Formation Control for Multi-Robot Teams Using A Data Glove

Nuttapon Boonpinon and Attawith Sudsang
Department of Computer Engineering
Chulalongkorn University
Bangkok 10330, Thailand
{*nuttapon,attawith*}*@cp.eng.chula.ac.th*

***Abstract*—The Formation control problem is one of fundmental problems in multi-robot applications, such as exploration and mapping. In this paper, we present a distributed formation control algorithm for a group of mobile robots moving in a obstacle filled workspace. The algorithm is based on the concept of a spring force and a potential filed. Our algorithm works under limited sensory information and require no communication between the robot team. We also introduce an intuitive human-multi robot interaction via a data glove. The human operator can control a group formation parameter using only his hand gesture. Preliminary simulation results are presented confirming effectiveness of the presented approach.**

***Index Terms*—Multi-Robot, Formation Control, HRI**

I. Introduction

Recently, it is suggested that multi-robot systems tend to have many advantages over single robot counterparts [11]. A team of robots can handle a wider range of tasks and accomplish some tasks more efficiently than a single robot. Furthermore, deploying a group of robots can provide robustness and fault tolerance. These properties are crucial in some applications such as military operations [17] and rescue missions [6]. From economic perspective, the cost of building many simple robots is significantly lower than the cost of a large and complex monolithic robot. From these advantages, it comes as no surprise that Multi-Robot Systems are gaining more popularity both in research communities and practical applications.

Formation control is a classic problem in Multi-Robot Systems. Many formal definitions of formation control can be found in Multi-Robot literature [1], [2]. Although these definition were defined independently, they share some common features. Generally speaking, formation control is a coordination between a group of robots to get into certain shapes and to maintain the formation during group movement.

Formation control itself is not a topic that emerges from the field of multi-robot systems. In fact, it is observed in nature, such as bird flocking, animal herding, fish schooling. Formation behaviors benefit these animals by giving them higher chance of survival or increasing their hunting performance. For example, a hawk will have significantly less chance of success when attacking a flock of pigeons comparing to a single pigeon. Another example of formation in animals is found in wild geese [9]. When the winter comes, wild geese have to migrate in a long distance; they fly in a formation of V shape. Studies have shown that this shape can reduce wind resistance to each individual goose and increase the range for migration at least 71 percents [9]. Study of herding also shows that animals can combine sensing ability to increase chances of detecting predators [3]. Thus, It is believed that groups of robots could benefit from formation behaviors as well.

Multi-Robot Formation can find its application in various areas, such as search and rescue, land mine removal, mapping and exploration and surveillance. The use of Multi-Robot formation tends to grow rapidly because the recent technological development that makes it feasible to deploy a large group of simple mobile robots, hundreds or even thousands. Some applications that require substantial number of robots, such as mobile sensor network, can be realized now.

In this paper, we propose a fomation control algorithm for maintaining and navigating a team formation in an obstacle-filled worksapce. We also introduce an human-robot interfce via a data glove which allows a human operator to control a group of robot using only his gesture.

Controlling a group of robot using only one human operator is a challenging problem. Because, a human has limited perception and action capabilities. Hence, a human operator capabilities are very limited resources and should not be wasted on controlling an individual robot in a team. The control commands are often complex and not suitable to be used with traditional keyboard-mouse system. The human-robot interface that can map the natural action of a human operator onto group level commands for a robot team is highly desirable.

The remainder of the paper is organized as follow. In Sec. II, we review prior works in formation control. In Sec. III, we introduce system architecture used in our experiments. In Sec. V, we present our formation control method. In Sec. V, we present experiment results along with discussion about the results. Finally, we conclude our paper in Sec VI.

II. Related Works

There have been increasing research activities in the field of Multi-Robot formation control during the past decade. Many works have extensively investigated many problems in this field, including coverage control, formation coordination and communication issues. Broader overview of multi-robot formation can be found in literature reviews of Arai [1], Cao [7] and Parker [14]. A more specific and deeper review on Multi-Robot formation can be found in [2]. In this section, we review some priori works related to our study.

