A manipulator control in an unknown environment

To cite this article: Pavel Lopatin 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **537** 042079

IOP Conf. Series: Materials Science and Engineering **537** (2019) 042079 doi:10.1088/1757-899X/537/4/042079

A manipulator control in an unknown environment

Pavel Lopatin

Department of Informatics and Computing Techniques, Reshetnev Siberian State University of Science and Technology, 660037, Krasnoyarsk, Krasnoyarskiy rabochiy av., 31, Russian Federation

E-mail: robot-2006@yandex.ru

Abstract. An algorithm for a *n*-link manipulator movement amidst unknown static obstacles in a continuous space is presented. Given theorem stating that if the manipulator moves according to the algorithm will be discovered in a finite number of steps whether a given target configuration is reachable or not. The number, shapes and dispositions of obstacles may be arbitrary.

1. Introduction

In this article we propose an algorithm for a solution of the following Problem: a manipulating robot (MR) moving from a start configuration $\boldsymbol{q}^0$ amidst unknown obstacles, using limited information from its sensor system (SS), should in a finite number of steps discover whether a given target configuration $\boldsymbol{q}^T$ is reachable or not. We will call the $\boldsymbol{q}^T$ reachable if it satisfies both conditions: 1) it is not forbidden; 2) it may be reached from a $\boldsymbol{q}^0$ in a finite number of steps only moving from one not forbidden configuration to another. A configuration is forbidden if it intersects with obstacles or does not satisfy constructive limitations.

There are graph searching methods [1-3] which may solve the Problem. It is easier to use such methods in the case where we have full information about free and forbidden configurations before the beginning of the movement. A computer may then calculate a preliminary path and after that the MR may execute this path. But in case of unknown obstacles the MR has to investigate its environment and plan its path alternately. Then the following difficulty often arises: suppose we have just finished considering the vertices adjacent to a vertex $\boldsymbol{q}$ and we have to consider vertices adjacent to a vertex $\boldsymbol{q}$' and the $\boldsymbol{q}$ and $\boldsymbol{q}$' are not adjacent. In order to consider vertices adjacent to $\boldsymbol{q}$' the MR at first has to come to the $\boldsymbol{q}$'. So we get a problem of the MR movement from $\boldsymbol{q}$ to $\boldsymbol{q}$'. The necessity of searching and executing paths for multiple different $\boldsymbol{q}$ and $\boldsymbol{q}$' makes the total sum of the MR movements very big. In case we plan a path in known environment a computer simply switches its “attention” from $\boldsymbol{q}$ to $\boldsymbol{q}$', which are stored in the computer’s memory.

Currently an approach based on sampling-based algorithms is being actively developed. On the first stage nodes in robot’s configuration space are chosen by some method. On the second stage the nodes are connected by edges and as a result we get a graph [1]. In various publications different methods for choosing nodes on the first stage and connecting them by edges on the second stage are proposed. The sampling-based approaches usually achieve resolution completeness, meaning that they will find a solution if one exists, but may run forever if one does not, or probabilistic completeness, meaning that the probability tends to one that a solution is found if one exists (otherwise, it may still run forever) [4]. This approach may be developed to an effective one for known environment when we have full information about obstacles in advance. But in unknown environment the nodes (configurations) and edges we have generated may intersect with obstacles and methods how to choose new nodes and edges in order to solve the Problem in a finite number of steps were not proposed. It is possible to outline such representatives of the sampling-based approach as algorithms based on randomized potential field, Ariadne's Clew algorithm [1, 2], probabilistic roadmaps [5], rapidly-exploring random trees [6-9], expansive-space trees[10, 11].

In [12] we gave an algorithm solving the Problem in the continuous configuration space. It was supposed that the MR's SS system may supply information about free and forbidden points from a small neighborhood of the MR's configuration space points. In this article we consider more general form of the neighborhood.

2. Task Formulation and algorithm

2.1. Preliminary Considerations

1) The MR's movement should take place in a hyperparallelepiped X defined by inequalities

$$\boldsymbol{a}^1 \leq \boldsymbol{q} \leq \boldsymbol{a}^2, \tag{1}$$

where $\boldsymbol{q}\in$X, $\boldsymbol{q}=(q_1, q_2\ldots, q_n)$ – vector of generalized coordinates, n – number of MR's links, $\boldsymbol{a}^1$ - vector of lower limitations on the values of generalized coordinates, $\boldsymbol{a}^2$ - vector of higher limitations. A configuration $\boldsymbol{q}$ will be considered as allowed (not forbidden) if it satisfies both conditions: 1) it has no common points with any obstacle; 2) it satisfies constructive limitations that is no prohibited intersection of links occurs and $\boldsymbol{q}\in$X. X is continuous. The disposition, shapes and dimensions of the obstacles do not change during the whole period of the MR movement. Their number may not increase.We suppose that $\boldsymbol{q}^0\in$X and $\boldsymbol{q}^{\mathrm{T}}\in$X.

