Autonomous Identification of Local Agents in Multi-Agent Robotic Swarms

Aaron Roggow
aaron.roggow@valpo.edu

Georges El-Howayek
georges.el-howayek@valpo.edu

Sami Khorbotly
sami.khorbotly@valpo.edu

Electrical and Computer Engineering Department
Valparaiso University, Valparaiso, IN 46385

Abstract – **With the emergence of swarm intelligence and evolutionary algorithms, system designers are creating robotic swarms of continuously increasing sizes. As the size of a swarm increases, it is imperative to have a uniform program collectively downloaded on all the agents in it. These uniform programs impede the designer's ability to designate distinct identities for various agents in the swarm.**

In this work we propose an algorithm that, when implemented in a robotic swarm, allows the locally interacting agents to autonomously designate unique identifications to each other at run time. The results show that when used on a small swarm of autonomous robots, running the algorithm led to a steady state where every agent in a local neighborhood was assigned a unique identification value.

I. Introduction

It may be somehow challenging to offer a precise definition of a multi-agent system because, as eloquently stated in [1], many competing, mutually inconsistent answers have been offered in the past. The imprecise definition, however, is that a multi-agent systems consists of multiple interacting computing elements [2]. The agents in a multi-agent system can be computers, software agents, or robots, among many other possibilities. Robotic swarms are only a small class in the world of multi-agent systems. The main characteristic of a robotic swarm is the obvious fact that the agents are exclusively robots. Moreover, while a generic robotic multi-agent system may consist of a few robots, it usually takes a significantly larger number of robots for the group to be considered a swarm. The main purpose of the swarm is to use the collective effort of individual agents to achieve a goal, which a single or few members of the swarm cannot accomplish on their own [3]. Success depends heavily on the collaboration and coordination between the different swarm members. Therefore, it is important for a robot to determine who are they collaborating with and it is desirable to have a methodology to ensure unique identifications to local agents.

In smaller multi-agent robotic systems such as a robotic soccer team [4] or a robotic football team [5], heterogeneous robots are used to perform various functionalities. For example, the goalie would be equipped with algorithms that are different from those of a field playing robot and a quarterback will be significantly different from a kicker. However, with the larger number of robots in a swarm, swarm designers are inclined to use homogeneous robots with identical functionalities and identical programs. The homogeneity in a robotic swarm makes it possible for the swarm designer/operator to simultaneously download the same program into all the robots in the swarm, also known as the common input signal method [6]. This capability is extremely important as it eliminates the need for tedious individual programming and manipulation of every single robot in a swarm that may comprise hundreds or even thousands of robots [7].

While there are great advantages associated with the use of a common input signal to control and program a robotic swarm, the one drawback associated with it is that it eliminates the flexibility of characterizing the robots and makes it impossible to identify which robot does what at run-time. For example, it will not be possible for the swarm designer to create a pilot robot or a leader of the swarm unless it is programmed separately with a different code.

The robot characterization issue can be more problematic when implementing an algorithm that requires a robot to communicate with a specific number of other fellow robots. For example, if an arbitrary robot in a certain application, needs to receive the approval of n neighboring fellow robots to proceed with an action. It will be practically impossible for that robot to distinguish the reception of n distinct signals from the repeated reception of the approval signal from the same robot.

In order to address this issue, we have developed an algorithm that enables the local agents in a swarm to autonomously assign unique identities to one another. In this context, two agents are considered to be local if they are either directly connected to one another or have at least one agent that is connected to both of them. Also in this context, two agents are said to be connected if they are within communication range from one another. For example, in the configuration shown in Figure 1, agents A & B are connected to one another. On the other hand, agents A and D are not connected but they are local since they are both connected to agent B. Agents C and F are neither connected nor local.

