UNSYS digital

Journal of Unmanned System Technology

Simulation of Swarm Robot Flocking Assisted by Explicit Communication

Bayu Erfianto and Bambang Riyanto Trilaksono
School of Electrical Engingeering and Informatics, Bandung Institute of Technology
Bandung, Indonesia

***Abstract*—**Flock-like motion has been intensively studied in swarm robotics. The aim of flocking is to make a formation of mobile robots involved, which is to mimic crowd and movement of animal behavior. In this paper, we simulated a flocking of mobile robots, which was guided by explicit communication via broadcast mechanism in wireless network. Effects of explicit communication in flocking behavior were analyzed. The first is the ratio of broadcasted message to the flocking action. Second is to observe the connectedness of communication topology affected by flocking formation.

***Keywords*—** flocking, swarm robotics, explicit communication.

I. INTRODUCTION

The research of flocking is categorized under the area of swarm intelligence in which the domain of artificial intelligent is used to study swarm behavior in autonomous multi-agent robot. In swarm intelligence, flocking is required to synchronize the movement of agent in order to achieve collective goal in activities such as foraging, consuming, or grazing [1]. The synchronization of movement can be carried out either via local communication or global communication, with or without communication [2].

Penders in [2] identifies non-communication behavior of swarm robot for task distribution e.g. for obstacle avoidance, wall and track following, aggregation or clustering, area coverage, exploration behavior, and autonomous navigation. However, communication is required by swarm in order to control the behavior such as in flocking robots. In swarm robotics, an autonomous robot interacts with the other robots, as well as with the environment. This mechanism results a collective behavior-base robotics.

Bird flocking, animal herding and movements have been studied by Reynolds [3] via computer graphics modeling. Over the years, many researchers have been developing flocking model and algorithm as the improvement of Reynold's model. In this paper, we use the flocking models that are developed by some notable researchers such as Reynolds [1], flocking of multi-agent boids by Wilensky [4] and robots exploration as function of flocking by Bouroqadi [7]. The readers can also refer to other works, such as informed robots in flocking behavior by Turgut [8] and Celikkanat [9], boid-like flocking in sensor network by Chibaya [10], and minimalist flocking algorithm by Moeslinger [11]. In this paper, we extended the work of Turgut [8] by analyzing the effect of explicit communication in swarm robot flocking. We broadcasted flocking message that contains robot information state over the wireless network. The first result, we observed the efficiency of periodic broadcast message that affected the flocking actions. As the second result, we observed the connectedness of swarm robotics affected by flocking formation. It can be shown that flocking behavior can be used to maintain communication among robotics. And by maintaining communication topology, each robot can broadcast its information state to its neighbors in order to make flocking formation.

This paper extended the work of [16] by adding the analysis of connectedness of the use of explicit communication in flocking algorithm. The organization of this paper is as follows. In section II we briefly revisit the existing flocking model that is extensively used in swarm robotics behavior. In section III, we present the algorithm of communication assisted flocking behavior. In section IV, the results of simulation and the analysis are presented. Conclusion in section V ends the discussion in this paper.

II. FLOCKING MODEL IN SWARM ROBOTICS

In 1987, Reynolds developed a simulation of bird flock in computer graphic manner to mimic the bird flocking behavior [3]. He defined boids, which is to name the generic simulated flocking birds. In his simulation, boids are allowed to fly using three simple rules i.e. separation, alignment, and cohesion. Separation behavior is used by a boid to avoid crowding of local boids (flockmates). Alignment behavior is intended to steer boids toward the average flockmates heading. And after alignment, cohesion algorithm is used to move boids toward the average position of flockmates. The illustration of Reynolds' flocking behavior is depicted in **Figure 1**.

Uri Wilensky further developed a simulation of flocking bird inspired by Reynolds' boids algorithm [4]. The simulation developed by Wilensky is very similar to Reynolds' flocking rules; however it is built in 2D environment under Netlogo, a multi-agent simulator. For our work in this paper, we used flocking model library presented in Netlogo to simulate flocking of swarm robots. The algorithm of Reynold's flocking

Corresponding author: Bambang Riyanto (e-mail: briyanto@lskk.ee.itb.ac.id)

This paper was submitted on July 17, 2013; revised on August 18, 2013; and accepted on August 22, 2013.

rules is illustrated in **Figure 2** and the parameters used in Wilensky's Netlogo model are given in **Table 1**.

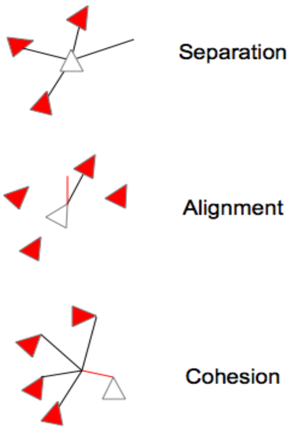

Figure 1 Original Reynold's flocking rules

Table 1 Parameters used in Wilensky's model

Parameter	Meanings
Population	The number of birds. In this paper, population is the number of mobile robots used during simulation.
Vision	The distance of vision a bird can see. In this paper, a vision is replaced with sonar sensor a robot can sense the neighboring robots and environment with.
Minimum Separation Distance	the minimum separation distance a robot maintains from the neighboring robots
Max-align-turn	max-align-turn is the maximum angle of turn an agent can align with local flockmates
Max-cohere-turn	the maximum angle an agent can turn toward local flockmates during cohesion
Max-separate turn	the maximum angle through which an agent can turn away to avoid collision with local flockmates during separation

Couzin et al. in [5] proposed a flocking behavior model in which each agent defines three zones of surrounding, i.e. zone of repulsion (*zor*), zone of orientation (*zoo*) and zone of attraction (*zoa*). *zor* is closest to the center of agent, while *zoa* is the outer zone, and *zoo* is in between. In this paper, we adapted the terminology of Couzin's zones of surrounding, however only *zoa* and *zor* are used.

