

Cooperative Avoidance Control-based Interval Fuzzy Kohonen Networks Algorithm in Simple Swarm Robots

Siti Nurmaini¹, Siti Zaiton², Ricy Firnando³

^{1,3} Robotic and Control Research Lab, Faculty of Computer Science, University of Sriwijaya,
Jl Raya Palembang-Prabumulih, Km 32, Inderalaya-Oganllir, Indonesia

² Soft Computing Research Group, Faculty of Computing, Universiti Teknologi Malaysia
81310 Skudai, Johor Bahru, Malaysia

e-mail: siti_nurmaini@unsri.ac.id¹, sitizaiton@utm.my², blacklist161@yahoo.co.id³

Abstract

A novel technique to control swarm robot's movement is presented and analyzed in this paper. It allows a group of robots to move as a unique entity performing the following function such as obstacle avoidance at group level. The control strategy enhances the mobile robot's performance whereby their forthcoming decisions are impacted by its previous experiences during the navigation apart from the current range inputs. Interval Fuzzy-Kohonen Network (IFKN) algorithm is utilized in this strategy. By employing a small number of rules, the IFKN algorithms can be adapted to swarms reactive control. The control strategy provides much faster response compare to Fuzzy Kohonen Network (FKN) algorithm to expected events. The effectiveness of the proposed technique is also demonstrated in a series of practical test on our experimental by using five low cost robots with limited sensor abilities and low computational effort on each single robot in the swarm. The results show that swarm robots based on proposed technique have the ability to perform cooperative behavior, produces minimum collision and capable to navigate around square shapes obstacles.

Keywords: cooperative avoidance, swarm robots, interval fuzzy-kohonen networks

1. Introduction

A swarm is a complex adaptive system, which is decentralized, self-organized and whose individuals are simple, homogeneous and autonomous [1]. In terms of robotics is an approach for coordinating multi-robot system. This is the fields considering a group of relatively simple individuals able cooperates to perform complex tasks, in a decentralized manner [2]. The concept of swarm robots is based on nature. The inspiration founds in the first line within animal societies, such as birds, ants and bees. Social insects exhibit successful behavior in performing complex tasks on the level of the group, and are able to eliminate noise, error, and failure of swarm members. The relevant properties of many of these biological systems are reviewed in [3],[4]. The main objective of this approach is the idea that collective intelligence can arise from the interaction of a high number of relatively simple units [5]. In swarm robot systems, each robot must behaves by itself according to the its states and environments, and if necessary, must cooperates with other robots in order to carry out a given task.

Today, number of swarm robotic system have been proposed for numerous applications where human intervention is not feasible such as radioactivity detection, firefighting and landmine detection, the robots need to be dispensable [6]. A large number of robots allow for redundancy and increase the robustness of the swarm. The increasing interest in swarm robotic system indicates that employing multiple inexpensive and simple mobile robots as opposed to a single expensive [7]. An expensive robot may be able to achieve the task but its failure can prove to be costly and dangerous in mission critical applications. By building swarm of robots with elementary features, the same task can be achieved for a lower cost and increased reliability [6]. In search applications also have an advantage of larger coverage of the search space and its simplicity of implementation. They could perform exploration tasks in a large-scale area more efficiently [8]-[10] due to, the swarm shares information about the environment and individual members interact with each other [11]. Without being able to process and respond to new information, a robot loses its ability to adapt.

The swarm robots need to be able to process and act upon any new information from its environment. Otherwise, the robots will have to rely on a static set of rules which may be

inadequate if the robot's circumstances change significantly. In the new environment, they must be capable of avoiding obstacles, controlling the group speed and modifying the inter-robot distance [12],[13]. Several control action solutions are proposed where robots agree on their speeds; therefore they need to know the speeds of the neighbors' robots [14]-[17]. However, the implementation of such an algorithm on real robots can be quite difficult due to the difficulty in obtaining the neighbors' speeds by observation and it produces a large amount of computational resources. In addition, forcing each robot to carry such resources internally means duplicating, perhaps unnecessarily and expensive assets [10]. The capability of the controller, sensors and communication system is a significant performance parameter for swarm robots. However, the economic cost problem is often associated with swarm applications due to the large number of robots required and the building of intelligent robots with a number of sensors for various parameters is expensive. Thus, a low cost swarm robot platform with inexpensive sensors for specific single and multi task scenarios is highly favourable.

The development of control algorithms for swarm robots is a challenging task as the global behavior that emerges from the many interactions between the robots is often hard to predict and experiments with swarm robots are often expensive and time consuming. In the present paper, a new algorithm Interval Fuzzy-Kohonen Network (IFKN) is introduced in swarm robots research and demonstration of techniques for smaller multi-robot systems. This paper is organized as follows: Section 2 presents cooperative behavior of swarm robots. Section 3 describes the design process of the proposed control strategy based on Interval Fuzzy-Kohonen Network (IFKN) algorithm. Section 4 discusses the experimental design and results. Section 5 gives the conclusion and future work.

2. Proposed Method

Behavior-based design is the most common way to develop a swarm robotic system. In an iterative way, the individual behavior of each robot is implemented, studied and improved until the desired collective behavior is obtained. In behavior-based swarm robots design, inspiration is often taken from the observation of the behaviors of social animals. The individual behaviors of the robots which result in the collective behavior of the swarm. They must integrate several goal-oriented behaviors concurrently in order to reach a goal. In this situation, multi-robots would communicate with each other and cooperate to execute a specific global behavior.

With the increasing requests on the application of multi-robot systems for complex tasks, the obstacle avoidance behavior of a multi-robot system becomes more and more crucial [18]. Many algorithms have been proposed for obstacle avoidance of a single robot. In addition, it is found that realizing obstacle avoidance of each robot in a multi-robot system is not enough. Further, it requests multiple robots to have the ability to avoid obstacles while keeping formation. If a multi-robot system in an unknown environment, it becomes more difficult to realize obstacle avoidance while keeping formation [19]. This system can improve the effectiveness of a robot's action in terms of performance and reliability. However, the difficulty arises in coordination and cooperation of multi-robots to perform a single, global task. There are important aspects such as how to design the local behavior action of each single robot and how to control the cooperative behavior of a swarm?

The chance of robots colliding against each other is yet another challenge in swarm robots formations. Some researchers use the term collision avoidance synonymous with obstacle avoidance [20],[21]. Collisions are avoided by maintaining strict buffer distances and consistent communication between robots [22]. Cooperative avoidance behavior tends to be based on speed adaptation, route deviation by one robot only and route deviation by both robots, also a combined speed and route adjustment. Each robot could communicate with surrounding robots in dynamic environmental perception and it should be able to determine at least one of the pieces of information concerning the relative position, orientation, and speed of other robots. However, the ability of the robot to communicate depends on the computational resources and also the type and amount of sensors that are employed on the robots [23].