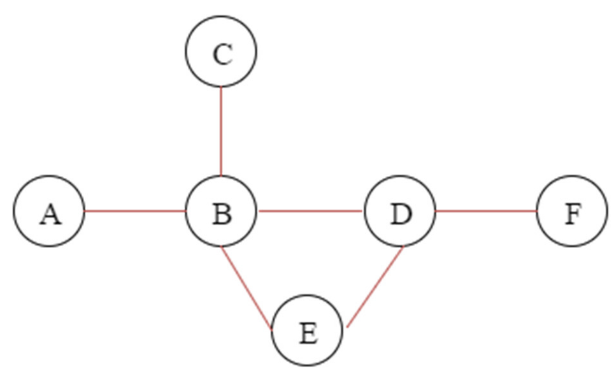

Figure 1. A sample configuration in a multi-agent system.

The rest of the paper is organized as follows: section II discusses commercially available robotic swarm platforms where this algorithm may be beneficial. Section III introduces and explains the suggested algorithm. Section IV discusses the experimental setup that was used for testing. The obtained results are presented in section V and the paper is concluded with the summary and suggested future work in section VI.

II. Robotic Swarm Platforms

As discussed earlier, the need for identifying local agents in a swarm arises mainly when using a homogeneous robotic swarm that is programmed with a common input signal. A variety of such robotic swarm platforms have been developed by research groups and are commercially available.

To our knowledge, the commercial development of swarm robotic platforms started with the Khepera robot swarm [8] in the early 1990s at the École Polytechnique Fédérale de Lausanne (EPFL). Since that time, Khepera has evolved into Khepera II, III, and the most recent version Khepera IV that is currently produced and distributed by the K-team [9]. Another commercially available platform is the Jasmine micro-robotic swarm that was first developed at the University of Stuttgart in 2004 [10]. This platform is currently produced by the swarm robot project [11]. The most recent robotic swarm platform, and the one used in this work, is the kilobot swarm that was originally developed by the self-organizing systems research group at Harvard University [12]. Just like the Khepera family, kilobots are currently produced and distributed by the K-team [9].

III. The Identification Algorithm

The goal of the suggested algorithm is to ensure that a unique ID is assigned to every local agent. In other words, no agent in the swarm is connected to more than one agent with the same ID value. The flowchart shown in Figure 2 describes the operation of the algorithm. Each agent starts by generating a random ID for itself. The agent then broadcasts its ID to inform all the connected agents that it is using that particular ID value. Simultaneously, agents receive all the broadcasted ID values from their connected neighbors. Any agent that detects multiple neighbors using the same ID, will generate a flag for the conflicting ID(s). The agent will then proceed to check if its own ID is flagged, in which case it will go back to the starting point where a new ID is generated. For example, in figure 1, if agents A and C are using the same ID, agent B is expected to detect the conflict and flag that ID.

If the agent's ID is not flagged, the agent will proceed to compare it with the IDs of its connected neighbors. If an ID match is detected, the agent will have to generate a new ID to resolve the conflict. A good example of this situation in figure 1 would be agents B and C having the same ID. Seeing how no agent is connected to both these conflicting agents, the conflicting ID will not be flagged. That is why having the agents compare their IDs with those of their neighbors is extremely important. The process is iteratively repeated until there no more flag messages are generated and no matching IDs are detected.

