Collecting a Swarm in a 2D Environment Using Shared, Global Inputs

Shiva Shahrokhi and Aaron T. Becker

***Abstract*— Consider a swarm of robots initialized in a 2D planar grid world where each position is either free-space or obstacle. Assume all robots respond equally to the same global input, such as an applied fluidic flow or electric field. This paper investigates what input sequence should be applied to collect all the robots to one position.**

I. Introduction

This paper builds on the techniques for controlling many simple robots with uniform control inputs presented in [1]–[3], and outlines new research problems.

a) Definitions: The 'robots' in this paper are simple particles without autonomy. A planar grid *workspace* W is filled with a number of unit-square robots (each occupying one cell of the grid) and some fixed unit-square blocks. Each unit square in the workspace is either *free*, which a robot may occupy or *obstacle* which a robot may not occupy. Each square in the grid can be referenced by its Cartesian coordinates $\boldsymbol{x} = (x, y)$. All robots are commanded in unison: the valid commands are "Go Up" (u), "Go Right" (r), "Go Down" (d), or "Go Left" (l).

We consider two classes of commands, discrete and maximal moves. *Discrete moves*: robots all move in the commanded direction one unit unless they are prevented from moving by an obstacle or a stationary robot. *Maximal moves*: robots all move in the commanded direction until they hit an obstacle or a stationary robot. For maximal moves, we assume the area of W is finite and issue each command long enough for the robots to reach their maximum extent. A *command sequence* $\boldsymbol{m}$ consists of an ordered sequence of moves m_k, where each $m_k \in \{u, d, r, l\}$ A representative command sequence is $\langle u, r, d, l, d, r, u, \ldots \rangle$.

We consider two notions of a collected swarm: if robots are allowed to overlap, the swarm is connected when all robots share the same (x, y) coordinates. If the robots may not overlap, the swarm is collected when it forms one 4-connected component.

Research questions include

1) What connected topologies allow collecting a swarm?
2) What connected topologies do not allow collecting a swarm?
3) What connected topologies can be collected with a given command sequence?

S. Shahrokhi and A. Becker are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204-4005 USA {sshahrokhi2, atbecker}@uh.edu

II. Related work

A. Robot Rendezvous

Rendezvous usually concerns two independent, intelligent agents that must meet at a predetermined time.

B. Manufacturing

This is similar to work on draining a polygon [4], but with discrete inputs.

Or injection-molding (22 years ago).

C. Localization

Or localizing a blind robot in a known map.

maybe similar to localizing a robot with minimum travel, but this moves the robot and has it take additional measurements [5]

D. Robot Gathering

todo: scholar search on this keyword

E. Topologies that do not allow collecting a swarm

Robots initialized in two unconnected components i and j of a freespace cannot be collected. The proof is trivial, since robot in freespace i can not reach freespace j.

Under maximal inputs, the freespace can be constructed with spaces resembling bottles or fish weirs from which a single robot cannot escape, as shown in Fig. [TODO]. If the freespace contains at least two such bottles with at least one robot in each, the swarm cannot be collected.

The world must be bounded. Two initially-separated robots in an unbounded world without obstacles cannot be collected, however with discrete inputs, one obstacle is sufficient [6].

F. connected topologies allow collecting a swarm

With discrete inputs and robots that are allowed to overlap, the problem can be reduced to localizing a sensorless robot in a known workspace. This is similar to work on draining a polygon [4], but with discrete inputs.

Algorithm 1, brute-force-search: initializes a tree with the root node $\{C_0, \}$, where C_0 is the current configuration of possible robot locations, where C_0 is an ordered vector containing the locations of every possible location for a robot in the workspace and $\{\}$ is the shortest corresponding movement sequence to achieve this configuration. We then construct a breadth-first tree of possible configurations $\{u, r, d, l\}$, pruning leaves that already exist in the tree. We stop with the cardinality at a leaf $C_i = 1$, which indicates that the swarm has been collected (equivalently, the robot has been localized).

Algorithm 2, greedy-approach:

Algorithm 1 BlindLocalizationBruteForce(W, C_0

Require:

Ensure: $C_{\boldsymbol{m}} \leftarrow \text{ApplyMoves}[\boldsymbol{m}, C_0, W]$, $|C_{\boldsymbol{m}}|= 1$

1: $T \leftarrow C_0$,

2: **return** (r_1, r_2)

1) possible configurations = current config
2) initial cardinality = current config
3) construct a breadth-first tree of possible configurations $\{u, r, d, l\}$ prune leaves that already exist.
4) stop when the cardinality of possible positions in any leaf is less than initial cardinality

REFERENCES

[1] A. Becker, E. Demaine, S. Fekete, G. Habibi, and J. McLurkin, "Reconfiguring massive particle swarms with limited, global control," in *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, Sophia Antipolis, France, Sep. 2013, pp. 51–66.

[2] A. Becker, E. Demaine, S. Fekete, and J. McLurkin, "Particle computation: Designing worlds to control robot swarms with only global signals," in *IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong: IEEE, May 2014, pp. 6751–6756.

[3] A. Becker, E. D. Demaine, S. P. Fekete, G. Habibi, and J. McLurkin, "Reconfiguring massive particle swarms with limited, global control," in *Algorithms for Sensor Systems*, ser. Lecture Notes in Computer Science, P. Flocchini, J. Gao, E. Kranakis, and F. Meyer auf der Heide, Eds. Springer Berlin Heidelberg, 2014, vol. 8243, pp. 51–66. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-45346-5_5

[4] G. Aloupis, J. Cardinal, S. Collette, F. Hurtado, S. Langerman, and J. ORourke, "Draining a polygon—or—rolling a ball out of a polygon," *Computational geometry*, vol. 47, no. 2, pp. 316–328, 2014.

[5] G. Dudek, K. Romanik, and S. Whitesides, "Localizing a robot with minimum travel," *SIAM Journal on Computing*, vol. 27, no. 2, pp. 583–604, 1998.

[6] A. Becker, G. Habibi, J. Werfel, M. Rubenstein, and J. McLurkin, "Massive uniform manipulation: Controlling large populations of simple robots with a common input signal," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, Nov. 2013, pp. 520–527.