Multiple robots require frequent updating of sensor-based information between each individual unit and they must manage a large array of sensory information to determine their environment. Each sensor provides some input about the environment around the robots. That input being incorporated into a knowledge base. From this knowledge base, appropriate strategy

about control actions taken in response to the input is generated. These actions allow the robots, to interact with its surroundings for achieving the goal. However, creating and maintaining these control actions, as well as gathering new data for the knowledge base poses significant challenges. This is due to memory and processing power are issues in this research area.

Fuzzy logic has been a widely used method for swarm robots [6],[12],[25]. Other methods explored in literature include navigation based on intelligent data carrier systems [14], neural network [25],[26], particle swarm optimization [27],[28], support vector machine [17], Foraging algorithm [29] have been explored for this purpose. However, in the hardware implementation, some instrumentation elements usually introducing some sort of unpredictable values in the collected information of environmental measurement, called uncertainty, such as amplifier, sensors, digital to analog (DAC), analog to digital converters (ADC), and actuator. These uncertainties cause difficulty in determining the control action in real time situation. While control actions are determined and tuned in certain environmental conditions, it might have to be changed in other environments [30],[31].

Fuzzy logic systems (FLSs) employ a mode of approximate reasoning that makes them a suitable tool to implement a robust robot behavior tolerating noisy and unreliable sensor information. However, the FLSs have the common problem that they cannot fully handle or accommodate for the linguistic and numerical uncertainties associated with changing and dynamic environment because they use precise fuzzy sets. Hence, it needs a method to overcome the uncertainty problem to realize a safe and efficient of swarm robots movement. It means the swarm robots do not collide with both of obstacles and robots, and they can reach a destination in a small amount of time.

In the real world, numerous natural agents like animals to recognize their environments just with low sensitive sensors without a geometric map [32]. They must learn to recognize the new environment by itself. Hence, developing adaptive technique in real environment for swarm behavior is desirable. This paper describes a practical implementation that highly intelligent and capable swarm robots for cooperative avoidance through environmental recognition based on a new algorithm of interval fuzzy-kohonen network (IFKN). The algorithm is implemented that integrates groups of smaller low-cost robots, which are equipped with limited-range communication ability and inexpensive distance sensors.

3. Research Method

In this paper, the Fuzzy-Kohonen Network (FKN) algorithm [30],[31] is extended to an Interval Fuzzy-Kohonen Network (IFKN) algorithm. To overcome the uncertainty in some elements instrumentation in the hardware implementation, interval fuzzy sets are used. Interval means that the input domains are characterized by interval fuzzy sets. In the IFKN algorithm design only few rules are created compare FKN algorithm for generating speed and steering angle of swarm mobile robots. Kohonen network has the advantage of the patterns recognition mechanism while the interval fuzzy logic plays a role in managing the input and output process of pattern recognition. The IFKN control algorithm is implemented on each robot to decide its action and computational metaphor inspired by social insects.

3.1. Pattern Classification

In order to enable the mobile robot to avoid the obstacles with reactive action, the better mapping relation between the sensor data as input and the speed control as output must be established. Environmental classification is modelled to summarize all environmental patterns as described in Figure 1. Each sample of such pattern consists of range readings obtained by five sensors. For reducing complexity of storage, representation and learning a long sequence of each sample is mapped to one class by IFKN algorithm. All possibilities of mobile robot environments are considered through fuzzifying process and combining these 15 classes of interval fuzzy rule base. The rule table is constructed exploiting the sequence of environmental pattern and speed levels. In this strategy 15 rules are employed to keep few compare to FKN algorithm and IFKN algorithm.

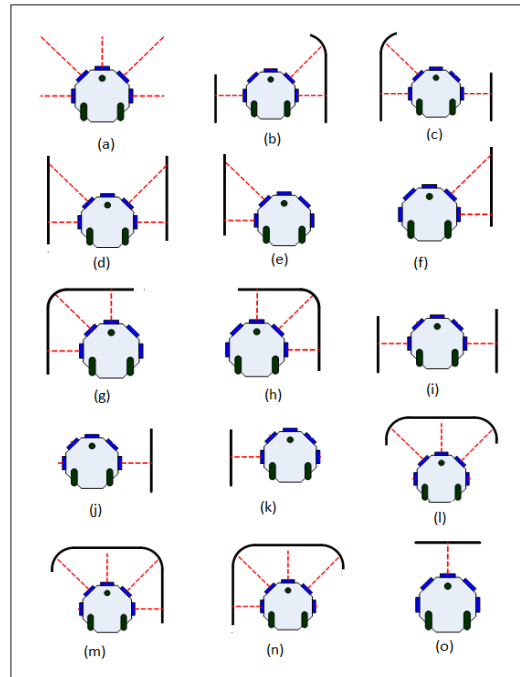


Figure 1. Environmental classifications

3.2. Interval Fuzzy-KohonenNetwork (IFKN) Design

The IFKN algorithm is proposed to incorporate learning rules to determine the distance and similarity between the input and initial patterns. Infrared sensors are in general used to obtain information regarding the local environment. The input patterns are constructed from current sensor readings and the algorithm design is described in the following steps:

Step 1. Fuzzification design

It is obvious that the interval fuzzy sets are in a region constructed by principal fuzzy set membership functions (FSMFs). An interval fuzzy system is obtained by using the fuzzy sets to partition the input domains of the baseline FSMFs with an interval of uncertainty as shown in Figure 2. The inputs (X_i) which are distances of the obstacles to the sensors. Fuzzification stage of interval fuzzy system produces four degrees of input membership, such as upper near, lower near, upper far, and lower far as shown in Figure 2. The membership functions that determine the degree of farness or nearness to the obstacle are defined as follows,

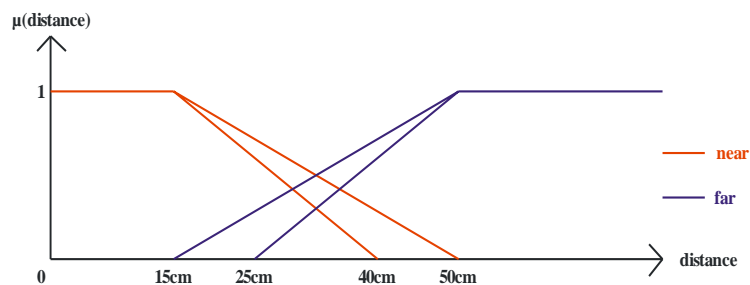


Figure 2. Interval fuzzification stage

In this work triangular membership functions (MFs) are used. The upper and lower MFs for interval fuzzy sets can be written in equations (1), (2), (3) and (4) respectively,

$$\bar{\mu}_{\text{near}}(x_i) = \begin{cases} 1, x_i \leq 15 \\ \frac{50-x_i}{50-15}, 15 < x_i < 50 \\ 0, x_i \geq 50 \end{cases} \quad (1)$$

$$\underline{\mu}_{\text{near}}(x_i) = \begin{cases} 1, x_i \leq 15 \\ \frac{40-x_i}{40-15}, 15 < x_i < 40 \\ 0, x_i \geq 40 \end{cases} \quad (2)$$

$$\bar{\mu}_{\text{far}}(x_i) = \begin{cases} 0, x_i \leq 15 \\ \frac{x_i-15}{50-15}, 15 < x_i < 50 \\ 1, x_i \geq 50 \end{cases} \quad (3)$$

$$\underline{\mu}_{\text{far}}(x_i) = \begin{cases} 0, x_i \leq 25 \\ \frac{x_i-25}{50-25}, 25 < x_i < 50 \\ 1, x_i \geq 50 \end{cases} \quad (4)$$