Nembrini proposed the minimalist algorithm to maintain network of robots. In his work, there were two types of sensors involved. First was infrared sensor which was used as proximity sensors. This infrared sensor was able to capture surrounding information to avoid collision among robot agents and obstacles. The second sensor used line of sight beacon sensor formed by special function mobile robots.

Nembrini introduced two agents as the main contribution of his research, i.e. α-agent and β-agent. The term of α-agent refers to agent whose interaction with other agents is such that a flock formation is maintained; β-agent refers to agent that maintains their flock formation by interacting with beacons. Due to obstacle's presence, line of sight can be obstructed, and as in Nembrini's work, the occluded robots could not see the light transmitted by a beacon since it is blocked by obstacle. Another robot received light from that beacon and relayed it to the others via networked of robots. The illustration of Nembrini's work is depicted in **Figure 2**.

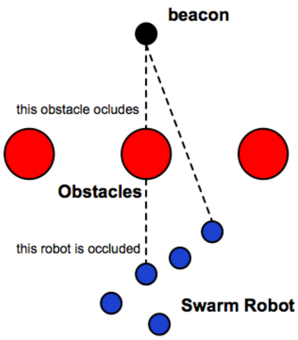

Figure 2 Nembrini's swarm environment in [13]

Olfati-Saber's algorithm [6] extended the work of Nembrini by incorporating 3 agents: α-, β-, and γ-agent. Like in Nembrini's, all single boid in a flock were α-agent or physical agent. The other two agents, beta and gamma, were defined as virtual agents, where β-agent represented the obstacles and γ-agent in Olfati-Saber's model was the predefined group objective. Olfati-Saber defined flocking algorithm as $u_i^\alpha + u_i^\beta + u_i^\gamma$, where the u_i^α is the consensus used in flocking behavior that is composed of (α,α) interaction, u_i^β denotes the (α,β) interaction, and u_i^γ is group objective which represents the distributed navigational feedback as the function of predefined group objective. The term of u_i^α is to maintain the agents always in a flock formation via consensus, while u_i^β interaction term is used for obstacle avoidance. In u_i^β, a virtual β-agent represents the closest point of the obstacle from an α-agent. The third agent, γ-agent, is a virtual leader, which is used to lead the flock to the desired location of group objective.

Bouraqadi et al. in their paper [7] maked use of multi-robots system in exploration of an unknown environment. They defined 5 steering rules of multi-robots flocking i.e. collision avoidance, separation, alignment, cohesion, and exploration. Bouraqadi et al. proposed a solution based on flocking rules in order to explore an open area while keeping robots close enough to avoid disconnections. They used similar flocking mechanism such as that based on Wilensky's flocking. However, in exploration steering, the communication mechanism was used to exchange information about the area being explored by each robot.

Chibaya and Bangay [10] used boid-like sensor network to extend the coverage area of sensor network. They used Reynold's flocking rules, and upon deployment, agents were set to be in separation mode. In their work, the cohesion rule then guaranteed all boid-like sensors network to be within the swarm and to cover area with its explored neighboring spaces.

Turgut et al. in their paper [8] described a virtual heading sensor, which broadcasted digital compass readings through wireless communication channel, and obtained the relative headings in a group of robots. Their method was based only on assumption that the sensed north direction remains the same over the neighboring robots. The performances they measured were: (i) entropy, which is to measure the positional disorder of the group, (ii) average velocity of the center of the flock, (iii) successful rate, which is to measure the run of the robot ("stuck/collided" to "not stuck" ratio).

III. FLOCKING ALGORITHM USING EXPLICIT COMMUNICATION

In this paper, we extended a flocking algorithm of Wilensky [4]. In the original algorithm, there was no explicit communication used. In other words, there was no information state exchanged among swarm robots, and the flocking algorithm relied only on local sensing on each robot. We further propose a guided flocking by making use of explicit communication in which wireless communication was used to carry robot information state that contains robot identity and current robot heading. The notations and parameters used in proposed algorithm are denoted in **Tabel 2**.

Table 2 Parameters used in our model

Parameter	Meanings
n_r	the number of robots used in simulation
d_{zor}	separation distance in zone of repulsion
d_{zoa}	distance of zone of attraction as the wireless network coverage
θ_s	angle of separation turn
θ_a	angle of align turn
θ_c	angle of cohesion turn
θ_{col}	angle of collision turn
δ_{sensor}	distance of object from sensor
$robot_i(heading_{now})$	current robot i heading
$robot_i(heading_{new})$	new robot i heading
$robot_j$	robot i flockmates
$heading_{avg}$	average heading towards alignment
t	period of broadcast

To explain our model, we present a flowchart of flocking algorithm using explicit communication as depicted in **Figure 3**. Suppose there was a $robot_j$, which was in communication range of $robot_i$. These robots would initiate a communication topology via wireless network. Once communication topology was initiated, each robot could further broadcast robot identity and current robot heading to initiate flocking. Neighboring robots that received broadcast message will initiate flocking by executing three rules: separation, alignment and cohesion.

