ISSN:0975-9646

Palash Shrawak et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 8 (1) , 2017, 116-119

Robotic Algorithm Development

Palash Shrawak, Tanzeem Kagzi , Ayush Pratap Singh, Brijesh Dobariya, Prof. Pranali Lokhande, Prof. B.R. Alhat

Dept of Computer Engg,MIT Academy of Engineering,Alandi(D.), Pune

Abstract—Robotics is a stream which has not been explored thoroughly. It is having possible application in many fields. As we all know many robots and automated systems has been developed with the help of Robotics and Artificial Intelligence that have been installed in different industrial sectors. In this project, we will implement a system which will be helpful for robots to satisfy some requirements like - they will cover the whole area where they will be deployed and without losing network connection with each other, they can be accumulated automatically without any centralized unit controlling them, they will properly explore the area and will accurately map the area and finally they will be able to identify the obstacles in that area and will try to avoid them. This will be a challenge as the robots will be deployed in an unknown area and there may be obstacles that may vary in size and shape. We are going to use various tools and simulators to test them.
General Terms—Swarm Robotics[7, 6], Dispersion[9, 8]

Keywords—Wireless Communication, Synchronization, Routing Protocols, Simulation, Message Sending

1. INTRODUCTION

Deploying large numbers of small simple robots in a decentralized swarm like fashion is gaining recognition and popularity in many problem domains. One of the primary issues is how the swarm of robots will move[7]. A common task is to spread out and cover an unknown area as thoroughly and quickly as possible in order to setup a sensor network for surveillance. This may be useful in areas hostile to humans such as disaster or military zones, or even planetary exploration.

In a swarm approach each robot is small, simple, and executes the same software program. Swarm methods bring many advantages. Since they are decentralized there is no single point of failure that can bring the entire system down. In fact many of the robots can fail and the swarm system will still function. Due to the simplicity of the robots they will be cheap to manufacture, another reason they are expendable. The designer of a swarm hopes that through individual actions based on local decisions the swarm as a whole will produce the intended emergent behavior. As in the natural world, unexpected emergent behavior may even occur.

A common theme in robot dispersion research is a connection to behavior in the natural world, whether it be of a biological, chemical, social, or physics based behavior. In particular there are many approaches that draw on the Luke Ludwig and Maria Gini general concept of repulsive and attractive forces between robots. There are subtle differences between these approaches, but the general concept is the same; move away from neighboring robots, but not too far away[5]. All of these approaches assume that the relative locations (distance and bearing) of neighboring robots is available to the robot through its sensors.

This can be obtained using a 360o laser range finder or an omnidirectional camera. Size is the limiting factor. The robots must be large enough to carry the laser range finder or have enough processing power to analyze the images from the camera which is inherently processor intensive[3]. A common laser range finder, the SICK LMS 200, has dimensions of 15 cm x 18 cm x 15 cm. This is a rather large payload for a robot such as the University of Minnesotas Scout, which is only 11 cm long by 4 cm wide. Small robots have the advantage of being cheaper, simpler, and less noticeable. Bob Grabowski from Carnegie Mellon has classified existing small robots by size in his Small Robot Survey, and at least half of the robots in this survey would be incapable of sporting either a 360o laser range finder or an omni camera.[2]

The question we attempt to answer is whether or not a swarm of small robots can be dispersed effectively without knowing the relative locations of neighboring robots. In particular, we use wireless signal intensity as an approximation of distance to assist in the dispersion. This requires a wireless 802.11 card on the robots, which is considerably smaller than the 360o laser range finder.

Theoretically, signal intensity varies according to the law of inverse signal propagation, which simply means the signal intensity is proportional to the inverse square of the distance it travels. In a practical setting, the environment plays a huge role by providing obstacles that cause noise in this signal. However, it is unnecessary for the signal intensity to be very accurate in order to provide some indication to a robot for which way it should travel.[4, 1] The primary property needed is for the signal intensity to decrease over time as the distance between robots increases, and vice-versa.

