimal planning considering acceleration limits, *Robotics and Autonomous Systems*, vol. 45, pp. 199-210, 2003.

[13] The MathWorks Inc., Spline Toolbox User’s Guide, Version 2, 1999.