The first study of formation control did not come from robotics community. In fact, a computer graphic simulation of bird flock [15], [16] is recognized as the first attempt of formation control in a group of artificial agents. For a formation control in robotics, a behavior based control scheme, such as subsumption and motor schema is widely used. Parker [13] introduced a line-formation control method; a proper balance between global and local control was discussed. the simulated agents are programmed with subsumption architecture. The robots in the teams use a leader following approach to keep formation. However, this study presented only line formation and did not propose extension for other type of formation. Later, Balch and Arkin [3] proposed a behavior-based control algorithm for a robot team. The method generates four distinct shapes, line, column, diamond and wedge, by executing layers of motor schemas, each of which has its own important gain and parameter values. The important gain can be adjusted to change the weight of its corresponding behavior in the final vector summation (the process for generating control input to the robot). A motor schema control architecture is selected because it provides more flexibility than a subsumption architecture.

Sugihara and Suzuki[18] proposed distributed motion coordination algorithms for forming various geometric shapes including circle, line and rectangle. Each of the shapes requires its own formation control algorithm. For example, to generate a filled circular shape, each robot must keep the distance between itself and the furthest robot close to the diameter of the desired circle. The result shows successful formation in a large group of simulated agents. However, knowing the distances to every robot in the workspace is rarely feasible due to sensory and communication limitation.

In [8], a graph theoretic approach for modeling mobile robot formation was presented. The framework allows transition between different formations and avoiding obstacle. In this formation technique, every team of robots will have a leader robot that directly or indirectly controls other follower robots. Directed graph is used to represent leader-follower relationship between each pair of robots. This work contributes in mathematical representation of formation by graph theoretic approach. However, formation control mechanic is not discussed.

A covalent bond inspired method was proposed by Balch and Hybinette[4], [5]. In this study, *attachment sites* are assigned to every robot. There are four basic geometric patterns of attachment(+, – , X and I) where each robot is located at the center of the shapes and attachment sites are at the end of lines. Each robot will attach its attachment sites to other robots' attachment sites in order to form a complex geometric structure mimicking crystal forming in nature. This method requires neither global communication nor broadcasting. Hence, the method is suitable for large scale pattern formation where global communication is not feasible. However, due to fixed attachment sites, the robot teams cannot change formation structures while operating. And, the final formations are restricted by the shape of attachment sites.

The area of human-robot interaction has been studied extensively during recent years. But, only limited works focus on Multi-Robot systems. Nielsen [12] propose coordination methods between a human operator and a robot team, namely teleoperation, point to point and region of interest. In the point to point method, the human operator issues commands for the robot teams to move from one landmark to another. In the region of interest method, The operator issues commands for a robot to move to a specific region of the workspace and the robot will move to that region autonomously. However, all of these methods require a human operator to individually issue command to a single robot not a group which decreases the system scalability for a large robot team.

The method presented in this paper differs form above mentioned works in a number of ways. Firstly, we address the problem of human operators controlling a group of mobile robots whereas others' works discuss only moving formation along pre-defined paths. Secondly, we propose a formation control algorithm that facilitates real time adjustment of formation properties. Finally, we impose many constraints that imitate real world problems such as communication limitation and sensing ranges.

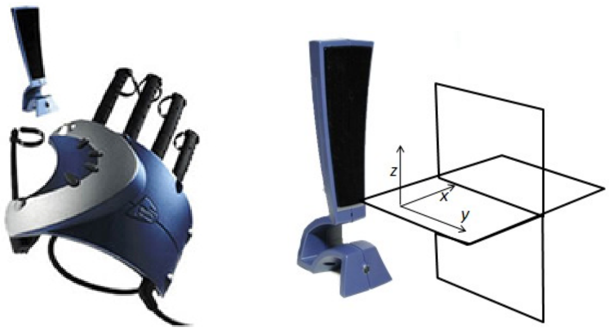
III. Architectures and Frameworks

A. System Architecture

Our system can be divided into two parts. One is a human operator interface side; it composes of data glove to capture a human operator's hand gesture. The data from this side is transmitted on standard TCP/IP network which allows us to set up the human operator side as a teleoperation system. On the other end, it is robot side. This system can be either simulation platform or real robot teams control system.