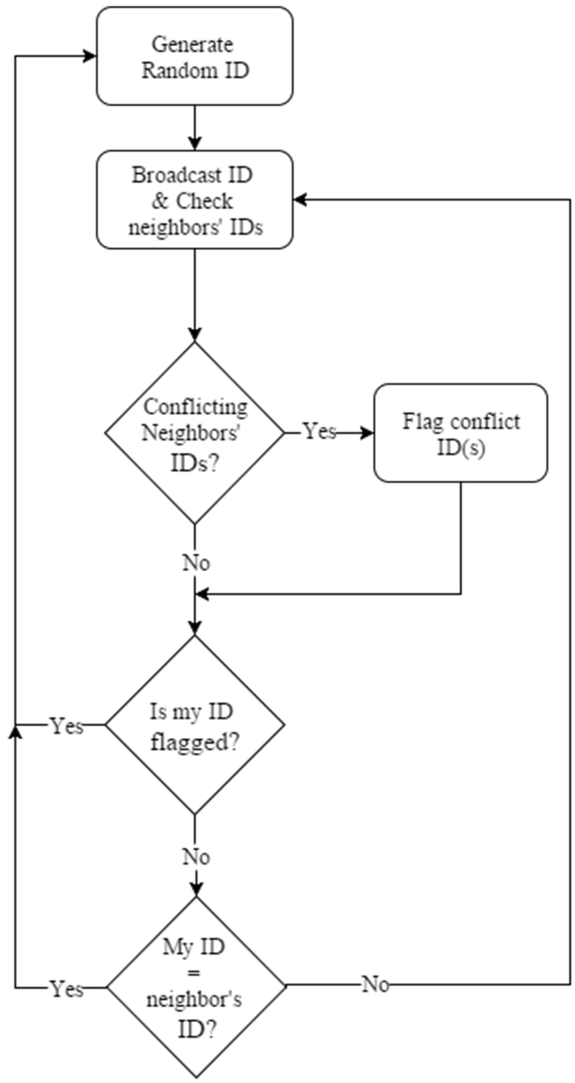

Figure 2. The flowchart of the identification algorithm.

IV. The Experimental Setup

In order to experimentally verify the functionality of the algorithm, an experimental setup was created using the formerly discussed kilobots. The kilobots were chosen because of their relatively low cost and their suitability to implement swarm algorithms. A kilobot, shown in Figure 3, is a cylindrically shaped robot with a diameter of 33 mm and a height of 34 mm, not including the detachable battery charging hook on top of it. Two vibration motors are housed on the sides of the kilobot. The two motors are independently controlled to allow for clockwise rotation, counter-clockwise rotation, or forward motion through vibration of the legs. Kilobots are also equipped with ambient light sensors to measure the brightness of the light in their environment.

An infrared (IR) transmitter/receiver pair, facing downwards on the bottom of the kilobot board, enables it to communicate with neighboring kilobots by bouncing IR signals off of the ground surface. The communication range can vary with the reflectiveness of the ground surface but is generally in the range of 5-10 cm.

The communication system allows kilobots to, not only exchange information with their neighboring kilobots, but also estimate their distance to them based on the strength of the received signal. This distance will be used in the algorithm implementation to differentiate between messages from the same robot and messages from two different robots. For instance, when a broadcasted ID is received, the kilobot uses the associated distance to determine if this ID is a re-broadcast from the same robot or it is from a different robot using the same ID as another neighbor.

The same IR communication system is also used to program a swarm of kilobots with a common input signal kilobot controller board. This controller board when suspended above the kilobots ground surface, as shown in Figure 4, can program all the kilobots within a 1 m diameter below the controller. A user can also use the controller to run/pause the kilobots at any time during their operation.

Each kilobot is equipped with an atmega 328 microcontroller with 32Kbytes of program memory and 1Kbyte of EEPROM. In the algorithm, the kilobots store the received IDs (with their associated distances) in a FIFO stack to identify any possible conflict between their neighbors. As more ID receptions are stored, the oldest data is overridden. This serves both to preserve memory, as well as to recycle and reuse all the IDs that have not been broadcasted for a period of time. Were an agent retain all of its receptions, it is likely that it will eventually have data points for all possible IDs, including the one that has been abandoned by the agents. This generates an infinite loop of false flags, leading to continuously changing IDs. For this purpose, our reception stack was limited to store the latest 10 received IDs.

A Red, Green, Blue (RGB) LED is mounted on the top of the kilobot's board and can be used to visually indicate its run time state. The LED is also used to indicate a low battery to get the user's attention. In order to test the functionality of the suggested algorithm, the LEDs on each kilobot is used to display its individual identity. With the three color components in an RGB LED, it is possible to theoretically display one of 8 colors as shown in Table I. For example, a kilobot with ID =2 will have the values (R = 0, G = 1, and B = 0) leading to the LED displaying a Green light on the LED.