Step 2. Euqlidian distance

For calculating the Euqlidian distance in upper and lower membership d_{ij} that is responsible for comparing the input pattern $X_{i \text{ upper}}(\bar{X}_i)$ and $X_{i \text{ lower}}(\underline{X}_i)$ with every initial pattern W_{ij} and find the winner takes all neuron. The two equations can be written in equation (5) and (6) respectively,

$$\bar{d}_{ij} = \|\bar{X}_i - \bar{W}_j\|^2 = (\bar{X}_i - \bar{W}_j)^T (\bar{X}_i - \bar{W}_j)_j \quad (5)$$

$$\underline{d}_{ij} = \|\underline{X}_i - \underline{W}_j\|^2 = (\underline{X}_i - \underline{W}_j)^T (\underline{X}_i - \underline{W}_j)_j \quad (6)$$

Where, \bar{d}_j is upper euclidean distance, \underline{d}_j is lower euclidean distance, n is the number of sensor, w is prototype pattern and j is pattern index. There are many initial patterns, which represent a characteristic pattern in every layer. The originality of the learning process in kohonen network is unsupervised, however that is taking a long time in the training proces, to find the weights that meet the good performance. To reduce the space complexity and to facilitate fast learning of sample sequences by the IFKN algorithm, all these patterns are set as a weight in the distance layer.

To obtain information regarding the environment it is essential to keep track of changes in the pattern of a sequence. Often the changes in the sequence are more important than simply the sequence itself. For simple calculation, the rule base table is utilized. The plan of the rule's table is created based on environmental classifications, where, S_1 is sensor 1, S_2 is sensor 2, S_3 is sensor 3, S_4 is sensor 4, and S_5 is sensor 5. The number of rules equals with environmental patterns. In this work, initial pattern are derived from previous experimental data base. The patterns are associated with a pair of motor speed reference M1 and M2 as shown in Table 1.

Step 3. Rule tables

Once the similarity value is obtained by using the Euclidean distance, then the degrees of membership μ_j can be calculated. There are two similarity values of upper and lower membership are referred to the equation (7) and (8) respectively,

$$\bar{\mu}_{ij} = \begin{cases} 1, \bar{d}_{ij} = d_{\min} \\ \frac{d_{\max} - \bar{d}_{ij}}{d_{\max} - d_{\min}}, d_{\min} < \bar{d}_{ij} < d_{\max} \\ 0, \bar{d}_{ij} = d_{\max} \end{cases} \quad (7)$$

$$\underline{\mu}_{ij} = \begin{cases} 1, \underline{d}_{ij} = d_{min} \\ \frac{d_{max} - \underline{d}_{ij}}{d_{max} - d_{min}}, d_{min} < \underline{d}_{ij} < d_{max} \\ 0, \underline{d}_{ij} = d_{max} \end{cases} \quad (8)$$

The membership degree μ_{ij} represents the similarity between current patterns X_i and prototype patterns W_j , and $\mu_{ij} \in (0,1)$. Where, $\bar{\mu}_{ij}$ is upper degree of membership, $\underline{\mu}_{ij}$ is lower degree of membership, d_{ijmin} is minimum Euclidean distance and d_{ijmax} is the maximum Euclidean distance. The sum of the membership degree outputs μ_{ij} equal to 1. After the rule base and similarity values are known, then the speed of the motors can be determined by finding the rule base that has the highest level of similarity. The results are calculated with the reference speed.

Table 1. Rule base table of IFKN

Pattern	Sensor Input					Motor Output	
	S1	S2	S3	S4	S5	M1	M2
1	Far	Far	Far	Far	Far	90	90
2	Near	Far	Far	Far	Far	73	90
3	Far	Far	Far	Far	Near	90	73
4	Near	Near	Far	Far	Far	90	83
5	Far	Far	Far	Near	Near	83	90
6	Near	Near	Near	Far	Far	85	55
7	Far	Far	Near	Near	Near	55	85
8	Near	Near	Near	Near	Far	85	53
9	Far	Near	Near	Near	Near	53	85
10	Near	Near	Far	Far	Near	87	77
11	Near	Far	Far	Near	Near	77	87
12	Near	Far	Far	Far	Near	87	87
13	Near	Near	Far	Near	Near	86	86
14	Far	Far	Near	Far	Far	87	60
15	Far	Near	Near	Near	Far	85	53

Step 4. Determine the outputs

Calculate the crisp output value from average membership degree. The average values of the upper and the lower degree of membership as a single degree of membership for each pattern is calculated by using equation (9) below,

$$\mu_j = (\bar{\mu}_{ij} + \underline{\mu}_{ij})/2 \quad (9)$$

After the rule base and maximum similarity patterns (μ_{ijmax}) are known, the mobile robot outputs are determined by finding a rule that has the highest level of similarity from upper and lower index (μ_{jmax}). The final value is calculated by multiplying the similarity level of the maximum degree of membership with percentage of maximum left and right motor speed values, given in equation (10) and (11) as follow,

$$Out_1 = \mu_{jmax} * M_1 \quad (10)$$

$$Out_2 = \mu_{jmax} * M_2 \quad (11)$$

Where, Out_1 is percentage of speed for left motor output, Out_2 is percentage of speed for right motor output, μ_{jmax} is patterns index which has a maximum degree of membership, M_1 left motor output speed percentage and M_2 is output right motor speed percentage for each pattern.

3.3. Hardware Design

The experimental works are conducted to evaluate the swarm’s controller performance. To validate the characteristic of swarm behavioural, the experiments are implemented on five smaller robots. The development of the hardware architecture is the initial work that marks the translation of the designed software into the hardware platform. In this phase all algorithms of embedded control systems as the destination platform, are implemented using a standard microcontroller platform. The swarm robots platform employs AVR ATMEGA 16 microcontroller series as the main processor for managing all modules. They use clock at 8.0 MHz using the internal RC oscillator to provide the necessary computational power for real-time sensory system. This series has 16 kilobytes of programmable flash memory and one kilobyte of internal SRAM that provides more than enough space to implement basic reactive behavior and different complexity of swarm algorithms.