Detailed explanation of flowchart in **Figure 3** is expressed in Algorithm 1. To perform simulation, we required input parameters such as number of robots involved in flocking, sonar sensor range, wireless signal range, angle of robot, when to perform flocking, and the velocity of robot. Initially, each robot executed collision avoidance to avoid obstacles including the nearest robotj that could possibly collide with robot.

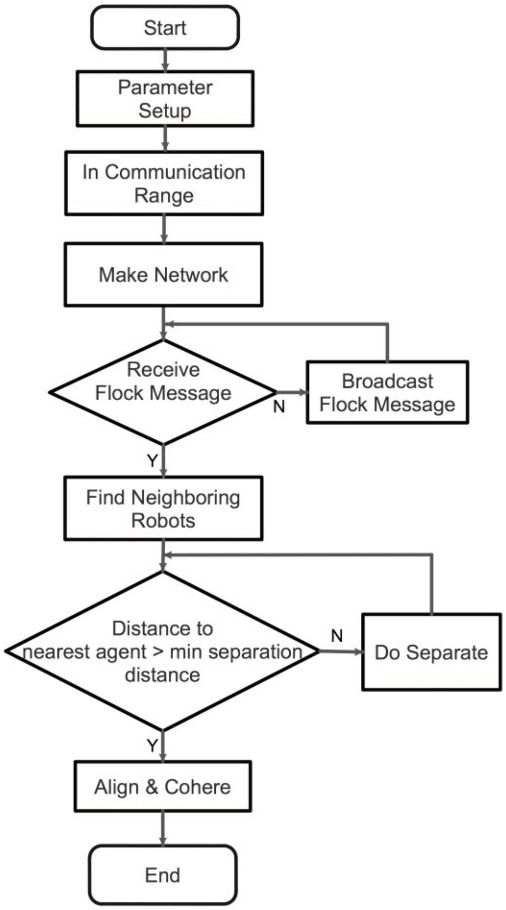

Figure 3 Proposed flocking algorithm using explicit communication

Algorithm 1 Flocking using Explicit Communication

```
Input: $n_r, d_{zor}, d_{zoa}, \theta_s, \theta_c, \theta_a, v$
Output: $h_{robot}$
while $robot_i(v)$ do
    Do Collision-Avoidance
    while $robot_i(d_{zoa}) \leftarrow robot_j$ do
        Do Make-Network
    end while
end while
```

Suppose the $robot_j$ was entering the wireless communication range, i.e. zone of attraction (zoa) of $robot_i$. $Robot_i$ would detect the presence of other robots and immediately executed algorithm Make-Network to make wireless network topology, as presented in line 4 in Algorithm 1. The detail of Make-Network algorithm is presented in Algorithm 2. Once the communication graph was created, the robots that were in wireless communication coverage (zoa) either could broadcast information state or receive information state from others. For instance, robot$_i$ broadcasted a flocking message within a period of t time unit (line 2 Algorithm 2). Suppose robot$_i$ received flocking message from robot$_j$, it then reacted by executing flocking algorithm (line 5 Algorithm 2).

Algorithm 2 Make-Network

```
1: for every $(t)$ do
2:     $robot_i(broadcast(heading_{now}))$
3: end for
4: if $robot_i(receive(robot_j(heading_{now})))$ then
5:     Do Flock
6: end if
```

The Flock algorithm, as presented in Algorithm 3, would work as follows. In line 1 of Algorithm 3, if robot$_j$ was reaching zone of repulsion of robot$_i$, a proximity sensor of robot$_i$ would detect the presence of robot$_j$ within the zone of repulsion of robot$_i$. The value of robot$_i$(δ_{sensor}) contains the distance of robot$_j$, and if the value of robot$_i$(δ_{sensor}) $\leq$ robot$_i$(δ_{zor}), meaning that the distance of robot$_j$ was less than the threshold defined from the zone of repulsion of robot$_i$, separation algorithm had to be executed, otherwise robot$_j$ would collide with robot$_i$. If robot$_j$ was not in the zone of repulsion of robot$_i$, alignment and cohesion algorithm would be executed, as presented in line 4 and 5 of Algorithm 3. Details of alignment algorithm are presented in Algorithm 4, and of cohesion algorithm in Algorithm 5.