B. Data Glove Model

In this study, we use a P5 data glove from Essential Reality, shown in Fig. 1, as our input device. It can measure hand relative position in three dimension and bending of each finger. The glove composes of two separated units, a glove unit and a base unit. A glove unit has five bend sensors, one for each finger, and has eight infrared LED embedded throughout the glove body. The positions of these LED can be read individually which allows us to determine a palm posture. The base unit acts as our reference point and will not be relocated during our experiments.



(a) (b)

Fig. 1. P5 Data Glove System (a) a glove and a base unit (b) coordinate system of a data glove

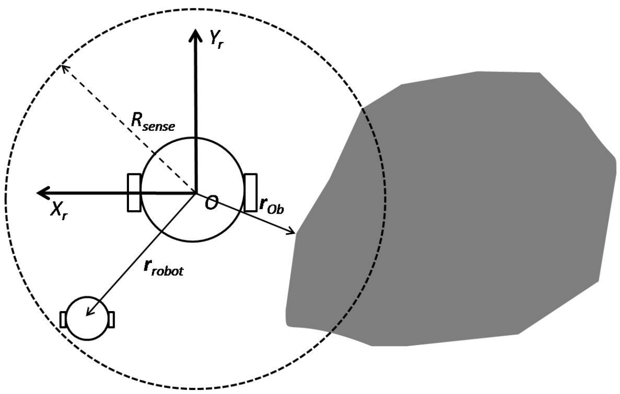


Fig. 2. Robot Model

C. Robot Model

We consider a group of N holonomic robots. These robots do not have ability to localize, i.e., cannot determine their absolute position in the plane; this means that they cannot be directly commanded to move to a given position. Each of these robots is equipped with a sensor capable of measuring the relative position of the other robots or objects with respect to itself . As shown in Fig. 2, R_{sense} represents robot sensing radious. $\boldsymbol{r}_{obs}$ represents positional vector from the robot to the closet point of the sensed obstacle. And, $\boldsymbol{r}_{robot}$ represents positional vector from the robot to other team member in its range; the total number of $\boldsymbol{r}_{robot}$ is equal to number of neighboring robot being sensed. A human operator can provides intructions via broadcast communication. But, there is no communication between team members.

IV. Methodology

A. Control Algorithm Overview

The most important issue in formation control is generating and maintaining desired formation. In this study, we propose an approached based on a spring kinetic force to generate and maintain formation. Roughly speaking, every robot has its own reference robots to refer its position in a formation. It can have more than one reference robot to increase robustness. The reference robot will push it back if it move too close and pull them toward if it move too far from its correct position. By calculating this simple force law, a robust formation can be achieved.

Another issue is to navigate the team safely to the destination. We implement the concept of potential field. Every robot is treated like an electrical charged particle. It repels against obstacles and attracted by the goal. By summing all of the previous mentioned force. we can maintain and navigate a formation safely in obstacle filled workspace.

The following pseudocode describes the control algorithm executed on each robot at some constant time interval. The objective of the algorithm is to compute a new velocity for the robot. In the pseudocode, this velocity is denoted by vector $\boldsymbol{v}$. We also denote by vector $\boldsymbol{r}_i$ the position of robot i (of course, the position is in the frame of reference of the robot that is executing the control algorithm). We assume that the robot have sufficient acceleration to reach desired velocity instantaneously. Hence, we simplify our calculation to use the resulting force to vary velocity directly.

Algorithm 1 Formation Control

for each robot i being sensed **do**
  $\boldsymbol{v}$ += calculateFormationForce($thisRobot$,i)
end for
$\boldsymbol{v}$ += calculateObstacleForce($thisRobot$)
$\boldsymbol{v}$ += calculateGoalForce($thisRobot$,i)
if $|\boldsymbol{v}| > V_{max}$ **then**
  $\boldsymbol{v} = V_{max} * \frac{\boldsymbol{v}}{|\boldsymbol{v}|}$
end if
if $|\boldsymbol{v}| < V_{min}$ **then**
  $\boldsymbol{v} = 0$
end if
execute velocity command $\boldsymbol{v}$

As seen in the psuedocode, the robot calculates velocity from three different force sources, formation forces, obstacle forces and a goal force. The details regarding calculation of each force are explained later in this section. The robot move along the velocity vector under maximum V_{max} velocity. The algorithm stops the robot when its speed becomes lower than the threshold V_{min}.