The swarm robots have three infrared sensors for obstacle detection and a digital compass for heading measurement. Communication is the most important aspect of swarm robots therefore multiple individuals can share information to function as a whole. An IEEE 802.15.4/Zig Bee compliant X-Bee wireless module with a range of approximately 20 m indoors are used for communication between robots and between the robots and a computer console. Where the microcontroller will serve the input with its interrupt routine, 20 MHz AVR AT Mega 16 microcontroller, which provides a control step duration of 110 ms is employed. The microcontroller can be programmed through a wireless communication link. The low-power design of its systems lets swarm robots operate for 10 hours with a 2000 mAh Li-Poly battery. Figure 3 shows the detail implementation of hardware design by using Proteus Simulator.

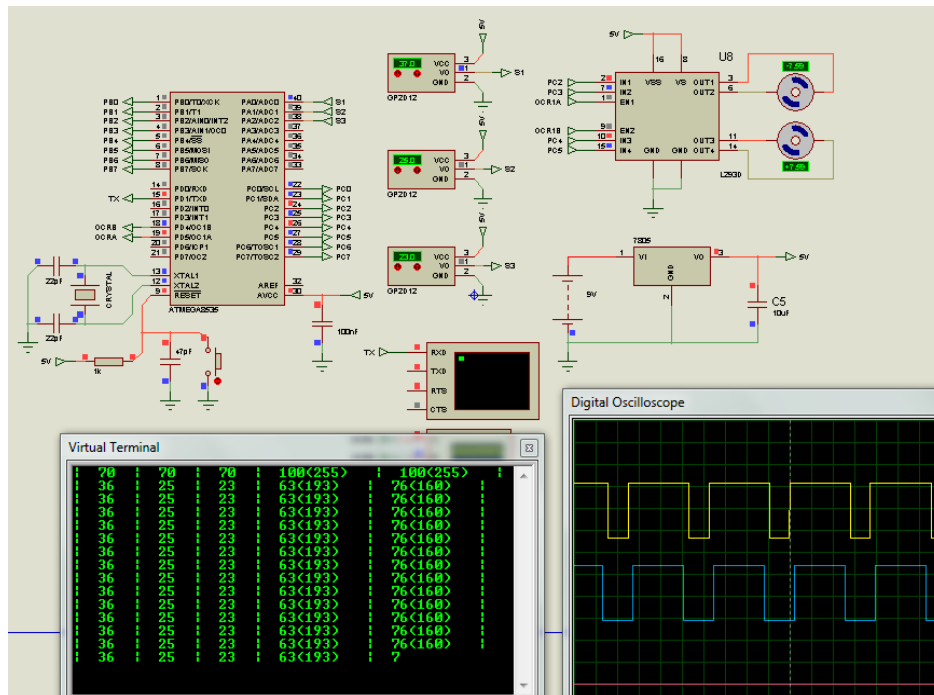


Figure3. Hardware design

4. Results and Analysis

Real-time navigation involves decision making according to the perception of the local environment. The fuzzy inferencing method has been shown to be successful in real-time navigation with cluttered environments [30]. However, when the environment is filled with obstacles in the form of loops, mazes, and other complicated structures the robots tends to lose track of direction and gets trapped. In this section various fundamental benchmark problems in

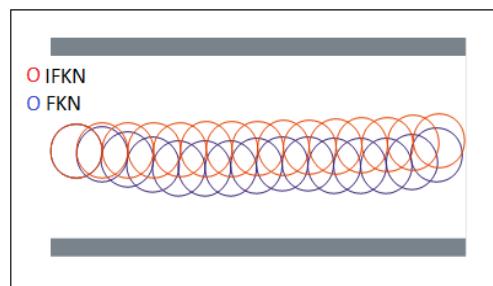
navigation are experimentally demonstrated to allow us to evaluate overall performance and to examine possible faults in recognition and communication between objects.

In the implementation of the proposed IFKN algorithm on real swarm robots, some practical issues are expected to resolve. In this section, some major issues in the light of some preliminary studies are pointed out, with an experimental setup composed of a number of mobile robots. Before implementing the proposed IFKN algorithm, it is necessary to perform some tests in order to determine the structure of the IFKN including number of input patterns and the environmental condition.

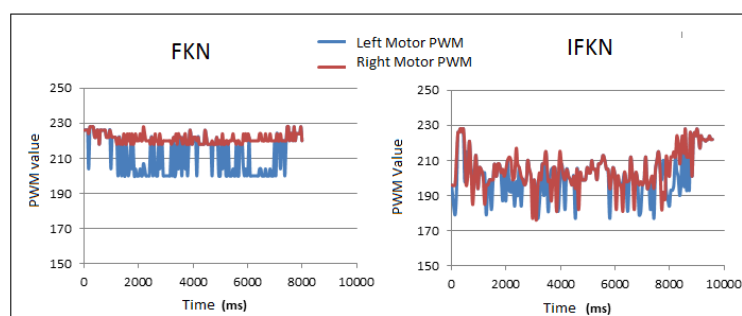
4.1. Obstacle Avoidance Behavior in Single Robot

In order to verify the effectiveness of the proposed algorithm, mobile robot is set up in several environments. In this work, IFKN algorithm is compared to FKN algorithm in single robot action as demonstrated in Figure 4, 5 and 6 respectively for obstacle avoidance behavior. From the results, both of the algorithms have used in single mobile robot successfully to perform obstacle avoidance task. However, mobile robot based on IFKN algorithm produces stable movement, because of in IFKN algorithm environmental patterns changing in interval values. Therefore the motor output produces constant speed.

In the long corridor environment, single mobile robot based on IFKN algorithm more stable compare to FKN algorithm, due to the mobile robot movement always keep on in the centre of the corridor environment as shown in Figure 4(a). The recorded graph of the mobile robot PWM with FKN algorithm and IFKN algorithm when avoids the wall as shown in Figure 4(b).



