Switching formation of robotic swarms in occluded environments using sampling-based algorithms

Sreejeet Maity
Indian Institute of Science
Bangalore, India
sreejeetm@iisc.ac.in

Nilay Srivastava
Indian Institute of Science
Bangalore, India
nilays@iisc.ac.in

Mohit Kumar Gupta
Indian Institute of Science
Bangalore,India
mohitg@iisc.ac.in

***Abstract*—In this paper, we introduce a sampling-based algorithm to reconfigure the classical multi-robot motion planning problem to realize a wide range of significant formations and internal switching patterns. A swarm of geometrically identical robots does not call for a unique target for each agent, and as a result of that any possible permutations of the target configurations are mathematically valid, and complexity reduces by a consequential amount. Thereby, with the help of the above-mentioned class of algorithms, we explored and successfully implemented specific formations of our interest. The contributions can prove handy in many real-life industrial problems such as object transportation in occluded environments in the presence/absence of static obstacles ensuring zero agent-agent and obstacle-agent collision.**

***Index Terms*—swarm robotics, motion-planning, sampling, switching formation**

I. Introduction

A multi-robot system is a collection of robotic units that work in cooperation to solve problems of various interests. One of the crucial aspects of swarms are their abilities to configure them in various configurations. This feature enables the agents to implement a certain class of co-operative tasks. Apart from that, it is to be remembered that the central theme of motion planning is to find collision-free paths for each agent of a swarm through an occluded environment in presence of obstacles.

Before we dive in the actual problem, let us take a moment to discuss the nature of the agents in our swarm. A swarm is not necessarily composed of only identical agents, rather there could be an amalgamation of different types of geometrically identical robots. We will specifically reserve the variable 'k' to describe the number of different types of geometrically similar agents that are present in the swarm. However, in this paper we have considered the unlabelled case k = 1 for all our simulations and we have used identical translating disc robots to achieve our goal of formation switching.

We have specifically dealt with a sampling based algorithm named UPUMP introduced in [1], which allows us to sample random pebbled graphs[1] and connect them in a way such that it ensures no robot moving through this constructed edges ever collides . On top of that , we are able to retrieve paths between any two possible configurations through the sampled pebble graphs.In this paper, we explicitly tried out a class of switching formations by keeping UPUMP as the central theme for motion planning of the disc robots.

II. Mathematical Preliminaries

A. Primordial terminologies

We are starting with a robot r which is supposed to work in a workspace W.In this paper, we have considered the geometry of the workspace to be rectangular. However, a slight change in code will allow us to define any possible workspace of our interest. Let C $=\{c_1,c_2,...,c_m \mid c_i \epsilon F\}$ be a set of 'm' single robot configurations be called configurations. Let configuration C occupied by the robots be called as placements. Now, we denote by $F(r)$ the free space of a robot r: the set of all single-robot configurations that are collision free [1].Given s, $t \ \epsilon \ F(r)$, there exists a path for the robot r from s to t is a continuous function $\pi : [0,1] \rightarrow F(r)$, such that $\pi(0)$ = s, $\pi(1)$ = t. In order to extend our argument of free space , we can assume with mathematical certainty that the geometrically identical robots share the same free space. Mathematically, given two identical robots r and r_1, their free space are respectively equal to $F(r)$=$F(r_1)$.

Let R = $\{r_1,r_2,...,r_m\}$ be a set of m geometrically identical robots in the workspace W. And we define $r(c) \subset$ C, for c ϵ F, to represent the section of the workspace which is covered by a robot r ϵ R placed in the single-robot configuration c.Note that two robots from R collide, when placed in c,c' ϵ F, if r(c)∩r(c') $\neq \phi$.

B. Unlabelled multi robot motion planning

In an unlabelled multi robot motion planning we are starting with a unlabelled problem U = (R, S, T), where R ($|R|$=m) be the set of m geometrically identical robots and S= $\{s_1,s_2,...,s_m\}$,T=$\{t_1,t_2,...,t_m\}$ be the start and target placements respectively ensuring $|S|$=$|T|$=m. The goal of the problem is to find the path π_u=$\{\pi_1,...,\pi_m\}$ where, π_i is the path for the i^{th} robot such that $\pi(0)$=$s_i \epsilon S$ and $\pi(1)$=tϵ T.

C. Pebble motion problem

Before we introduce the focal algorithm in the next section, we will take a moment to discuss a tweaked version of a pebble motion problem. In pebble motion problem, we place 'm' pebbles in a graph $G(V, E)$ with $|V| = n$ and we take n>m.We are allowed to move pebbles along the edges of the same graph which allows us to achieve a series of configurations. However, we are adding one additional constraint such that only one pebble can move at a time.