2) MR has a SS which is turned on in every path changing point $\boldsymbol{q}^i\in$X, i=0, 1, ... and supplies information about a hyperball with the centre in the $\boldsymbol{q}^i$ and with a radius r > 0. Let us call such hyperball as r-hyperball. The value of r is such that the r-hyperball includes all points $\boldsymbol{q}$ which are immediately adjacent to $\boldsymbol{q}^i$ and all points $\boldsymbol{qq}$ immediately adjacent to every point $\boldsymbol{q}$. So it is possible to inscribe into the r-hyperball an ε-hyperball with the centre in $\boldsymbol{q}^i$ and with a radius 0<ε<r. The case when the r-hyperball consists only from $\boldsymbol{q}^i$ and the points $\boldsymbol{q}$ is inappropriate, such r-hyperball is too small. Consider that SS supplies information about every point from the r-hyperball whether it is allowed or forbidden and this information is exact and reliable. Exact information means that if MR asks SS to supply information about a point $\boldsymbol{q}^*$ then SS supplies information namely about $\boldsymbol{q}^*$, not about $\boldsymbol{q}^{**}\neq \boldsymbol{q}^*$. Reliable information means that if SS tells that a point $\boldsymbol{q}^*$ is forbidden then it is really forbidden and if SS tells that a point $\boldsymbol{q}^*$ is allowed then it is really allowed. Further the exact and reliable information we will simply call reliable. A set Y($\boldsymbol{q}^i$) consists from all points of the r-hyperball with the centre in $\boldsymbol{q}^i$. SS writes all points reliably defined as forbidden from the Y($\boldsymbol{q}^i$) into a set Q($\boldsymbol{q}^i$), and all points reliably defined as allowed - into a set Z($\boldsymbol{q}^i$).

When MR is in $\boldsymbol{q}^i\in$X, i=0, 1, ... its SS may also supply information about a Y1-neighborhood of $\boldsymbol{q}^i$. Consider that the Y1-neighborhood is a convex set of points, for example ellipse. It should be possible to inscribe in it the r-hyperball with the centre in $\boldsymbol{q}^i$. The set Y1($\boldsymbol{q}^i$) consists from all points of the Y1-neighborhood. The origin of a set Y1($\boldsymbol{q}^i$) is in the point $\boldsymbol{q}^i$. The shape and size of the Y1($\boldsymbol{q}^i$) should be the same for any $\boldsymbol{q}^i$, i=0,1, … That is if we have a set Y1($\boldsymbol{q}^i$) with an origin in a point $\boldsymbol{q}^i$ and a set Y1($\boldsymbol{q}^j$) with an origin in a point $\boldsymbol{q}^j$ then the Y1($\boldsymbol{q}^j$) has the same number of points as the Y1($\boldsymbol{q}^i$) and if $(q_1^j, q_2^j, \ldots, q_n^j)=(q_1^i+b_1, q_2^i+b_2,\ldots, q_n^i+b_n)$, where $b_1,b_2,\ldots,b_n$ –some real numbers, then for every point $\boldsymbol{q}'\in$Y1($\boldsymbol{q}^j$) there will be one and only one point $\boldsymbol{q}''\in$Y1($\boldsymbol{q}^n$) such that $(q_1{}', q_2{}', \ldots, q_n{}')=(q_1{}''+b_1, q_2{}''+b_2, \ldots, q_n{}''+b_n)$. In other words we get Y1($\boldsymbol{q}^j$) by displacing the Y1($\boldsymbol{q}^i$) on the vector $\boldsymbol{b}=(b_1, b_2\ldots, b_n)$.