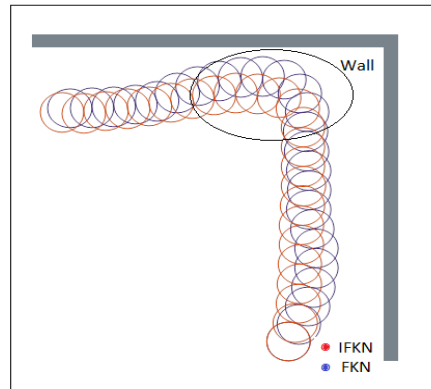
(a) Corridor



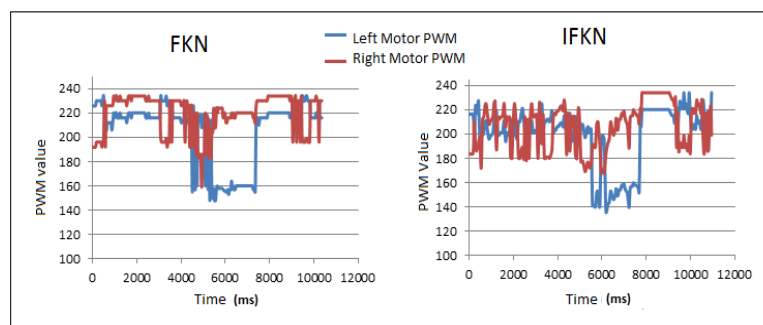
(b) PWM output

Figure 4. Mobile robot movement in the corridor

In the right corner environment, mobile robot with IFKN algorithm produce keep a safe distance from the right wall compare to FKN algorithm. At the corner area by using IFKN algorithm, mobile robot is able to make the movement more stable than FKN algorithm.



(a) Right corner



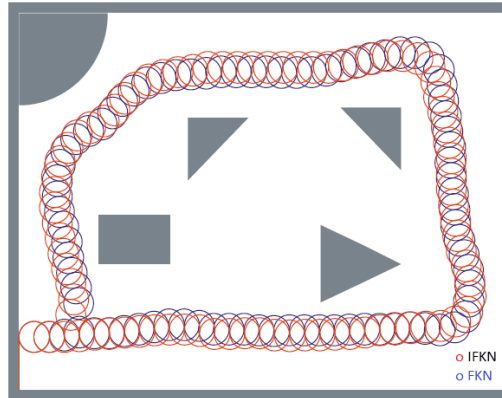
(b) PWM output

Figure 5. Single robot movement in the right wall

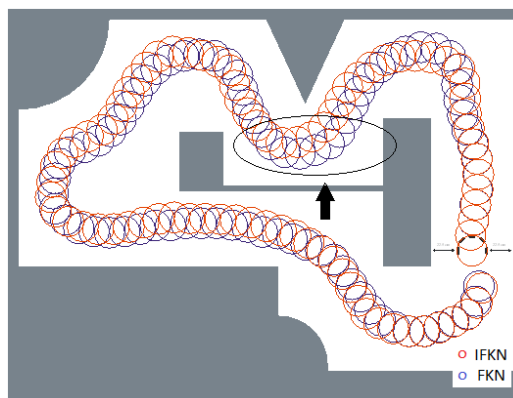
It's seen in Figure 5(a)-(b) mobile robot movement with FKN algorithm moving closer to the wall when it makes turn left action, whereas by using IFKN algorithm produces smooth movement. This is because in IFKN algorithm design, the sensors provide more data within the range of interval for solving imprecision in the sensors system when it detect the environment. While in the FKN algorithm design, sensor data are expressed only in certain values. The mobile robot performance test is conducted by placing a mobile robot in several environments such as simple environment, complex environment and the U-Shafe situation respectively to analysis the comparison of the proposed algorithm. Figure 6(a) shows a trajectory performed by one robot using propose technique in complex environment containing four obstacles.

It can be seen that single mobile robot movement utilizes IFKN algorithm more carefully compare to FKN algorithm, it keeps a distance from the unstructured wall. Experimental results in cluttered environment with more obstacles as depicted in Figure 6(b). In this environment, we dealt with more noise like unstructured wall. The experimental results highlight the fact that by using the IFKN algorithm can enhance environmental sensing capacity. The same fact is observed from the outputs of various experiments performed in different environmental conditions.

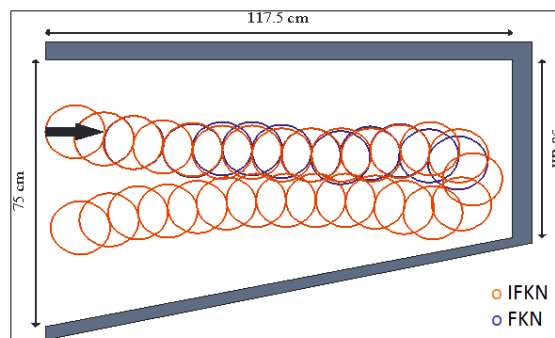
In complex environments, IFKN algorithm can solve a problem commonly encountered during implementations, such as errors in range readings due to multiple reflections at the corners. In complex and unstructured environments. IFKN algorithm are able to produce stable movement by generating only 6 patterns, compared to FKN algorithm produce 14 patterns. Having goal on the side of the wall long-corridor environment may cause a mobile robot to be trapped in a wrong boundary-following direction. The U-shape situation is difficult to be solved because the sensing capability of infra-red sensors and noises in the sensors make it difficult to determine the size or location of obstacles when this information is required for the escape criterion.



(a) Simple environment



(b) Complex environment



(c) U-Shape situation

Figure 6. Mobile robot trajectory

In some condition, avoid-obstacle behavior may not function properly and the mobile robot move too close to an obstacle, which is a U-shape situation. In such case, mobile robot is required to stop the movement or even, in some instances, the mobile robot need to make aturn back move. This condition depends on the safest allowable distance between the mobile robot and the obstacle. This behavior has the highest priority. By using an IFKN algorithm mobile robot is able to make turn left action, but a mobile robot with FKN algorithm getting trapped in a local minimum as shown in Figure 6(c). This strategy produces good action to guide the mobile robot out of the traps.

Table 2. Comparison of FKN and IFKN algorithm

No.	Description	FKN	IFKN
1	Environmental pattern use	21	15
2	Software resources (byte)	3794	3376
3	Time of computation (ms)	42.33	54.60
4	Pattern recognition (%)	85.71	86.66
5	Number of data output/process	939	975

From the experiment, the proposed IFKN algorithm successfully to recognize the environmental patterns, and produce pattern recognition percentage is 86.66%, while FKN algorithm produce 85.71%. The results show that cooperative learning behavior by using IFKN algorithm can achieve fast behavioral adaptations compare to FKN algorithm, even if the single robot have no intrinsic behavioral intelligence to begin with. From the single mobile robot experiments, the general comparison of FKN algorithm and IFKN algorithm can be seen in Table 2. Based on Table 2, it can be concluded that single robot uses FKN algorithm has the advantage in terms of processing speed. While the single robots using IFKN algorithm has the advantage in terms of pattern recognition accuracy and small resources.