Algorithm 3 Flock

```
1: if $robot_i(\delta_{sensor}) \leq robot_i(d_{zor})$ then
2:     Do Separate
3: else
4:     ALIGN
5:     COHERE
6: end if
```

Algorithm 4 SEPARATE

```
$x = |robot_i(heading_{now}) - robot_j(heading_{now})|$
if $x > 0$ then
    $robot_i(\text{turn\_left}(\theta_s))$ OR $robot_i(\text{turn\_right}(\theta_s))$
end if
```

Algorithm 5 ALIGN

```
$x = \sum_n sine\_of(robot_i(heading_{now}))$
$x = \sum_n cosine\_of(robot_i(heading_{now}))$
if $x > 0$ and $y = 0$ then
    return $robot_i(heading_{new}) = robot_i(heading_{now})$
else
    return $(heading_{avg})$
    return $(heading_{avg})$ = convert(x,y)
    $robot_i(heading_{avg}, \theta_a)$
end if
```

IV. SIMULATION AND ANALYSIS

A. *Simulation Environment*

In this paper, simulation of flocking algorithm by explicit communication was performed in Netlogo, a multi-purpose multi-agent simulator. Since Netlogo does not provide a precise and specific model of mobile robots, we created a customed turtle definition in Netlogo as a mobile robot. In Netlogo, there were actually four basic agents: turtle, link, patch and observer. Turtle was a representation of agent that moves around the environment. Link was used to model the communication network among agents. Patch was the environment or the arena of simulation. The environment itself was two-dimensional space composed of grid of patches. Each patch was a square piece of ground over which a turtle could exist, move, and perform action. The observer gave instructions to the agents involved during simulation.

For modeling purpose, we defined four agents exist in each mobile robot i.e. robot agent itself, proximity sensor agent, network (communication) agent, and motor agent to drive differential wheel as depicted in **Figure 4**.

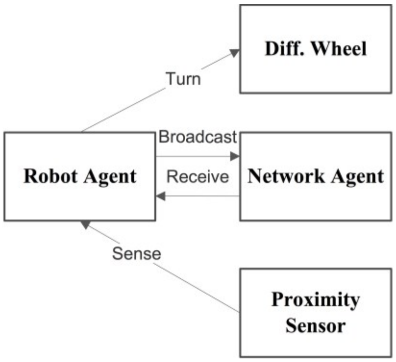

Figure 4 Definition of agents for simulation purpose

Network Agent

Every robot was equipped with a network agent to communicate with each other to exchange its current heading to other robots while in zones of attraction. The functionality of this network agent was to handle explicit communication among robots. This network agent would execute Algorithm 2 to broadcast and receive messages concurrently. For instance, roboti broadcasted information of its heading to other robots. While broadcasting, robot$_i$ might also receive information of robot$_j$'s heading. The implementation of network agent in Netlogo is presented in the following scripts.

```
to make-network
let zoa (wireless-radius)
let wlan-range (zoa)
let connected robots with[self != myself and
distance myself <=zoa ]
ifelse any? Connected
[ask one-of robots
create-links-from other robots in-radius zoa]
receive ([heading] of robots)]
[broadcast ([heading] of robots)
[ask links with [zoa > wireless-radius] [die]]
end
```

Lines 2-4 are the declaration of parameters. Line 5-10 is "if" condition. If other robot within zone of attraction applies, then communication topology is created. Otherwise, topology is disconnected.

Robot Model

It was assumed that a robot made use of differential wheel drive. The robot was also equipped with a wireless network antenna that covers up to *zoa* (zone of attraction), and five proximity sensors that were to cover 270^0 within *zor* (zone of repulsion). The illustration of robot used in our model is depicted in **Figure 5**.

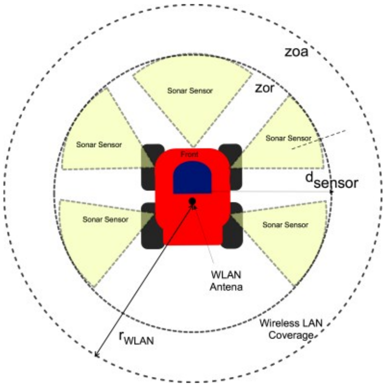

Figure 5 Robot agents in Netlogo

Netlogo doesn't support direct computational of kinematics and dynamics of mobile robots. However, it was assumed that our simulated mobile robot follows a kinematics as explained in Dudek and Jenkin [14].

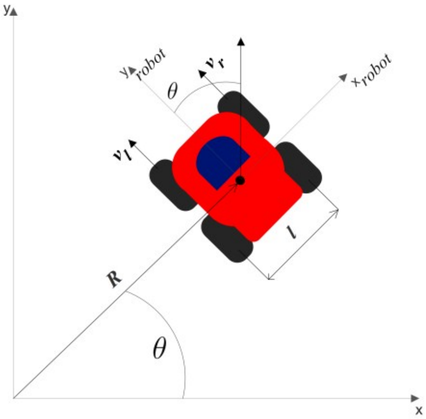

Figure 6 Kinematic model differential drive robot

Differential drive mobile robot could perform rotation about a point that lies along their common left and right wheel axis. The point that the robot could rotate about is known as the ICC (Instantaneous Centre of Curvature). Suppose ω was the rate of rotation of robot body due to differential wheel, and *l* was the lateral distance between wheels. Regarding **Figure 6**, thus we can calculate VR and VL as follow:

$$R = \frac{l}{2}\frac{V_L + V_R}{V_L - V_R}; \omega = \frac{V_R - V_L}{l} \qquad (1)$$

In differential drive mobile robot, there were three interesting condition regarding equation 1. For $V_L = V_R$, the robot has forward linear straight motion, R becomes infinite, and no rotation because ω is zero. For $V_L = - V_R$, R = 0, and the robot rotates in place. For $V_L = 0$, robot turns left because the left wheel rotates. It is the same for $V_R = 0$, the right wheel rotates. The implementation of robot motion in Netlogo was assumed to be as simple as possible, as illustrated in the following script.

```
1:    ....
2:    ask robots [fd speed]
3:    .....
4:    ask robots  [if any? zor
5:    [if any? Collide
6:    [ carefully [ separate ]
7:    [iflse .....
8:    [ ask myself [ rt theta fd speed]
9:    [ ask myself [ lt theta[fd speed ]
10:   ]
11:   …..
```