2. LITERATURE SURVEY

Fernando Diaz-del-Rio and Daniel Cagigas-Muiz, presented the paradigm of Multi-Agent Systems (MAS). Multi-Agent Systems come in the form of robots coordinating in a team, sensor networks based on mobile robot system, and robots in Intelligent Environ ment [1]. For many of the complex robotic systems , a solution based on Multi-Agent Systems is the most suitable. Multiple algorithms like SLAM, Bayesian estimation, motion planning etc. are used over multiple frameworks like ROS, OROCOS etc. This include heterogeneous mobile robot teams, robots working in ambient intelligence environment, and mobile sensor networks based on robots. It is noteworthy that, as shown in this work, RSFs and MASFs provide distinct and different features for the development of MARS, due to their differences in the areas to which they have been applied till now. On one hand, MASFs provide a basic framework to build MAS by focusing on features such as agent mobility, inter-action among agent patterns, and ontologies Robotics software evolution reflects General-Purpose Software Engineering (GPSE).

Sarah Bergbreiter and K.S.J. Pister presented the CotsBots are inexpensive and modular mobile robots built entirely from commercial off-the-shelf components [2]. These robots give a convenient platform on which to investigate algorithms, cooperation, and distributed sensing in large robot networks. An off-the-shelf robot platform for distributed robotics has been shown. The CotsBots provide an extremely flexible hardware and software platform for user. Hardware is bought off-the-shelf and only very simple modifications are necessary to build a CotsBots system. It is hoped that the CotsBots will provide a simple, easy-to-use platform for research in distributed robotics.

Serguei Vassilvitski, Mark Yim presented a reconfiguration algorithm [3] for metamorphic robots made of Telecubes, six degree of freedom cube shaped modules presently being developed at PARC. The reconfiguration algorithms for metamorphic systems have missed at least one of the desired properties: local decision making, completeness of reconfiguration or parallel execution. We have presented an algorithm which has all of the three qualities and is guaranteed to run in place of worst case quadratic time. We resort to the use of meta-modules, as otherwise the space of possible configuration can be divided into classes with no configuration possible between members of different classes.

Ali Turan and Seta Bogosyan describes the development of a client-server communication method using the TCP/IP protocol to run Matlab/Simulink compatible motion control units operating on the remote server side [4]. The developed method is currently in use to allow access to a Hardware-In-the-Loop (HIL) robot simulator made for both on-site and remote use. The client - server communication is developed in C/C++ using wx Widgets to communicate with the Matlab/Simulink downloadable DS1104, which is used to develop different variety of robot dynamics configurations and control algorithms on the robotic simulator. The major contribution in this study is the development a client-server communication method for remote experimentation on a DS1104 driven HIL simulator setup.

Although the developed communication currently serves the HIL simulators in the laboratory, it could be used to run any type of Matlab/Simulink based application remotely; i.e. robotic systems driven by Matlab/Simulink compatible controllers as well as computer simulations prepared in Matlab and Simulink. The developed system not only allows the remote user to experiment with existing robot dynamics and control algorithms made available in the remote laboratory, but also gives the user the capability to experiment with the robot configuration and control algorithm of his/her choice, hence providing a low-cost and flexible solution for robotics, mechatronics and control engineering related education and research.

We discuss the fundamental problems and practical issues underlying the deployment of a swarm of autonomous mobile robots that can potentially be used to build mobile robotic sensor networks [8]. For the purpose, a geometric approach is proposed that allows robots to configure themselves into a two-dimensional plane with uniform spatial density. Particular emphasis is paid to the hole repair capability for dynamic network reconfiguration. Specifically, each robot interacts selectively with two neighboring robots so that three robots can converge onto each vertex of the equilateral triangle configuration. Based on the local interaction, the self-configuration algorithm is presented to enable a swarm of robots to form a communication network arranged in equilateral triangular lattices by shuffling the neighbors [5, 6, 10]. Convergence of the algorithms is mathematically proved using Lyapunov theory. Moreover, it is verified that the self- reparation algorithm enables robot swarms to reconfigure themselves when holes exist in the network or new robots are added to the network. Through extensive simulations, we validate the feasibility of applying the proposed algorithms to self-configuring a network of mobile robotic sensors.