4.2. Cooperative Avoidance Behavior for Swarm Robots

The experimental set-up utilizes 5 sensors from the right side to the left side of the mobile robot to determine their distance to the obstacle. Swarm robots must recognize their environment in order to perform its tasks in the dynamic world [31]. Therefore the environmental recognition problem must be addressed in order to have robust performance of the swarm controller. The problem of identifying the environmental pattern is often hard, due to the sensor readings which are usually uninformative and produce large amounts of noise [30]. In this work IFKN algorithm is utilized in order to recognize the environmental pattern and identify the near by swarm robots. Figure 7(a)-(d) show the starting positions of the 5 smaller robots with different positions of square-shape obstacles. It is observed that the swarm robots have managed to navigate successfully.

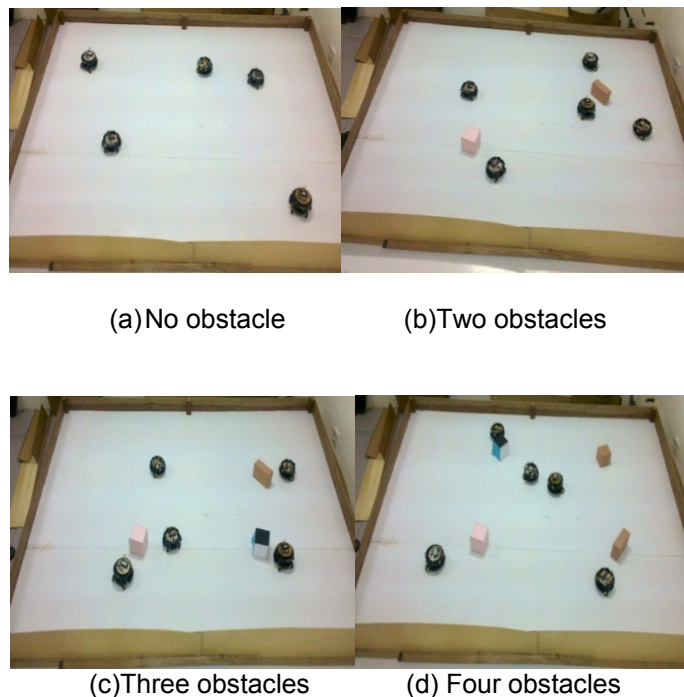
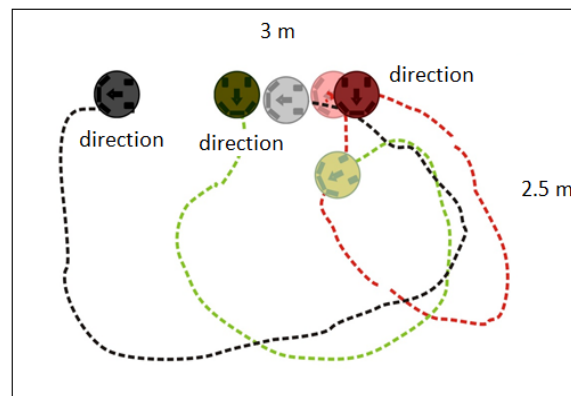


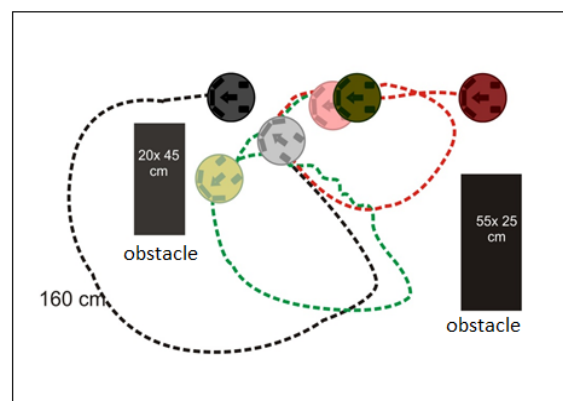
Figure 7. Five (5) swarm robots in real environment

The main difficulty when trying to achieve a goal, collision avoidance and flocking is that prioritizing the sometimes contradictory requirements, therefore this issue is explored. However, collisions between two or more robots must be prohibited. This can be treated as a dynamic situation of the obstacle avoidance problem. The highest priority is given to collision avoidance, because if the collision safety than the robots will reach the goal. Finally, the swarm robots should stay together in a flock while moving.

For flocking behavior structure composed is proposed corresponding which consist of three components: cohesion-separation, velocity matching, and obstacle avoidance behaviors. As demonstrated in Figure 8(a) and (b), two mobile robots approach each other. Once the relative distance is within the threshold, both robots will make decisions to avoid collisions based on the obstacle avoidance use IFKN algorithm addressed above. The real environment situations are tested individually to see how quickly the robots would learn the appropriate behavior. Measurements are taken of robot's performance every 30 ms and produce smooth paths during flocking. The results produce safe paths in the tracking the reference path and avoiding obstacles in the unknown environment. As demonstrated by the graph in Figure 8 (a), the robots became successful in surviving the obstacle they encountered. Figure 8 (b) shows how, over time, by using the IFKN algorithm mobile robot is able to "learn" to make the best actions to overcome any given obstacle. In this work, swarm robots communicate using an X-bee interface.