B. Formation Force

The formation force is a force that hold robots together as a group. This force based on a concept of spring kinematic force. Basically, every robot is placed in its position by virtual spring. At the equilibrium state, the spring will place the robot at the desired position. If the robot move away from this position, it will be *pull back* by the virtual sping. On the other hand, if the robot move too close to its reference, it will be *push away* by the virtual sping. This spring based idea is illustrated in Fig 3.

The reference neighbor appear to the robot as a static reference position. For every virtual spring, there are three pa-

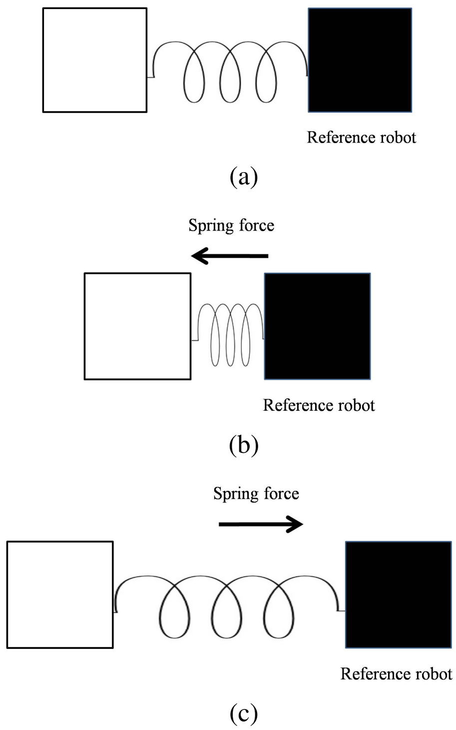


Fig. 3. The spring force approach (a) at equilibrium state (b) at compressed state (c) at stretched state

TABLE I
A FORMATION DATA TABLE

Reference	length	angle	$\mathbf{k_{spring}}$
Ref_1	L_1	Θ_1	k_1
Ref_2	L_2	Θ_2	k_2
...	...	...	...
...	...	...	...

rameters to determine formation properties namely alignment, length and spring constant. Table I shows the formation data in each robot The angular position between a robot and its neighbor is governed by the alignment parameter. The length parameter controls a distance between a robot and its reference. Lastly, the spring constant is used for controlling formation behaviors. there can be one or any numbers of reference neighbors for a robot in the formation. Hence, if there are errors in sensing some neighboring robots, perceived information of other neighboring robots can still be used to generate and maintain the formation. This increases a robustness of our approach against noisy sensor information.

The force of our virtual spring is inspired by Hook's law. Let $\boldsymbol{F}$ denotes the spring force vector. $\boldsymbol{x}$ is positional vector from a robot to its virtual spring's equilibrium position. And, k_{spring} denotes a spring constant. The hook's law equation is as follow.

$$\boldsymbol{F} = -k_{sping}\boldsymbol{x} \tag{1}$$

As previously discussed, there can be more than one reference robot for each team member. Hence, the resulting formation force is a summation of all virtual spring attached to that robot. Let $\boldsymbol{F}_{formation}$ denotes the resulting spring force vector. And, the robot senses total n reference neighbors. The resulting formatoin force can be written as follow.

$$\boldsymbol{F}_{formation} = \sum_{i=1}^{n} \boldsymbol{F}_i \tag{2}$$

C. Obstacle & Goal Force

Besides of keeping team members in the desired formation, the robot team need to navigate itself to its goal without collision with obstacles along its path. In our study, we implement a potential field based apporach [10], which is widely used in mobile robotics, as a navigation algorithm. The fields are design in the way that each robot is repelled by obstacles and attracted by a goal.