The information supplied by SS about the points which are beyond the r-hyperball and inside the Y1-neighborhood may be not reliable. SS writes all points reliably defined as forbidden from the Y1($\boldsymbol{q}^i$) into a set Q1($\boldsymbol{q}^i$), and all points reliably defined as allowed - into a set Z1($\boldsymbol{q}^i$). MR stays in a path changing point $\boldsymbol{q}^n$ until the set

$$\mathrm{FRBDN} = (\bigcup_{i=0}^{n} \mathrm{Q}(\boldsymbol{q}^i)) \bigcup (\bigcup_{i=0}^{n} \mathrm{Q1}(\boldsymbol{q}^i))$$

is formed. So the set FRBDN is the set of all points about which MR has reliable information that they are forbidden. The sets Y($\boldsymbol{q}^i$), Q($\boldsymbol{q}^i$), Z($\boldsymbol{q}^i$), Y1($\boldsymbol{q}^i$), Q1($\boldsymbol{q}^i$), Z1($\boldsymbol{q}^i$) may be written using one of such methods like formulas, lists, tables and so on, but we suppose that we have such method. We will not consider the SS structure.

3) We have a program PI($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$, FRBDN, X) which solves a PI problem. The PI problem is: in a finite number of steps either generate a path L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) satisfying the following conditions:

3.1. L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) connects $\boldsymbol{q}^n$ and $\boldsymbol{q}^{\mathrm{T}}$;

3.2. L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) $\cap$ FRBDN = Ø;

3.3. L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) $\in$ X

in a case if at least one path satisfying conditions 3.1-3.3 exists in X or discover that no path satisfying conditions 3.1-3.3 exist in X. $\boldsymbol{q}^n$, n=0,1,… is a path changing point (see Algorithm), $\boldsymbol{q}^{\mathrm{T}}$ is the target point. The condition 3.2 means that no point from L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) should coincide with any point from FRBDN. The condition 3.3 means that every point from L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) should satisfy (1). If PI() succeeds in generating a L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) it returns 1, if it fails – it returns 0. We propose to use an existing algorithm for the procedure PI() (see, for example, [13]), or a specially developed one.

4) MR executes any path L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) in the following way. Consider that MR is in a point $\boldsymbol{q}^* \in$ L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$). MR gets reliable information about the point $\boldsymbol{q}^{**} \in$ L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) which is the next after $\boldsymbol{q}^*$. If $\boldsymbol{q}^{**}$ is allowed then MR moves in $\boldsymbol{q}^{**}$, if $\boldsymbol{q}^{**}$ is forbidden then MR stays in $\boldsymbol{q}^*$.

2.2. The Algorithm for Manipulators' Control in the Unknown Environment

We will denote the points where generation of a new path occurs as $\boldsymbol{q}^n$, n=0, 1, … We will call such points "path changing points". Before the Algorithm work n=0 and $\boldsymbol{q}^n = \boldsymbol{q}^0$.

Algorithm

1	The MR is in $\boldsymbol{q}^n$. Turn on SS and form the set FRBDN.

2	/*If the attempt to generate a path L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) is unsuccessful*/
	if (PI($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$, FRBDN, X)=0)
	   return ($\boldsymbol{q}^{\mathrm{T}}$ is not reachable);
	endif
	/*Otherwise go to Step 3 to execute the path*/

3	MR begins to execute the path L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$). There may be two results of execution: 1) MR comes to the point $\boldsymbol{q}^c \in$L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) preceding $\boldsymbol{q}^{\mathrm{T}}$ and discoveres that $\boldsymbol{q}^{\mathrm{T}}$ is not forbidden. In this case: MR moves in $\boldsymbol{q}^{\mathrm{T}}$; return($\boldsymbol{q}^{\mathrm{T}}$ is reachable). 2) MR comes to a point $\boldsymbol{q}^c \in$L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) and discovers that the next point $\boldsymbol{q}^* \in$L($\boldsymbol{q}^n$, $\boldsymbol{q}^{\mathrm{T}}$) is forbidden. In this case: n:=n+1; $\boldsymbol{q}^n$:=$\boldsymbol{q}^c$; go to 1.

Theorem. If MR moves according to the Algorithm it will solve the Problem in a finite number of steps.

Proof. Before the Algorithm considering let us formulate the following Question: "Have we received the information that $\boldsymbol{q}^{\mathrm{T}}$ is reachable/unreachable?" There may be three answers to the Question: 1) "Yes, we have received the information that $\boldsymbol{q}^{\mathrm{T}}$ is reachable"; 2) "Yes, we have received the information that $\boldsymbol{q}^{\mathrm{T}}$ is unreachable"; 3) "No, we have not received the information whether $\boldsymbol{q}^{\mathrm{T}}$ is reachable or unreachable yet".

IOP Conf. Series: Materials Science and Engineering **537** (2019) 042079 doi:10.1088/1757-899X/537/4/042079

Consider at first the case when the Y1-neighborhood of every path changing point $\boldsymbol{q}^n$, n =0,1,2,… has the shape of the r-hyperball.