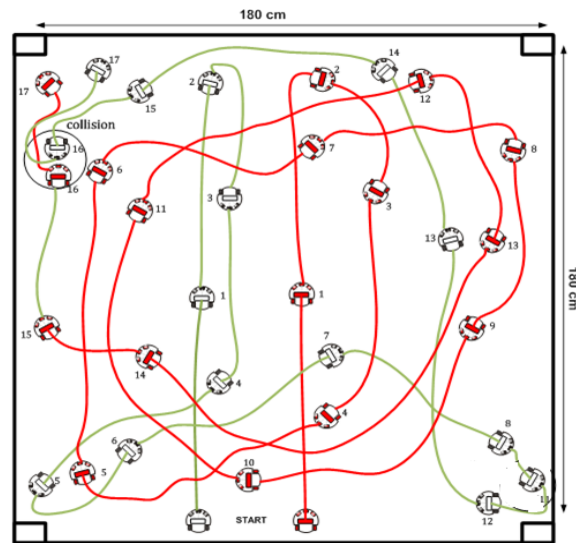
(a) No obstacle



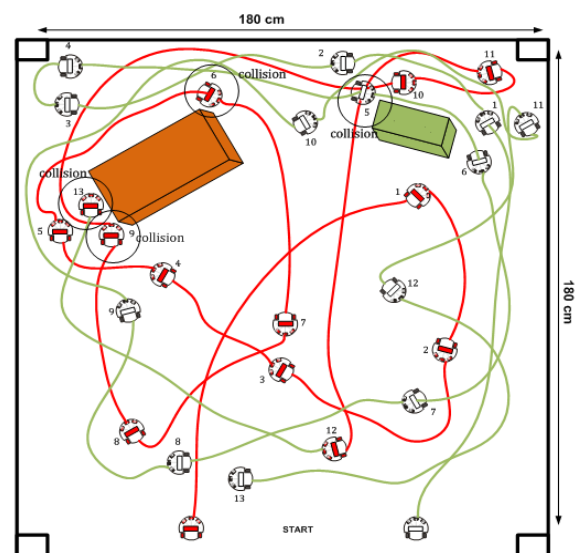
(b) Two obstacles

Figure 8. Cooperative avoidance with flocking behavior

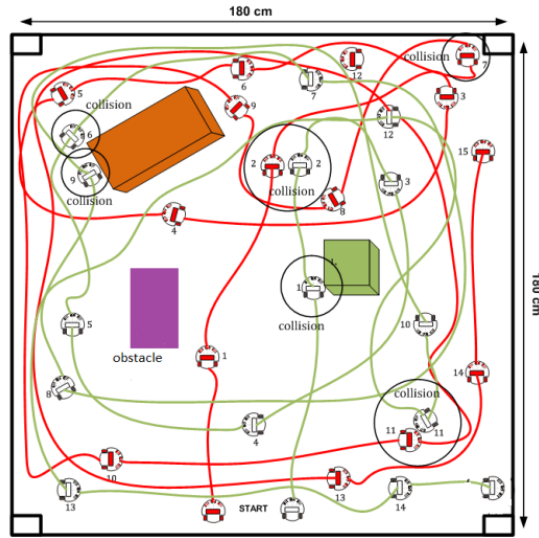
Figure 9(a)-(c) shows the initial random mobile robot movement. IFKN algorithm is used in cooperative obstacle avoidance strategy to navigate the robots around square shape obstacles. The robots' paths are traced by a line with each line representing one iteration's position. The nearest corner of the obstacle is decided based on the calculated Euclidean distance of the pattern's current position relative to each obstacle corner whereby the corner with the smallest value or distance is chosen. The results show, there are a small collisions between the robots, also between the robots and the obstacles. It is clear from Figure 9 (a)–(c), the two robots move through the search space, gaining one new position for every iteration, a conditional statement checks to see if the next position of the robots will fall within the boundaries of the obstacle. When two robots collide with the obstacle, they will not converge to the target.



(a) No obstacle



(b) Two obstacles



(c) Three obstacles

Figure 9. Collision avoidance results with 2 robots

The navigation coordination of the swarm robots is addressed through a complex environment without hitting the obstacles and being trapped in dead-end passageways. This work is motivated by the observation that swarm robots exhibit emergent group behaviors as shown in Figure 10 (a) –(d). IFKN algorithm is used in cooperative obstacle avoidance strategy to navigate swarm robots around square shape obstacles.

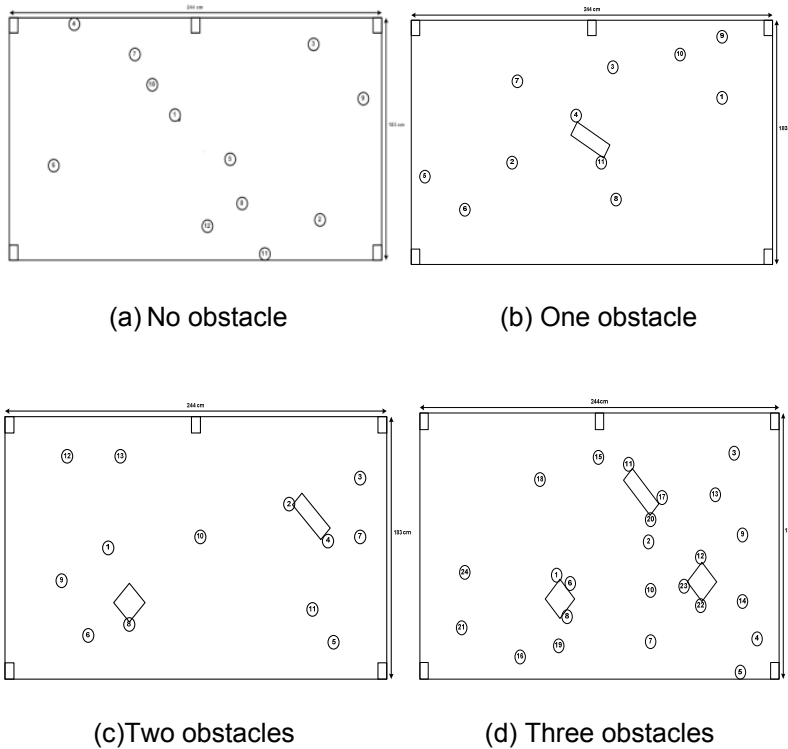


Figure 10. Cooperative avoidance results

The swarm robots' paths are traced by a dotted line with each dot representing one iteration's position. The nearest corner of the obstacle is decided based on the calculated Euclidean distance of the pattern's current position relative to each obstacle corner whereby the corner with the smallest value or distance is chosen. For instance, when two robots face the obstacles, they split into a plurality of smaller groups to avoid collision and then merge into a single group after passing around the obstacle. It is also worth noting that when this group of robots facing a dead end they can even get out of the area. Every moment the swarm robots communicate with their neighbors and all individuals adjust their movement strategy by communicating to exchange position and speed with their neighbors.

5. Conclusion and Future Works

In this paper the proposed IFKN algorithm has been exploited in order to control the cooperative avoidance behaviour of swarm robots formations in presence of obstacles in the environment. Each robot recognize the environment through sensing device and decide the cooperative behavior oneself through communication. The proposed algorithm have the advantages in system robustness, scalability, and individuals simplicity. This strategy can be easily modified in order to allow the robots to successfully consider the risks of the environment and avoid possible obstacles. While still continuing on an efficient trajectory leading towards total swarm convergence to the target. Provided that all the robots are able to keep their desired relative positions to their respective leaders, avoidance of collision is guaranteed. The results show that swarm robots based on proposed IFKN algorithm have the ability to perform cooperative behavior, produces minimum collision and capable to navigate around obstacles. Proposed IFKN algorithm has 86.66% of pattern recognition level compare to FKN only 85.71%. This algorithm produce only 3376 bytes resource of code, due to small pattern are used. In the future, the performance of this swarm robots in both the absence and presence of obstacles are analyzed. Experimental results demonstrate the effectiveness and reliability of the proposed control strategy. There exist various potential future research directions. A major one is real-time experimentation of the proposed control strategy, whose results may lead to some modifications and enhancements in the proposed scheme.

Acknowledgments

Author thank to Higher Education General Director (DIKTI), the Ministry of National Education Department Indonesia and Sriwijaya University under International Research Collaboration for their financial support in Competitive Grants Project. Our earnest gratitude also goes to all researchers in Robotic and Control Laboratory, Computer Engineering, Sriwijaya University who provided companionship and sharing of their knowledge.