We can describe the potential field caused by obstacles, represented as U_o, as follow

$$U_o = \begin{cases} k_o \sum_i \frac{1}{|\boldsymbol{r}_i|} & \text{if } |\boldsymbol{r}_i| \leq R_{sense} \\ 0 & \text{if } |\boldsymbol{r}_i| > R_{sense} \end{cases} \tag{3}$$

The summation is done over all obstacle perceived by the robot. k_o denotes a constant value governing the strength of the obstacle field. Let $\boldsymbol{x}$ be the position of the robot and $\boldsymbol{x}_i$ be the position of the closet position of obstacle i. Then, r_i is described as $r_i = |\boldsymbol{x}_i - \boldsymbol{x}|$ Having the obstacle potential field defined, we can derived the force caused by this repulsive field on the robot as follow.

$$\boldsymbol{F}_o = -\frac{dU_o}{d\boldsymbol{x}} = -\sum_i \frac{dU_o}{dr_i}\cdot\frac{dr_i}{dx} \tag{4}$$

$$\boldsymbol{F}_o = \begin{cases} -k_o \sum_i \frac{1}{|\boldsymbol{r}_i|^2}\cdot\frac{\boldsymbol{r}_i}{|\boldsymbol{r}_i|} & \text{if } |\boldsymbol{r}_i| \leq R_{sense} \\ 0 & \text{if } |\boldsymbol{r}_i| > R_{sense} \end{cases} \tag{5}$$

We can see from Equation 5 that the repulsive force from obstacles tends to increase rapidly in magnitude when the robot move toward the obstacle. This prevent the robot from colliding with obstacles.

For the attractive potential field caused by the robot goal, U_{goal}, we can descibe it as follow.

$$U_{goal} = -\frac{1}{2}k_{goal}|\boldsymbol{r}_g|^2 \tag{6}$$

k_{goal} is a constant describing the strength of the attractive potential field caused by a goal. Let $\boldsymbol{x}$ be the position of the robot and $\boldsymbol{x}_g$ be the posotion of the goal. Then, r_g is descibed as $r_g = |\boldsymbol{x}_g - \boldsymbol{x}|$. we can apply a similar derivative as used in Equation 4 to obtain:

$$\boldsymbol{F}_{goal} = k_{goal}\boldsymbol{r}_g \tag{7}$$

Upon this point, we have got all forces that apply on each robot, namely the formation force, the obstacle force and the goal force.

D. Human-Robot Interface

Controlling a group is a challenging task for a human operator because enormous actions and commands are needed in order to control the group. It is desirable to have an intuitive control interface to control robot team as a whole not as individual robots. In our study, a data glove is selected because its remarkable property that can intuitively map human natural gestures onto robot control commands.

We allow two major properties, a formation distance factor and a goal, of robot formation to be adjusted online during operations. The distance of every virtual spring can be adjusted by imposing a formation distance factor issued via the data glove. And, the goal position is issued to every robot in the team using the data glove.

As previously presented in Subsection IV-B, the formation distance between any pair of robots is governed by the length of the virtual spring. Here, we utilize the tightness of a human operator fist measured by the data glove's bending sensors to adjust the effective virtual spring length. The data glove provides eight bit value from five bending sensors; one for each finger. Let T be the tightness value, an average bending value from every fingers. L_d , c_1 and c_2 represent a default virtual spring length , an arbitrary constant and a closet distance for any pair of robot respectively. We can write down the effective virtual spring length, L_{eff}, as follow:

$$L_{eff} = \frac{c_1 L_d}{T} + c_2 \qquad (8)$$

Regarding a goal position, the glove position in two dimensional plane , $x - y$, is used for controlling the goal position. The glove position value is adjusted to start from $(0, 0)$ and scaled to match workspace size.

V. Simulation Experiments

We have conducted our experiment in our multi-robot simulator. The simulator calculates the new control command for each robot using the control algorithm introduced in Sec. IV. The command composes of a velocity and a direction for the robot. After that, the position of each robot is updated using the calculated control command. In the next iteration, the control command calculation will be based on these updated positions.

We have defined many parameters that might affect our coverage control process. In all experiments, amount of time will be measured in iterations. The maximum speed is set to 1 unit per iteration. The workspace is set to be a square with the dimension of 400 by 400 units.

A. Basic Formation Results

In this first set of experiments, we want to investigate the basic behavior of our formation control method. We perform several runs of this experiment with different initial positions. The k_{spring} is set at 0.05. The target formation is shown in Fig 4 The snapshots in Fig. 5 show result of our algorithm from one experiment run.