Initially, robot moved forward with certain speed such as 0.1 patches per time tick in Netlogo (line 2). Suppose the $robot_i$ detected other robots in zone of repulsion *zor*. To avoid collision, the $robot_i$ had to perform separation from them and randomly turn left θ degree with command in Netlogo *lt theta fd 0.1.* To turn right θ degree, $robot_i$ executed command *rt theta fd 0.1*, which means once the robot had turned theta degree right, the robot further moved forward with value of speed 0.1 patch per tick in Netlogo (line 8 - 9).

B. Simulation Results

Flocking Behavior

Simulation was run to see the flocking behavior of swarm robotics in which explicit communication was used. The arena of simulation is a 35 x 35 patch in Netlogo environment. For this purpose, the swarm $robot_i$ was run in two different arenas e.g. the arena without obstacle and the arena with obstacle in the middle. Since there was no specific mission, we only define one specific goal i.e. doing flocking in the defined arena.

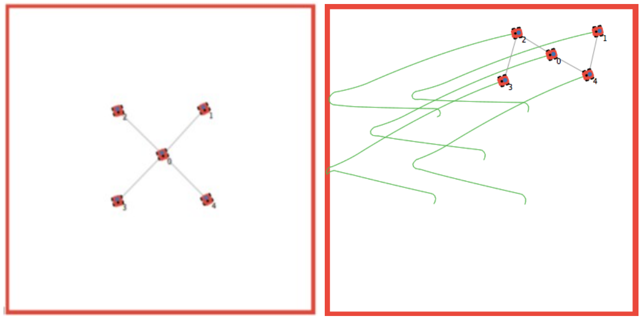

Figure 7 Simulation results of flocking using NetLogo. Left: Initial position. Right: The flocking robots are moving together, while maintaining communication. Robots are moving in the same direction

As depicted in **Figure 7**, each robot initially had different heading. However, regarding their position, it was possible to make communication topology. Once the communication topology had been created, the flocking message was transmitted. In our flocking algorithm, there was no leader hence the robot would randomly follow other robot. Furthermore, the robot started changing their heading. As the results, they move together in flocking formation, as depicted in the right of **Figure 7**.

When there were obstacles in the arena, swarm robots behaved differently. In **Figure 8**, the arena was restricted with red obstacle in the middle. The initial position was the same with initial position in **Figure 7**. Suppose each robot could detect the existence of other robot, therefore communication network could be built immediately to carry flocking instruction. However, robot 4 could not turn to the same heading as robot 1, 2 and 3. This was because of the obstacle lied in front of robot 4, hence robot 4 could not maintain formation as well as its communication topology. As depicted in **Figure 8**, robot 4 chose different angle and moved to different direction to avoid obstacle. The network could be considered disconnected.

Performance of Flocking

The objective of the work presented in this paper is to measure the efficiency of explicit communication used in swarm robotics flocking behavior. Since broadcast message was used to carry flocking instruction, we observed the number of broadcast messages that affected to successful flocking actions performed by swarm robots. We measured the ratio of number of broadcast messages to the number of flocking actions in our first performance model.

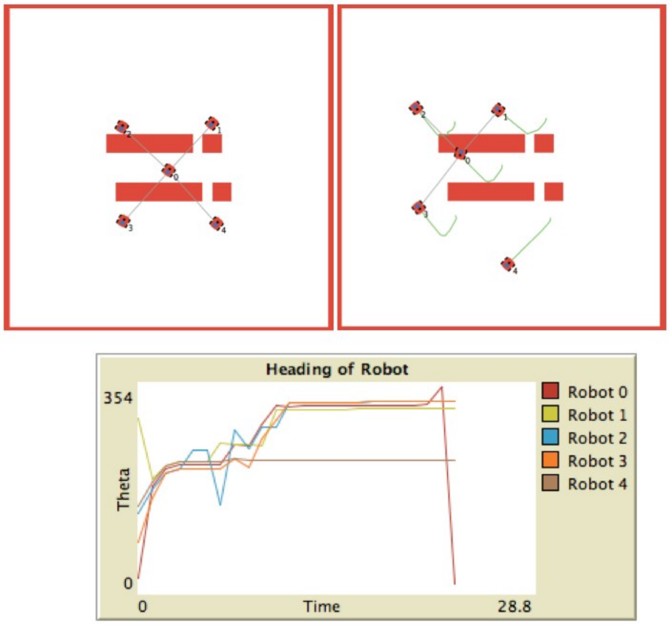

Figure 8 Top left: Initially, each robot built communication topology. Top right: Obstacles were detected and 4 robot moved towards the same direction while robot 4 chose different direction due to the existence of obstacle. Bottom: The angle of robots were observed and plotted

We observed the number of flocking actions of swarm robot when no explicit communication (no comm.) was used. Without explicit communication, robots made use their proximity sensors, e.g. sonar sensor, to form flocking formation, and no information exchange among the robots. It can be seen in **Figure 9** when explicit communication was used with certain radius of transmission, flocking actions might increase in number compared to non-communication flocking. Greater radius of wireless communication increased more flocking action. This is because the number of flocking actions was determined by the flocking instruction received by robot$_j$ upon receiving broadcast information from robot$_i$.