References

- [1] Li, W., Shen, W. Swarm behavior control of mobile multi-robots with wireless sensor networks. *Journal of Network and Computer Applications*. 2011; 34:1398–1407.
- [2] Takemura, Y., Sato, M., shii, K. Toward realization of swarm intelligence mobile robots. *International Congress Series*. 2006; 1291: 273–276.
- [3] Bonabeau, E., Dorigo, M., Theraulaz, G. Swarm intelligence: From natural to artificial systems. In Santa Fe Institute Studies in the Sciences of Complexity. New York, NY: Oxford University Press. 1999: 306
- [4] Camazine, S., Deneubourg, J.L., Franks, N.R., Sneyd, J., Theraulaz, G., Bonabeau, E. *Self organization in biological systems*. Princeton, NJ: Princeton University Press. 2003: 560.
- [5] Beni, G. From swarm intelligence to swarm robotics. In E. Sahin & W. Spears (Eds.), *Proceedings of the SAB 2004 Workshop on Swarm Robotics*, Santa Monica, CA, USA, July, 2004. Lecture Notes in Computer Science. 2005; 3342: 1–9.
- [6] Venayagamoorthy, K.G., Grant, L.L., and Doctor, S. Collective Robotic Search Using Hybrid Techniques: Fuzzy Logic and Swarm Intelligence Inspired by Nature. *Engineering Applications of Artificial Intelligence*. 2009; 22: 431-441.
- [7] Sahin, E. Swarm robotics: From sources of inspiration to domains of application, in: E. Sahin, W. Spears (Eds.), *Swarm Robotics Workshop: State-of-the-art Survey*, in: *Lecture Notes in Computer Science*, Springer-Verlag, Berlin Heidelberg. 2005; 3342: 10–20.

- [8] Doctor, S., Venayagomoorthy, G., Gudise, V. Optimal PSO for Collective Robotic Search Applications. *IEEE Congress on Evolutionary Computations*. 2004: 1390–1395.
- [9] Hereford, J., Siebold, M., Nichols, S. Using the particle swarm optimization algorithm for robotic search applications. *Proceedings of Symposium on Swarm Intelligence*. 2007: 53–59
- [10] Mc. Carty I., Manic, M. Adaptive behavioral control of collaborative robots in hazardous environments. *Proceeding of International conference Human System Interactions*. 21-23 may. 2009: 10-15.
- [11] Martinez, S., Cortes, J., Bullo, F. Motion Coordination with Distributed Information. *IEEE Control Systems Magazine*. 2007; 27(4): 75-88.
- [12] Dadios, E., Maravilla Jr., O. *Cooperative mobile robots with obstacle and collision avoidance using fuzzy logic*. In Proceedings of International Symposium on Intelligent Control. 2002: 75–80.
- [13] Ducatelle, F., et al. Mobile Stigmergic Markers for Navigation in a Heterogeneous Robotic Swarm. *Swarm Intelligence*, Springer. 2010: 456-463.
- [14] Konishi, K., Kurabayashi, D., Asama, H., Shin, S. *Analysis of emergence by intelligent data carrier system for collective robots based on stochastic models*. Proceedings of International Conference on Systems, Man, and Cybernetics. 1999; 4: 769–774.
- [15] Jadbabaie, A., Lin, J., Morse, AS. Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules. *IEEE Transactions on Automatic Control*, 2003; 48(6): 988–1001.
- [16] Efrima, E., Peleg, D. Distributed algorithm for partitioning a swarm of autonomous mobile robots. *Theoretical Computer Science*. 2009; 410: 1355–1368.
- [17] Seo, SW., Ko, KE., Yang HC, Sim, KB. Behavior Learning and Evolution of Swarm robotic system using SVM. *Proceedings of International Conference on Control, Automation and Systems*. 2007: 1238-1242.
- [18] Koivo, AJ., Bekey, GA. Report of workshop on coordinated multiple robots: planning, control and application. *Robotics and Automation*. 1998; 14(1): 91-93.
- [19] Zhang, T., Li, X., Zhu, Y., Chen, S., Cheng, Y., Song. J. Multi-Robot Systems, Trends and Development, Published by *InTech*. 2011: 21-32.
- [20] Cai, C., Yang, C., Zhu, Q., Liang, Y. *Collision avoidance in multi-robot systems*. Proceedings of the IEEE International Conference on Mechatronics and Automation. 2007(a): 2795–2800.
- [21] Cai, C., Yang, C., Zhu, Q., Liang, Y. *A Fuzzy-based collision avoidance approach for multi-robot systems*. Proceedings of the International Conference on Robotics and Biomimetics. 2007(b): 1012–1017.
- [22] Koh, WL., Zhou, S. *An Extensible collision avoidance model for realistic self driven autonomous agents*. Proceedings of the IEEE International Symposium on Distributed Simulation and Real-time applications. 2007: 7–14.
- [23] Kettler, A., Szymanski, M., Worn, H. The Wanda Robot and Its Development System for Swarm Algorithm. *Advances in Autonomus Mini Robots*. 2012: 133-146.
- [24] Zhang, N., Wunsch II, D. *Fuzzy Logic in Collective Robotic Search*. Proceeding the 12th IEEE International Conference on Fuzzy Systems. 2003(a); 2: 1471–1475
- [25] Zhang, N., Wunsch II, D. *Acomparision of Dual Heuristic Programming (DHP) and Neural Network Based Stochiastic Optimization Approach on Collective Robotic Search Problem*. Proceedings of International Joint Conference on Neural Networks. 2003(b); 1: 248–253.
- [26] Ohkura, K., Yasuda, T., Matsumura, Y. Coordinating the adaptive behavior for swarm robotic systems by using topology and weight evolving artificial neural networks. *IEEE Congress on Evolutionary Computation*. 2010: 1-8.
- [27] M. Clerc, J. Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*. 2002; 6(1): 58-73.
- [28] Chyan, SG., Ponnambalam. Obstacle avoidance control of redundant robots using variants of particle swarm optimization. *Robotics and Computer-Integrated Manufacturing*. 2012; 28: 147–153.
- [29] Liu, W., Winfield, A., Sa, J. *Modelling swarm robotic systems: a study in collective foraging*. Proceedings of Towards Autonomous Robotic Systems. 2007: 25–32.
- [30] Nurmaini, S. Memory-based Reasoning Algorithm Based on Fuzzy-Kohonen Self Organizing Map for Embedded Mobile Robot Navigation. *International Journal of Control and Automation*. 2012; 5(3): 47-64.
- [31] Nurmaini, S., Aditya PP., Rendyansah. Intelligent Navigation in Unstructured Environment by using Memory-based Reasoning in Embedded Mobile Robot. *European Journal of Scientific Research*. 2012; 72(2): 228-244.
- [32] Yamada, S. Recognizing environments from action sequences using self-organizing maps. *Applied Soft Computing*. 2004; 4: 35–47.