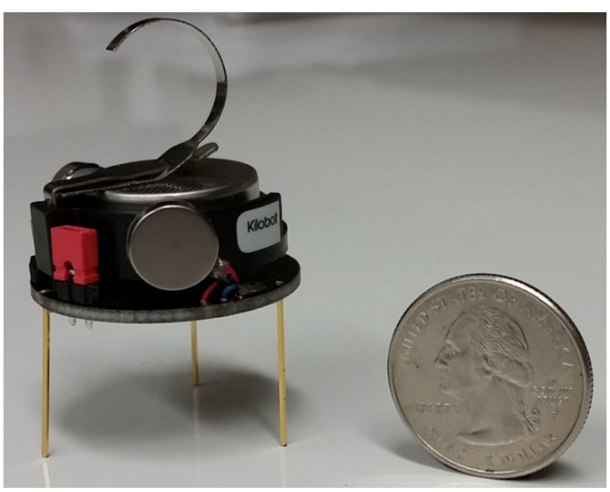

Figure 3. One of the kilobots used in testing the algorithm.

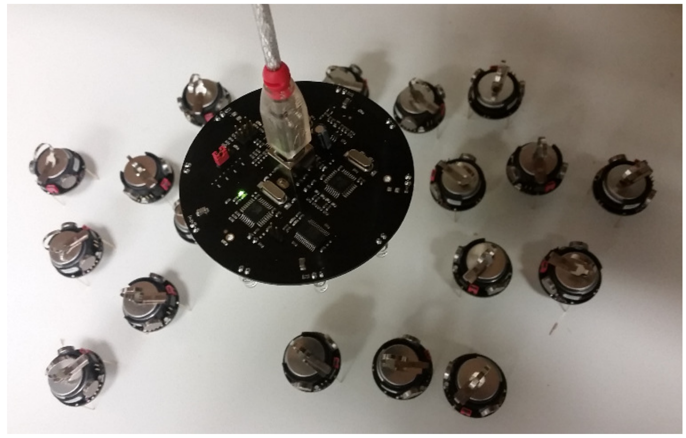

Figure 4. A group of kilobots being programmed by a suspended controller board.

Table I. The 8 possible ID values and their corresponding colors

ID value	R	G	B	Color
0	0	0	0	No light
1	0	0	1	Blue
2	0	1	0	Green
3	0	1	1	Cyan
4	1	0	0	Red
5	1	0	1	Magenta
6	1	1	0	Yellow
7	1	1	1	White

V. Results

The algorithm was tested in various swarm configurations. In one configuration, shown in figure 5, the kilobots are clustered in small disconnected groups of 2-5 kilobots in each group. The ID of each kilobot can be construed from the color of their LED. Moreover, for the sake of readers of monochrome versions of this paper, the figure was modified by superimposing the ID of each kilobot next to it in the image. When the program is installed on the kilobit agents, the system starts in a transient mode where the ID values of the different agents change in a pseudo-random operation as described in section II. The number of cycles consumed in the transient mode is statistically dependent on the maximum number of agents in a locality and the number of possible ID values that an agent can take. At the end of the transient mode, the system moves to the steady state mode where no two local agents have the same ID value. As can be seen in Figure 5, the 15 agents are clustered in 4 different groups with the following ID values: [1, 2, 3, 4, 5], [2, 5, 6, 7], [2, 7], and [1, 2, 3, 4].

In another experiment, the kilobots are organized in a strongly connected graph configuration, where every agent is reachable from every other agent in the swarm. Since the swarm includes more than 7 agents (the maximum number of available ID values), it is imperative for multiple agents have the same ID. Nevertheless, as shown in figure 6, the algorithm successfully led to unique ID values for all the local agents.