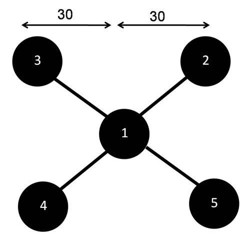


Fig. 4. The target formation

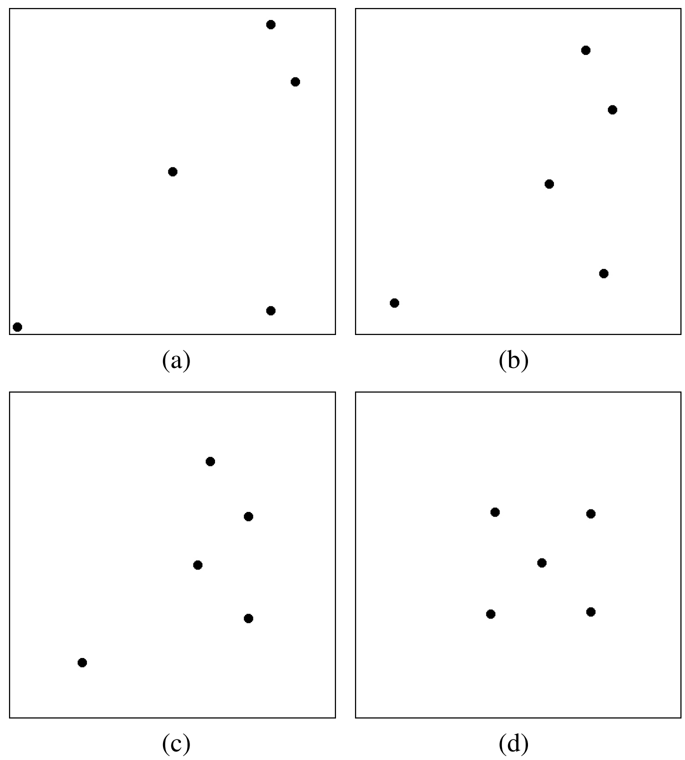


Fig. 5. Snapshots of simulation results at iterations (a) 0, (b) 50, (c) 100 and (d) 200

The experiment start from initial position shown in Fig. 5 (a). After that, the robots rapidly move toward their corresponding formation position. The robots seem to move closer to the desired formation in Fig 5(c),(b). As the time pass (Fig 5(d)), the resulting coverage seem to be more stable. And, the robots in each region appear to be uniformly distributed.

B. Formation Accuracy

In this experiment, we want to investigate the error in formation at each time step. The formation error defines by the summation of a deviation distance of each robot to tis correct formation position. The setup of this experiment is the same as that of the previous one. The plot in Fig. 6 shows the formation error of the resulting formation control algorithm at each time step. As shown in this plot, the formation error decrease monotonically and it is the same for every V_{max} value in our experiment.

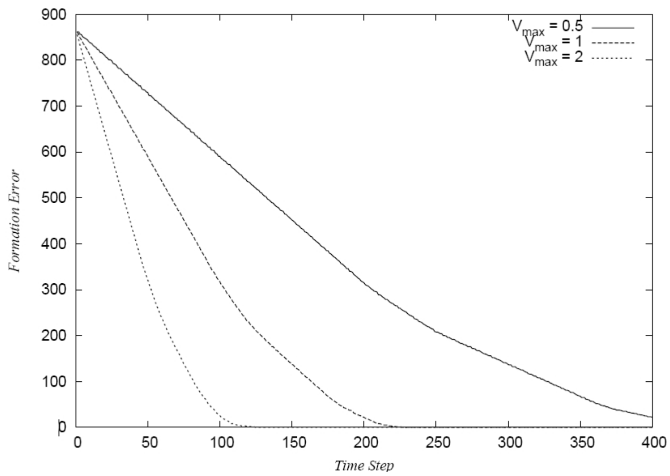
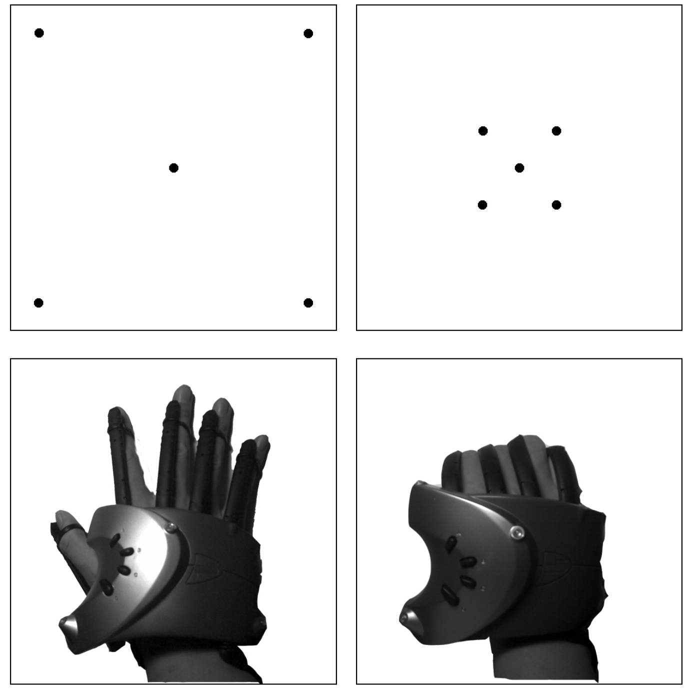