A natural problem that arises in the study of ”swarm robotics” is how to effect a fast dispersal and filling of an environment by a group of robots moving according to a set of local rules [9]. We consider two versions of the problem, in which the environment R may be discrete or continuous. A discrete environment is composed of unit squares (pixels) that are induced by the integer grid within a polygonal domain. There is at most one robot per pixel and robots move horizontally or vertically at unit speed.

A continuous environment is modeled as a polygonal domain (polygon with holes). The robot swarm begins as a dense confguration of points with the domain and then spreads out as each robot executes a local strategy based on crude sensor data it acquires about its immediate surroundings.

Inspired from the complex behaviors observed in natural swarm systems (e.g., social insects and order living animals), Swarm Intelligence (SI) is a new field that aims to build fully distributed de-centralized systems [10] in which overall system functionality emerges from the interaction of individual agents with each other and with their environment. As a result to try applying the insight gained from this domain research into multi-robotics, an emerging research area called Swarm Robotics (SR) has been issued.

Dispersing swarms of robots to cover an unknown, potentially hostile area is useful to setup a sensor network for surveillance.Previous research assumes relative locations (distance and bearing) of neighboring robots are available to each robot through sensors [5, 7]. Many robots are too small to carry sensors capable of providing this information. We use wireless signal intensity as a rough approximation of distance to assist a large swarm of small robots in dispersion. Simulation experiments indicate that a swarm can effectively disperse through the use of wireless signal intensities without knowing the relative locations of neighboring robots.

SR is the study of how to coordinate large groups of relatively simple robots through the use of local rules. It focuses on studying the design of large amount of relatively simple robots, their physical bodies and their controlling behaviors.

Multi-Robot Systems (MRS) are born to overcome the lack in information processing capability and many other aspects of single robots that are not capable to dial with special tasks; which, in order to be efficiently completed, need cooperation and collaboration between groups of robots.

Applications like semi-automatic space exploration (Visentin et al., 2001), rescue (Casper et al., 2000), or underwater
exploration (Ayers et al., 1998) need robust and flexible robotic systems. Most of these applications require systems combining the following three basic characteristics: [11]
—**Robustness** Unstable, very complex or extreme environments require robustness to severe hardware failures.
—**Versatility** The complexity of the task needs versatility in hardware shape and functionality. The robot has to perform well in very different terrains and in very different tasks such as displacement, exploration or object transportation.
—**All Terrain Navigation** Complex unstructured environments such as distant planets or catastrophic environments need a very flexible and efficient all-terrain navigation.

The SWARM-BOTS project aims at combining swarm intelligence and physical self-assembling features to provide the above mentioned characteristics to a group of robots.

3. METHODOLOGY

CLIQUE INTENSITY ALGORITHMS

We assume the robots have a few small and simple proximity sensors that extend at least a meter (ex. infrared) that allow the robot to avoid most collisions with walls and other robots. We also assume that the robots have a wireless 802.11 card and are capable of obtaining signal intensity measurements with incoming packets. This is a standard requirement of the 802.11 interface[4]. No other sensors are needed during dispersion, although most likely robots will be carrying some form of a camera or other sensor which is meant to be utilized once the sensor network is in place. The processing power on small robots is limited, which makes analyzing images from a camera during real-time motion difficult.

There are many ways to use wireless signal intensity to aid a swarm in dispersing throughout an unknown environment. In comparison to all of the repulsive/attractive dispersion research in which relative distance and bearing of neighboring robots is known, signal intensity gives only a rough approximation of distance and no bearing information. This signal intensity must be tracked over time to determine which direction the robot should move. In a swarm of robots, each one may be in contact with many neighbors at a time[6]. If it is known that one of the neighbors was stationary, then a robot could specifically reference the stationary robots intensity and attempt to move in a direction of decreasing signal intensity until some threshold is reached. This is a key concept in this algorithm.