In Step 1 of Algorithm one may see path changing points $\boldsymbol{q}^n$, n=0, 1, 2, … When the MR is in such point the procedure PI() is called. The task of PI() is to generate a path L($\boldsymbol{q}^n$, $\boldsymbol{q}^T$) leading to $\boldsymbol{q}^T$ because there is a hope that MR will not meet earlier unknown forbidden points and then MR will solve the Problem. In case when PI() fails to generate a path we get the answer (2), if PI() has generated a path L($\boldsymbol{q}^n$, $\boldsymbol{q}^T$) we get the answer (3) and Algorithm goes to Step 3. On Step 3 MR begins to execute the L($\boldsymbol{q}^n$, $\boldsymbol{q}^T$) according to the item 4 of the Preliminary Considerations. If MR executing L($\boldsymbol{q}^n$, $\boldsymbol{q}^T$) does not meet a forbidden point then it comes to the allowed point $\boldsymbol{q}^T$ and we get the answer (1). An execution of any path L($\boldsymbol{q}^n$, $\boldsymbol{q}^T$) is fulfilled in a finite number of steps because the length of any L($\boldsymbol{q}^n$, $\boldsymbol{q}^T$) is finite. But it may happen that MR will come to such a point $\boldsymbol{q}^c \in$L($\boldsymbol{q}^n$, $\boldsymbol{q}^T$), that the next after $\boldsymbol{q}^c$ point $\boldsymbol{q}^* \in$ L($\boldsymbol{q}^n$, $\boldsymbol{q}^T$) is forbidden. In this case MR makes: $n := n + 1$; $\boldsymbol{q}^n := \boldsymbol{q}^c$ that is $\boldsymbol{q}^c$ becomes a new path changing point and Algorithm goes to Step 1. Let us show that the number NPATH of the path changing points $\boldsymbol{q}^n$, n=0, 1, 2,…,NPATH-1 will be finite and all of them will be different.

Let us prove that all path changing points will be different. Suppose that the MR changed a path being in a point $\boldsymbol{q}^s$, and later it again changed a path, being in a point $\boldsymbol{q}^p$, that is $s<p$. Let us show that $\boldsymbol{q}^s \neq \boldsymbol{q}^p$. Suppose, at first, that, on the contrary, $\boldsymbol{q}^s = \boldsymbol{q}^p$. Then Q($\boldsymbol{q}^s$)=Q($\boldsymbol{q}^p$). When the MR was in $\boldsymbol{q}^s$, it generated a path which did not intersect with the sets Q($\boldsymbol{q}^i$), i=0, 1, ... , s. When the MR reached the point $\boldsymbol{q}^p$, it discovered that it was necessary to change the path that is this path intersected with the set Q($\boldsymbol{q}^p$). But Q($\boldsymbol{q}^p$)=Q($\boldsymbol{q}^s$) and Q($\boldsymbol{q}^s$) was taken into account when the path which brought MR to $\boldsymbol{q}^p$ was generated. It means that the MR cannot come to a point of the path changing $\boldsymbol{q}^p$ which will be equal to any other point of the path changing and it means that all points where the MR changes its path are different.

Now let us show that the number of path changing points is finite. Suppose that it is infinite. All path changing points 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 $\boldsymbol{q}^i$, 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 $\boldsymbol{q}^i$, $i>s$ will lie in an ε-neighborhood of $\boldsymbol{q}^s$. Let us take ε<r. Consider an arbitrary path changing point $\boldsymbol{q}^i$ lying in the ε-neighborhood of $\boldsymbol{q}^s$. As far as in $\boldsymbol{q}^i$ the MR had to change the path, it means that among the points immediately adjacent to $\boldsymbol{q}^i$ the MR met earlier unknown forbidden points. But all points immediately adjacent to $\boldsymbol{q}^i$ belong to the r-hyperball of $\boldsymbol{q}^s$, and forbidden points immediately adjacent to $\boldsymbol{q}^i$ belong to Q($\boldsymbol{q}^s$). But Q($\boldsymbol{q}^s$) ought to be taken and therefore was taken into account when generation of paths L($\boldsymbol{q}^i$, $\boldsymbol{q}^T$) occurred for every $j \geq s$. So, assuming that the number of path changing points is infinite we got impossible situation – some path changing points will lie in the ε-neighborhood of another path changing point. Therefore the number NPATH of path changing points is finite.