Fig. 6. The formation error as a function of iterations



(a) (b)

Fig. 7. Resulting formation at (a)an open palm (b)a clenched fist

C. Formation Properties Control

We have introduced the human-robot interface for our multi-robot formation. Here, we demonstrate the result of changing a human hand gesture that result on the robot team formation. In this experiment, we focus on adjusting the length of the virtual spring. Thus, we omit goal force and make robot No. 1 stay stationary as a reference. We let a human operator wear a P5 data glove and change the gesture of his hand from an open palm to a clenched fist. c_1 and c_2 are set at 40 and 0 respectively. Other settings are similar to those of the previous experiments. The result are shown in Fig. 7.

We can see from Fig 7 that the formation of the group change according to the human operator gesture. The explanation for this result is simple. At an open palm state, the data glove bending sensors are at full stretch position which result in lowering T in Equation 8. This increases the effective virtual spring length. On the other hand, clenching a fist yields the opposite result.

D. Moving Formation

This experiment's objective is to investigate the result of moving the formation in an obstacle filled workspace. We experiment with a team of robot already in a formation and moves toward a goal located at $(300, 300)$ on the top right corner of the workspace. In this experiment, k_o is set at 100 and k_{goal} is set at 0.01. And, the size of target formation is a half of that of the prevous experiment. In Fig. 8 , we show the snapshots from this experiment.

In Fig 8 (a), the robot team start with a complete formation and are ordered to move toward the goal. In Fig 8 (b), the robots on the right size encounter an obstacle. the formation shape is changed to avoid the collision. but, still maintain some level of formation. And, the same behavior appears in the robot on the left side as seen in Fig 8 (c). Finally, the robot team reach its destination with the desired formation.

This result shows the effectiveness of our algorithm in navigating the robot team in a complex environment. The important of formation maintenance and obstacle avoidance can be adjusted via force constants.

VI. Conclusions

We have presented a scalable and fully distributed method for the problem of formation control for a group of mobile robots. The method requires no priori information of a workspace nor communication. It has been shown in simulation experiment results that the method is effective at generating and maintaining a formation. We also introduce a simple and inntuitive way to control a group of robots using only one human operator. This human-robot interfce mechanism can reduce workload of a human operator and allows the operator to focus more on task oriented function. It is interesting to investigate how can we extend our method into higher dimension which will be useful for controlling UAVs or underwater vehicles. Beside this issue, we would like to analyze the stability of the formation in more detail.