The Clique Intensity Algorithm is designed for a distributed homogeneous swarm, therefore the algorithm operates and runs from the perspective of a single robot in the swarm. The knowledge of each robot is a graph with robots as nodes and signal intensities between robots as weights. This graph is referred to as the connectivity graph. Robots share portions of their connectivity graphs with their neighbors such that each robot has the knowledge it needs to execute the algorithm. A clique is a graph or subgraph in which every node is connected to every other node. A maximal clique is not a subgraph of another clique. For each maximal clique in the connectivity graph a single robot is chosen to be the sentry for the clique, meaning it remains stationary. The other robots in the clique attempt to move away from the sentry, which is done by monitoring the change in the signal intensity over time. Each robot behaves in such a way that causes the entire swarm to disperse in an attempt to create cliques in the connectivity graph of size three or two. This is an attempt to triangulate the map which is known to be the most effective static configuration for the area coverage problem[7]. The algorithm is roughly composed of five basic steps.
—Update connectivity graph from neighbors shared knowledge.
—Share edges incident on me with neighbors.
—Find all maximal cliques that I am in.
—Determine the sentry for each maximal clique.
—Choose and apply behavior based on sentries, cliques, and connectivity graph.

The Behaviours followed by the Robots are as follows :-

—**Avoid Collisions Behavior**: Utilize proximity sensors to avoid collisions.

—**Seek Connection Behavior**: Go in reverse for a bit and if this doesn't work pivot and move forward.

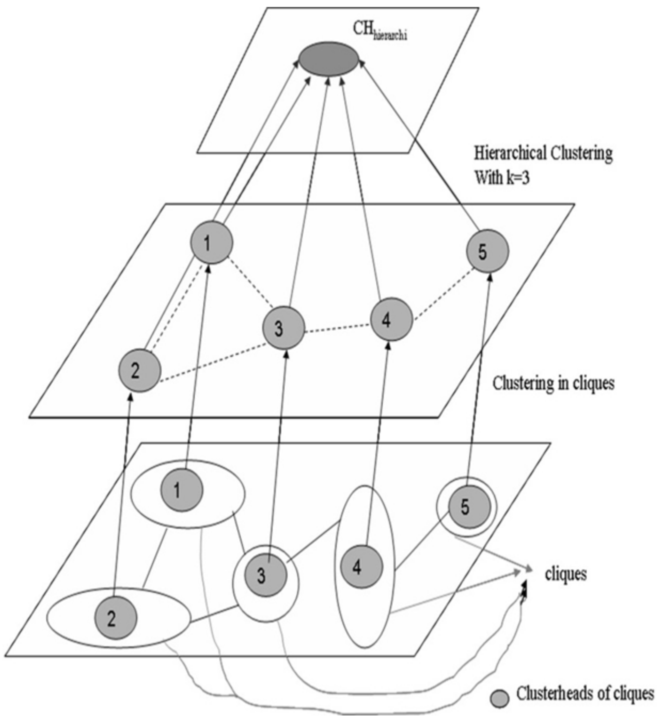
—**Disperse Behavior**: if my sentry intensity is decreasing over time then go straight otherwise pivot for a bit and then move forward, and check for decreasing sentry intensity again.

—**Guard Behavior**: Don't move.

Each robot in motion needs a sentry from which it monitors the signal intensity over time to determine which way to move. The primary decision to make is whether or not the robot is a sentry, and if not then the robot must decide which neighbor will be its sentry. This decision is made individually by each robot examining its connectivity graph and following a set of rules. The rules are structured such that each robot will arrive at the same decision as to which are sentries and which are in motion. Communication between robots of a bartering nature could be used to resolve the decision of which ones are sentries[7]. This was not done since an attempt was made to avoid communication overhead and to keep the algorithm as simple as possible. The only communication between robots is the sharing of knowledge described above.

4. Architecture Of Project

In this architecture, the root node i.e. CH hierarchi form a connectivity graph using the signal intensities. From the connectivity graph, number of clusters have been formed. These clusters then finally, form cliques (minimum size of 2). A clique is a subgraph that has every node connected to every other node. These cliques are then used to form sentries which helps the swarm in dispersion.