So, the number of path changing points $\boldsymbol{q}^n$, n=0,1,2,…, NPATH-1 is finite and they are all different. Consider that the MR came to the last path changing point $\boldsymbol{q}^{\mathrm{NPATH-1}}$. It means that the MR has done a finite number of steps moving from $\boldsymbol{q}^0$ and still has the answer (3) to the Question. When MR is in $\boldsymbol{q}^{\mathrm{NPATH-1}}$ PI() is called on Step 2. If PI() fails to generate a path we have the answer (2), if PI() has generated a path we get the answer (3). The MR begins to execute the path L($\boldsymbol{q}^{\mathrm{NPATH-1}}$, $\boldsymbol{q}^T$). As far as the path changing point was the last, MR will not meet earlier forbidden points, will reach $\boldsymbol{q}^T$ in a finite number of steps because the length of L($\boldsymbol{q}^{\mathrm{NPATH-1}}$, $\boldsymbol{q}^T$) is finite, will discover that $\boldsymbol{q}^T$ is allowed, the answer (1) will be received. So the answer whether $\boldsymbol{q}^T$ is reachable or unreachable will be received in a finite number of steps.

We proved the theorem for the case when the Y1-neighborhood of every path changing point $\boldsymbol{q}^n$, n=0,1,2,… is the r-hyperball. But one may see that the proof will be valid for any Y1-neighborhood satisfying the Preliminary Considerations – the number of path changing points will be finite, all of them will be different, but we should write into FRBDN all those points about which SS supplied reliable information that they are forbidden. The theorem is proved.

So, it was shown that the number NPATH of the path changes is finite, and therefore one may see that the solution of Problem is reduced to a solution of a finite number of problems of a path planning in the presence of known forbidden states with its subsequent execution. One may see that the Algorithm may be considered as a sampling-based algorithm, where sampling is made by using the procedure PI() and when the path in a point $\boldsymbol{q}^n$, n=0,1,2,… intersected with an earlier unknown forbidden state MR replenishes FRBDN by those points about which we have reliable information that they are forbidden and generates a new path satisfying conditions 3.1-3.3.

3. Conclusion

An algorithm for a *n*-link manipulator movement amidst arbitrary unknown static obstacles was presented. The obstacles number, shapes and dispositions may be arbitrary. The MR's sensor system supplies information about a neighborhood of every path changing point in predefined volumes. Given a theorem stating that if MR moves according to the algorithm it in a finite number of steps discover whether a target configuration is reachable or not.

References

[1] Choset H, Lynch K M, Hutchinson S, Kantor G, Burgard W, Kavraki L E and Thrun S 2005 *Principles of Robot Motion. Theory, Algorithms and Implementations* (The MIT Press)
[2] LaValle S M 1999-2006 *Planning Algorithms* Available from: http:// msl.es.uiuc.edu/planning
[3] Nilsson N J 1971 *Problem-Solving Methods in Artificial Intelligence* (New York: McGraw-Hill Book Company)
[4] LaValle S M 2011 *IEEE Rob. and Autom. Soc. Mag*. **18(1)** 79-89
[5] Ekenna D, Uwacu S, Amato N 2015 *Machine Learning in Planning and Control of Robot Motion Workshop (IROS-MLPC)* (Hamburg, Germany) Available from: https://parasol.tamu.edu/groups/amatogroup/publications.php
[6] Chaabani M, Bellamine S and Gasmi M 2014 *Int. J. on Soft Computing, Art. Intelligence and Applications (IJSCAI)* **3(3/4)** 1-17
[7] Devaurs T and Cortes J 2014 *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS) Sep 2014 Chicago United States* Available from: https://hal.archives-ouvertes.fr/hal-01057030
[8] McMahon T, Thomas S and Amato N 2015 *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA) May 2015 Seattle Washington* pp 2977-84 Available from: https://parasol.tamu.edu/groups/amatogroup/publications.php
[9] Vasilopoulos V, Vega-Brown W, Arslan O, Roy N and Koditschek D E 2018 *IEEE Int. Conf. on Robotics and Automation (ICRA '18)* May 2018 Brisbane pp 5683-90
[10] Haschke R 2012 *Workshop on Robot Motion Planning: Online, Reactive, and in Real-time. 2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2012 October 7-12 Vilamoura, Algarve, Portugal* pp 5-9
[11] Hsu D, Kindel R, Latombe J and Rock S 2002 *Int. J. of Robotics Research* **21(3)** 233-55
[12] Lopatin P K 2010 *Vestnik Scientific J. of Siberian State Aerospace University named after academician M.F.Reshetnev* **7(33)** 80-4
[13] Canny J 1988 *The Complexity of Robot Motion Planning* (Cambridge: The MIT Press)