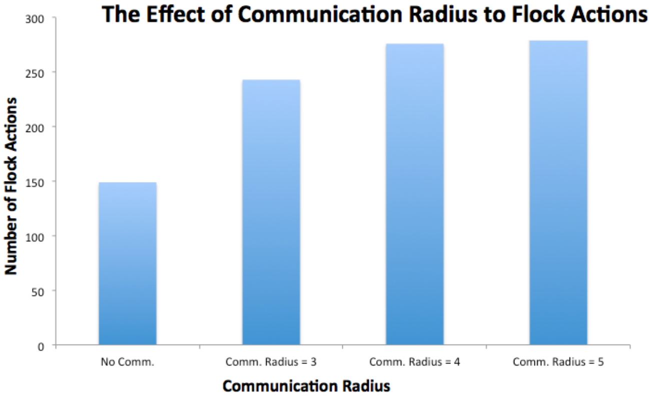

Figure 9 Effect of communication radius to flocking actions

We analyzed the performance of communication-assisted flocking algorithm by observing the ratio of broadcasted message to the flocking actions, denoted by r_{bf}. This ratio represents the effectiveness of the use of explicit communication in flocking behavior, which is defined as

$$r_{bf} = \frac{\overline{B_m}}{F_a} \tag{2}$$

where

$$\begin{aligned} \overline{B_m} &= \frac{1}{n}\sum_{i=1}^{n} B_{m_i} \\ \overline{F_a} &= \frac{1}{n}\sum_{i=1}^{n} F_{a_i} \end{aligned} \tag{3}$$

$\overline{B_m}$ is the average number of broadcast messages, $\overline{F_a}$ is the average number of flocking actions upon receiving broadcast message, and n is simulation duration. Smaller value of r_{bf} is considered to be more efficient, which means that less number of broadcast messages is required to make swarm robots flock. The period of broadcast messages was varied in every 1 tick, 5 ticks and 10 ticks. Ten ticks period means that flocking messages were broadcasted every 10 ticks in Netlogo. The number of robots involved in flocking behavior varies from 5, 10, and 20 robots, and the simulation was stopped at 300 ticks.

Regarding the ratio r_{bf} in **Figure 10**, **11**, and **12**, the more robots were involved in simulation, the less broadcast messages were required to make flocking formation. Since greater number of broadcasting messages require more energy to carry the flocking information, for the purpose of flocking formation, less broadcast messages is considered more efficient. We also observed the effect of broadcast period to the number of flocking actions. Longer broadcast period caused greater ratio r_{bf}. For the flocking formation purpose, it is considered less efficient compared to shorter period of broadcast. The summary of flocking performance can be seen in **Tabel 3**. Regarding **Table 3**, at the end of simulation time, the ratio r_{bf} of 20 robots with period of broadcast 10 ticks was higher than that with broadcast period 1 tick. It means that the longer period of broadcast messages caused less flocking actions in swarm robotics behavior. In this performance measure, we consider that smaller r_{bf} value can be relatively considered more efficient.

Since explicit communication was used to broadcast the flocking information, the communication radius had to be considered in the design of real swarm robots. Greater communication radius requires more power to transmit flocking information state. As depicted in **Figure 9**, when no communication mode was used, the number of flocking action actually depended on the proximity sensors. When communication assisted was used, the actions of flocking significantly increased, it is because longer communication range triggered swarm robots to execute flocking algorithm early.

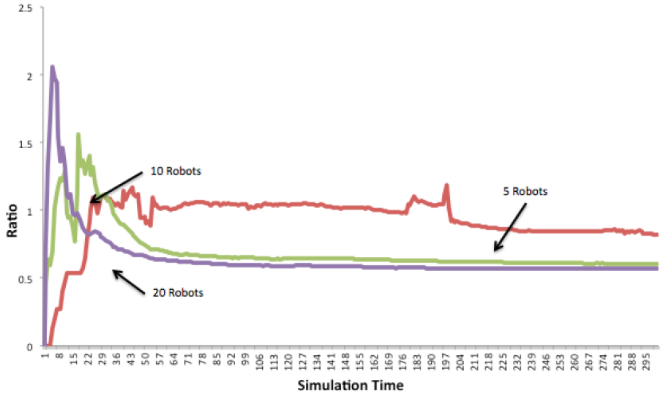

Figure 10 Ratio of broadcast to flocking action. Flocking message is broadcasted every 1 tick

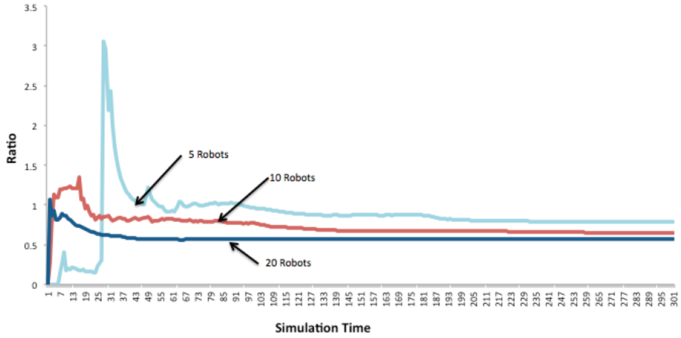

Figure 11 Ratio of broadcast to flock action. Flocking message is broadcasted every 5 ticks