In a broader mathematical sense, let us consider two placements P_1={p_{11},...,p_{1m}} and P_2={p_{21},...,p_{2m}}. These two placements are valid in a pebble motion problem if (p_{1i},p_{2j})ϵ E and (p_{1k}=p_{1l})$\forall$ $k \neq i$ and $l \neq j$.

We can check whether a pebble motion problem has a solution or not by introducing a simple mathematical test which we will be calling as signature throughout this paper [1]. Let V' be a pebble placement of a pebble problem $P(G,S,T,m)$ and let $\{G_1$, ... , $G_h\}$ be the set of maximal connected subgraphs of G, where G_i = (V_i, E_i).The signature of V' is defined as $\{|V_i \cap V'|\}_{i=1}^{h}$. Now, we claim that the equivalency of two placements V',V" on a pebble graph G is given by signature(V', G)=signature(V'', G).

Continuing from the mathematical premise described in the last section we can say For every pebble problem $P(G,S,T,m)$ such, there exists a pebble path from S to T if and only signature(S,G)=signature(T,G). However, from now on, we will use a simple equivalency symbol to represent equal signatures i.e we can re-frame the above sentence by saying there exists a path between S and T if $S \equiv T$.

III. Switching Formation using UPUMP

The major contribution of our paper was to use the agents in the swarm to realize various formations of our interest.This section is reserved for anatomizing the construction and the algorithms which helped us to achieve the same.In the next section, we have explicitly shown the effectiveness of our algorithm by featuring a subset of our results .

A. Formation Generation

We start by constructing our desired formation set as follows F_g={A,|,M,N,C,λ,ϕ}, where each $x \epsilon F_g$ is a configuration essential to form the particular formation with $|x|$=m (number of robots).Each x is formed within a bounding box of area scaled by a fraction of the configuration space.After generation, this formation is randomly rotated and shifted somewhere in the configuration space.

B. Application of UPUMP

In this section, we will introduce the algorithms that we designed specifically to solve the unlabelled motion planning problem using UPUMP. We are using disc robots to switch from one formation to another which is equivalent to finding a path in an unlabelled motion planning problem. Since, this is an unlabelled motion planning problem, we can use UPUMP algorithm to find a path for a switching formation problem

a) Generating Pebble Graphs: We are starting by generating a pumped configuration, which is a configuration PC such that $|PC|$= n $\geq$ m . We then use the EDGE PLANNER mechanism to form edges between the vertices generated by the pumped configurations.Let there be two vertices v_1,V_2 $epsilon$ PC then there exists an edge(v_1, v_2) which can also be denoted by $\pi_{v_1 v_2}(\theta)$.

We need to ensure r($\pi_{v_1 v_2}(\theta)$)$\cap$ r(u)=ϕ where uϵ PC and u$\neq$$v_1$, v_2

b) Connecting Pebble Graphs: This section gives a brief discussion about the two algorithms namely CONNECT and CONNECTION GENERATOR. The first algorithm (CONNECT) generates all possible edges between two pebble graphs. While the second one(CONNECTION GENERATOR) removes the interfering edges that can cause a collision between the robots.The algorithm transforms the problem of finding paths between pumped configurations into the problem of finding an independent set in a graph. We generate the set of pairs D=(v, v')|v ϵ V, v'ϵV',$\pi_{v,v'}$ $\neq$ ϕ.We say that two pairs (v,v'),(u,u') ϵ D if there exists θ ϵ[0,1] such that robot rϵR placed in $\pi_{v,v'}(\theta)$ collides with another robot r'ϵR placed in $\pi_{u,u'}(\theta)$. All the valid edges are then added to the roadmap graph H.

c) Query: We will use CONNECT algorithm defined in the previous section to check for connection between start and target vertices through the sampled pebble graphs.

d) Retrieve Path: We deployed three major algorithms in this section . GENERATE CONNECTIVITY MATRIX algorithm returns a matrix M such that $M_{ij}\epsilon\{0,1\}$. When $M_{ij} = 1$ it indicates that there is a connection between i^{th} graph and j^{th} graph else there is no connection between them. The first index in the M represents the start graph and the last index represents target graph.

The second algorithm that we have devised is the WAY FINDER. This algorithm returns the indices in the matrix M that are connected to each other by applying the standard method of back-tracing.The algorithm returns the indices of the graph that are connected to the start and target graphs.

The third algorithm, PATH RETRIEVAL uses the way obtained from the WAY FINDER to find the edges in the roadmap graph H for the respective graphs given by the WAY FINDER.

IV. Experimental Setup

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence.

V. Results and Discussions

TABLE I
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when

Fig. 1. Example of a figure caption.

writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization $\{A[m(1)]\}$", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

Acknowledgment

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks . . .". Instead, try "R. B. G. thanks. . .". Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

Please number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.