Sr. No.	Ideal Distance	RSSI From Robot	Distance From Source
1	5m	-50	5.05m
2	5m	-52	5.10m
3	5m	-49	4.80m
4	5m	-52	5.10m
5	5m	-48	4.77m

5. Conclusion

The results obtained from the experiments indicate that a swarm of small robots can be dispersed effectively through the use of wireless signal intensities, without knowing the relative locations of neighboring robots. Using small robots instead of larger ones capable of carrying a laser range finder means that many more robots can be deployed for the same cost. It would be interesting to analyze how swarms with limited sensing abilities compare to swarms a fraction of their size with more powerful sensing abilities.

References

[1] Pablo Inigo-Blasco, Fernando Diaz-del-Rio, Carmen Romero-Ternero, Daniel Cagigas-Muniz, ”Robotic Software Frame-works for Multi-Agent Robotic Systems Development ,” Contents lists available at SciVerse ScienceDirect , Robotics and Autonomous Systems 60 (2012) 803821

[2] Sarah Bergbreiter and K.S.J. Pister, ”CotsBots: An off-the-shelf Platform for Distributed Robotics ,” Proceedings of the IEEVRSJ InU. Conference on Intelligent Robots and Systems Las Vegas, Nevada ’ October 2003

[3] Serguei Vassilvitskii, Mark Yim, John Suh, ”A Complete, Local and Parallel Reconfiguration Algorithm for Cube Style Modular Robots ,” Proceedings of the IEEE International Conference on Robotics 8 Automation Washington DC 0-7803-7272-7/02/ May 2002.

[4] Ali Turan, Seta Bogosyan, Metin Gokasan, ”Development of a Client-Server Communication Method for Matlab/Simulink Based Remote Robotics Experiments , ” IEEE ISIE , Montreal, Quebec, Canada , July 9-12, 2006

[5] P. E. Rybski, S. A. Stoeter, M. Gini, D. F. Hougen, and N. Papanikolopoulos , ”Performance of a distributed robotic system using shared communications channels , ” IEEE Trans. On Robotics and Automation, 22(5):713727, Oct. 2002.

[6] M. Batalin and G. S. Sukhatme, ”Coverage, exploration and deployment by a mobile robot and communication network , ” Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks, 26(2):181196, 2004

[7] Luke Ludwig and Maria Gini, ”Robotic Swarm Dispersion Using Wireless Intensity Signals , ” Dept of Computer Science and Engineering, University of Minnesota

[8] Geunho Lee , Nak Young Chong , ”A Geometric Approach to Deploying Robot Swarms , ” School of Information Science, Japan Advanced Institute of Science and Technology

[9] Esther M. Arkin , Michael A. Bender , Sandor P. Fekete , Tien-Ruey Hsiang , Nenad Jovanovic , Joseph S.B. Mitchell , Marcelo Sztainberg , ”Online Dispersion Algorithms of Robot Swarms , ”

[10] Belkacem Khaldi, Foudil Cherif, ”An Overview of Swarm Robotics: Swarm Intelligence Applied to Multi-robotics , ” International Journal of Computer Applications (0975 8887) Volume 126 No.2, September 2015

[11] Francesco Mondada, Giovanni C. Pettinaro, Andre Guignard, Ivo W. Kwee, Dario Floreano, Jean-Louis Deneubourg, Stefano Nolfi, Luca Maria Gambardella and Marco Dorigo, ”Swarm-Bot: a New Distributed Robotic Concept , ”

[12] https://www.quora.com/How-do-I-calculate-distance-in-meters-km-yards-from-rssi-values-in-dBm-of-BLE-in-android

[13] http://stackoverflow.com/questions/13705647/finding-distance-from-rssi-value-of-bluetooth-low-energy-enabled-device

[14] http://beantalk.punchthrough.com/t/get-rssi-value-to-calculate-distance/2488

[15] https://forums.estimote.com/t/use-rssi-measure-the-distance/3665

[16] http://electronics.stackexchange.com/questions/83354/calculate-distance-from-rssi