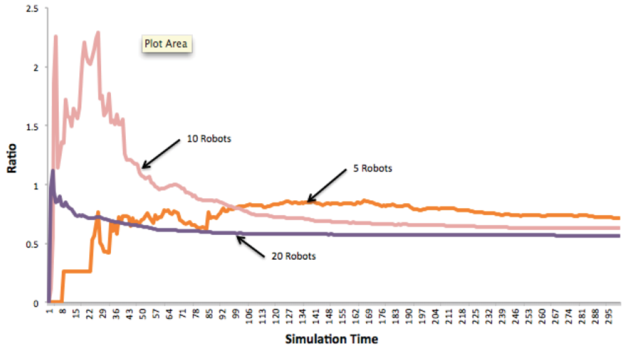

Figure 12 Ratio of broadcast to flock action.Flocking message is broadcasted every 10 ticks

Table 3 The Average of Broadcast to Flock Ratio r_{bf}

Period of Broadcast	5 Robots	10 Robots	20 Robots
1 tick	0.76	0.6	0.56
5 ticks	0.77	0.6	0.56
10 ticks	0.7	0.68	0.57

The second performance that we analyzed in this paper is about swarm robots formation that affected the connectedness of communication topology. The characteristics of connectedness are determined by second smallest eigenvalues of adjacency matrix in communication graph G. To observe the connectedness, we run the following simulation as depicted in **Figure 13**.

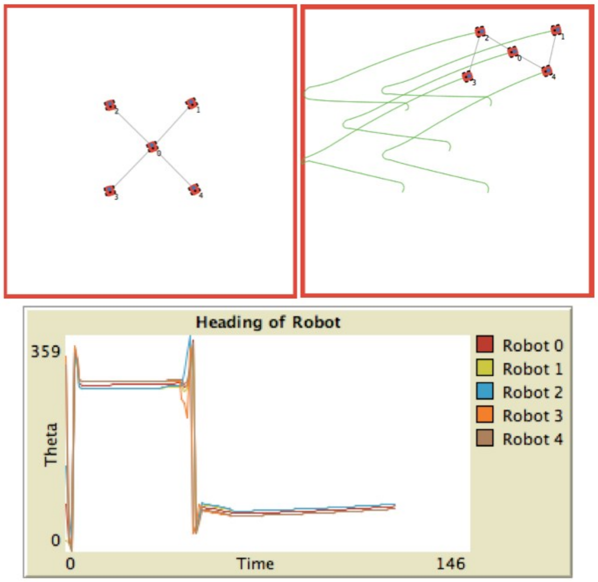

Figure 13 Simulation scenario to analyze the connectedness of communication topology affected by swarm robot flocking formation. Left: Initial position of swarm robots. Right: Swarm robots moves and flocks.

Let us consider our flocking formation that generates communication networks, or communication graph as shown in **Figure 13**. The interactions among swarm robot can be represented as a communication graph $G = (V, E)$, where $V = 1 \ldots N$ is the set of swarm robots. Edge $E \subseteq \{V \text{ x } V\}$ is the set of communication link, where $E = \{(i, j) \mid i, j \in V\}$, and $i \neq j$. Suppose the node i in graph G has neighbors N_i. Then $N_i = \{j \in V \mid (i; j) \in E\}$. Neighboring node determines the model of proximity in which the range of interaction may occur. Let proximity graph G is based on Euclidian distance. By taking into account the distance between two robots with radius r, the neighboring nodes covered in the range of interaction can determine the connectivity of graph G. Graph connectivity can further be determined by the adjacency matrix $A = a_{ij} \in \mathbf{R}^{n \times n}$, in which element a_{ij} defines the weight of connection link between the swarm robot i and j , with a value of 1 if $j \in N_i$ and zero otherwise.

The degree of graph G describes the number of links connected from a node, which can be defined as matrix $D = diag(\{d_i\})$, where $d_i = \sum_{i=1}^{n} a_{ij}$ is the degree of node i in graph G. Laplacian matrix $L(G)$ is the difference between degree matrix D and adjacency matrix A of graph G. L explains the number of spanning tree of node i, that is $L = D - A$. The element of L is defined as follows:

$$l_{ij} = \begin{cases} \deg(v_i) & ;if \qquad i = j \\ -1 & ;if \quad i \neq j, v_i \sim v_j \\ 0 & ;if \quad \text{otherwise} \end{cases} \tag{4}$$

The eigenvalues of a matrix L are precisely the solutions to the equation $det(L - \lambda I) = 0$. Generally, some well-known properties of a Laplacian matrix L shows that the eigenvalues of $\boldsymbol{L}$, is λ_i, $i = 1 \ldots N$ [15]. The values of $\lambda_2 > 0$ if and only if graph G is connected graph, and λ_2 is the algebraic connectivity of graph G if $a_{ij} = 1$, and 0 if otherwise.