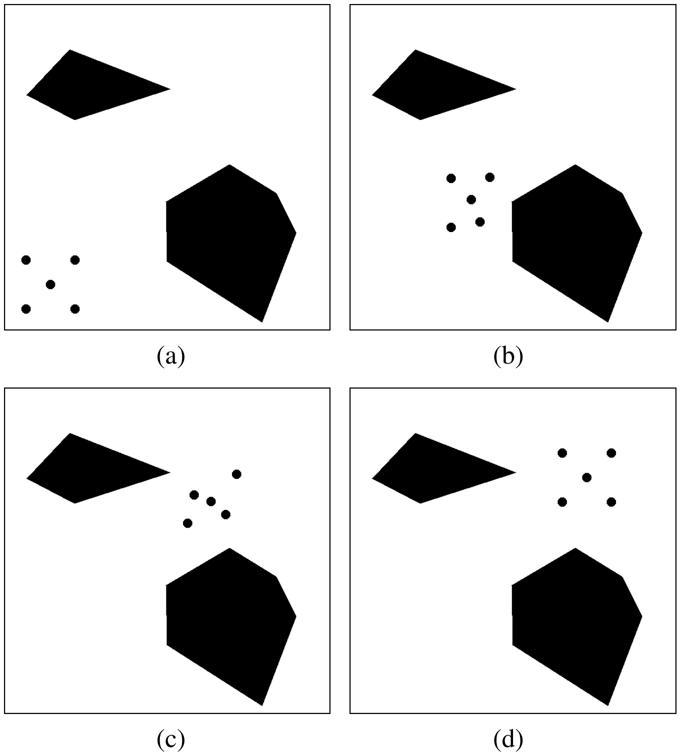


Fig. 8. Snapshots of moving formation at iterations (a) 0, (b) 150, (c) 350 and (d) 500

REFERENCES

[1] Tamio Arai, Enrico Pagello, and Lynne E. Parker. Editorial: Advances in Multi-Robot Systems. *IEEE Transactions on robotics and automation*, 18(5):655–661, october 2002.

[2] E. Bahceci, O. Soysal, and E. Sahin. A Review: Pattern Formation and Adaptation in Multi-Robot Systems. Technical report, Robotics Institute, Carnegie Mellon University, October 2003.

[3] T. Balch and R. Arkin. Behavior-based Formation Control for Multi-robot Teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[4] Tucker Balch and M. Hybinette. Behavior-based coordination of large-scale robot formations. In *Proceedings of the Fourth International Conference on Multiagent Systems (ICMAS '00)*, pages 363 – 364, July 2000.

[5] Tucker Balch and Maria Hybinette. Social Potentials for Scalable Multirobot Formations. In *IEEE International Conference on Robotics and Automation (ICRA-2000)*, San Fransico, 2000.

[6] Wolfram Burgard, Mark Moors, Dieter Fox, Reid Simmons, and Sebastian Thrun. Collaborative Multi-Robot Exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[7] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng. Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, 4(1):7–23, March 1997.

[8] Jaydev P. Desai. A graph theoretic approach for modeling mobile robot team formations. *Journal of Robotic Systems*, 19(11):511–525, 2002.

[9] Hainsworth and F. Reed. Wing movement and positioning for aerodynamic benefit by Canada geese flying formation. *Canadian journal of zoology, v. 67, March 1989, p. 585-589.*, 67:585–589, 1989.

[10] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5, 1986.

[11] Jiming Liu and Jianbing Wu. *Multi-Agent Robotic Systems*. The CRC Press, 2001.

[12] C. Nielsen, M. Goodrich, and J. Crandall. Experiments in human-robot teams. In *the 2002 NRL Workshop on Multi-Robot Systems*, 2003.

[13] L. E. Parker. Designing Control Laws for Cooperative Agent Teams. In *IEEE International Conference on Robotics and Automation*, 1993.

[14] L. E. Parker. Current State of the Art in Distributed Autonomous Mobile Robotics. In L. E. Parker, G.Bekey, and J. Barhen, editors, *Distributed Autonomous Robotic Systems 4*, pages 3–12, Tokyo, 2000. Springer-Verlag.

[15] Craig Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics. In *SIGGRAPH '87*, pages 25–34, 1987.

[16] Craig W. Reynolds. Flocks, Herds, and Schools: A Distributed Behavioral Model. *Computer Graphics*, 21(4):25–34, 1987.

[17] P. Santana, J. Barata, H. Cruz, A. Mestre, J. Lisboa, and L. Flores. A multi-robot system for landmine detection. In *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*, 2005.

[18] K. Sugihara and I. Suzuki. Distributed Algorithms for Formation of Geometric Patterns with Many Mobile Robots. *Journal of Robotic Systems*, 13(3):127–139, 1996.