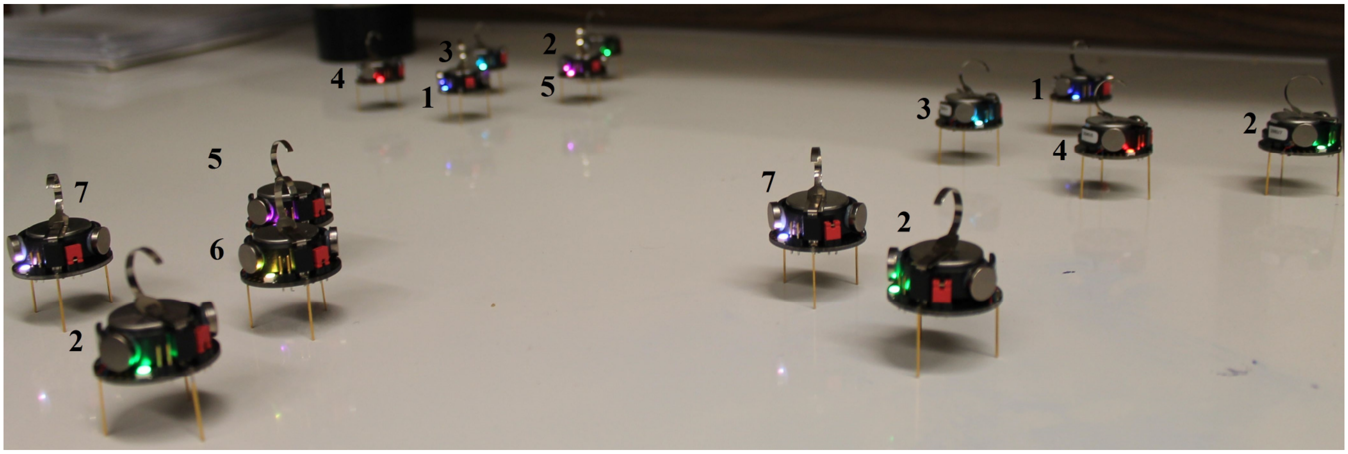

Figure 5. The Steady state identifications of all the agents in a swarm organized in small clusters configuration. This figure is best seen in color. A color version is available online.

Figure 6. The Steady state identifications of all the agents in a swarm organized in a strongly connected graph configuration. This figure is best seen in color. A color version is available online.

VI. Conclusion & Future Work

In this paper, we introduced an algorithm that can be used to autonomously assign unique identities to local agents. The algorithm can be uniformly installed on all agents using a common input signal. The algorithm was implemented and tested on the kilobot robotic swarm platform. The experimental results showed that, after a few iterations, the algorithm successfully led to a steady state where no local agents shared the same ID value.

An idea that may be investigated in further research is to have an agent store and transmit their ID data from previous iterations in addition to their current data. As the IDs are randomly generated, it is unlikely that agents sharing a current ID will also share an identical ID history. Further investigation is needed to quantify the efficiency of using such a methodology in minimizing the number of required iterations while also considering the additional burden of storing and broadcasting multiple ID values.

VII. References

[1] Y. Shoham and K. Leyton-Brown, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press. 2009.

[2] M. Wooldridge, An Introduction to MultiAgent Systems. John Wiley & Sons. 2009.

[3] I. Navarro and F. Matia, "An introduction to swarm robotics," International Scholarly Research Notices on Robotics. vol. 2013, article 608164, 2013.

[4] C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, and A. Sgorbissa, "Coordination among heterogeneous robotic soccer players," *IEEE International Conference on Intelligent Robots and Systems*, 2000.

[5] A. Stephon and S. Khorbotly, "A Camera-Based Target Tracking System for Football Playing Robots", *IEEE Southeast Symposium on System Theory,* Jacksonville, FL, March 2012.

[6] A. Becker, G. Habibi, J. Werfel, M. Rubenstein, and J. McLurkin, "Controlling Large Populations of Simple Robots With a Common Input Signal", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, Nov. 2013.

[7] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable Self-Assembly in a Thousand-Robot Swarm", *Science*, Vol 345, no 6198, 15 Aug 2014.

[8] F. Mondada, E. Franzi, and A. Guignard, "The development of Khepera," in *Proceedings of the 1st International Khepera Workshop*, vol. 64 of *HNI-Verlagsschriftenreihe, Heinz Nixdorf Institut*, 1999.

[9] http://www.k-team.com/mobile-robotics-products/khepera-iv.

[10] S. Kornienko, O. Kornienko, P. Levi, "Generation of desired emergent behavior in swarm of micro-robots", The European Conference on Artificial Intelligence, Valencia, Spain, 2004.

[11] http://www.swarmrobot.org/

[12] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A Low Cost Scalable Robot System for Collective Behaviors", *IEEE International Conference on Robotics and Automation*, St. Paul, MN, May, 2012.