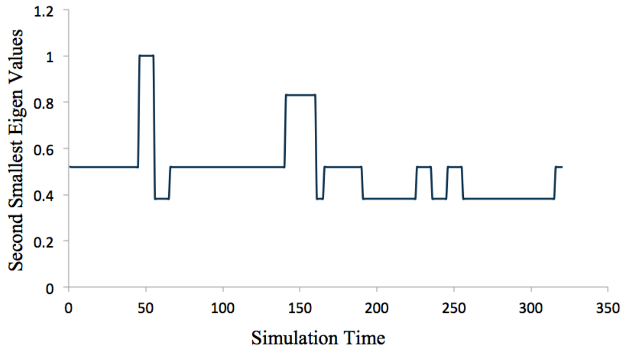

Figure 14 The Eigenvalue of Swarm Robotics Network during formation in Fig. 13

We plot the second smallest eigenvalues during simulation, and the result is depicted in **Figure 14**. From **Figure 14**, it can be inferred that communication-assisted flocking behavior relies on communication graph. Without communication graph, it is impossible to disseminate flocking instruction to all swarm robots. The formation of flocking robot also determines the connectedness of communication graph that is required to form flocking formation. Based on **Figure 14**, flocking formation makes communication graph formation connected, which is indicated by the second smallest eigenvalues always greater than 0. Therefore, flocking formation can also be used to maintain communication graph, and the communication graph itself is required to maintain the flocking formation

V. CONCLUSION

This paper presents a flocking behavior in swarm robotics that makes use of explicit communication via broadcast messages over wireless network. The simulation was carried out in Netlogo, a multi-purpose multi-agent simulation system. From the simulation results, it can be seen that explicit communication via broadcast messages was efficient enough to be considered in the flocking behavior, by varying the number of robots involved as well as the period of broadcasting message. With less number of broadcast messages, it could still affect flocking actions of swarm robotics during the simulation. The second performance measure shows that flocking formation was able to make communication graph formation connected as well, which is indicated by the second smallest eigenvalues always greater than 0. Therefore, in this situation, flocking formation can also be used to maintain communication graph, and the communication graph itself is required to maintain the flocking formation. In this simulation, the communication constituted complete graph. As future work, we will extend this model by involving minimum communication cost by reducing graph topology to perform explicit communication and analyze the stability of flocking formation that affects stability of communication topology.

REFERENCES

[1] T. Balch, Ronald, and R. C. Arkin, "Communication in reactive multiagent robotic systems," Autonomous Robots, vol. 1, pp. 27– 52, 1994. CrossRef

[2] J. Penders, "Robot swarming applications," in Liber Amicorum ter gelegenheid van de 60e verjaardag van Prof.dr. H. Jaap van den Herik, J. Donkers, Ed. Maastricht, Netherlands: Maastricht ICT Competence Center, October 2007, pp. 227–234.

[3] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in ACM SIGGRAPH Computer Graphics, vol. 21, no. 4. ACM, 1987, pp. 25–34.

[4] U.Wilensky. (1998) Netlogo flocking model. [Online]. VIEW ITEM

[5] I. Couzin, J. Krause, R. James, G. Ruxton, and N. Franks, "Collective memory and spatial sorting in animal groups," Journal of theoretical biology, vol. 218, no. 1, pp. 1–11, 2002. CrossRef

[6] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," Automatic Control, IEEE Transactions on, vol. 51, no. 3, pp. 401–420, 2006. CrossRef

[7] N. Bouraqadi, A. Doniec et al., "Flocking-based multi-robot exploration," Control Architectures of Robots, 2009.

[8] A. Turgut, H. Celikkanat, F. Gokce, and E. Sahin, "Self-organized flocking in mobile robot swarms," Swarm Intelligence, vol. 2, no. 2, pp. 97–120, 2008. CrossRef

[9] H. Celikkanat, A. Turgut, and E. Sahin, "Guiding a robot flock via informed robots," in Distributed Autonomous Robotic Systems (DARS 2008), Berlin, Germany, 2008, pp. 215–225. CrossRef

[10] Chibaya, Colin and Bangay, Shaun, "Flock Inspired Area Coverage Using Wireless Boid-Like Sensor Agents" in Proceedings of the Tenth International Conference on Computer Modeling and Simulation, Washington, DC, USA, 2008, pp. 144—149.

[11] C. Moeslinger, T. Schmickl, and K. Crailsheim, “A minimalist flockingalgorithm for swarm robots,” Advances in Artificial Life. Darwin Meets von Neumann, pp. 375–382, 2011. CrossRef

[12] S. Momen, B. Amavasai, and N. Siddique, “Mixed species flocking for heterogeneous robotic swarms,” in EUROCON, 2007. The International Conference on” Computer as a Tool”. IEEE, 2007, pp. 2329–2336.

[13] J. Nembrini, A. F. T. Winfield, and C. Melhuish, “Minimalist coherent swarming of wireless networked autonomous mobile robots.” In Proceedings of the 7th international conference on the simulation of adaptive behavior conference, vol. 7, 2002, pp. 273–382.

[14] G. Dudek and M. Jenkin, “Computational Principles of Mobile Robotics,” Cambridge University Press, 2010. CrossRef

[15] F. Chung, “Spectral graph theory,” Amer Mathematical Society, 1997, vol. 1992.

[16] Bayu Erfianto and Bambang Riyanto T. Swarm Robot Flocking Assisted by Explicit Communication. International Conference on Intelligent Unmanned System – ICIUS 2012, Singapore.