Polygon Shape Formation for Multi-Mobile Robots in a Global Knowledge Environment

Abdulmuttalib T. Rashid
Electrical Engineering Depart.
Basrah University
Basrah, Iraq
abdturky@gmail.com

Abduladhem A. Ali
Computer Engineering Depart.
Basrah University
Basrah, Iraq
abduladem1@yahoo.com

Mattia Frasca
DIEEI
University of Catania
Catania, Italy
mfrasca@diees.unict.it

Abstract *In coordination of a group of mobile robots in a real environment, the formation is an important task. Multi-mobile robot formations in global knowledge environments are achieved using small robots with small hardware capabilities. To perform formation, localization, orientation, path planning and obstacle and collision avoidance should be accomplished. Finally, several static and dynamic strategies for polygon shape formation are implemented. For these formations minimizing the energy spent by the robots or the time for achieving the task, have been investigated. These strategies have better efficiency in completing the formation, since they use the cluster matching algorithm instead of the triangulation algorithm.*

***Index Terms*—** **Polygon shape formation, Global knowledge environment, multi-mobile robot.**

I. INTRODUCTION

An autonomous robot can be considered as an intelligent agent, which plans its actions according to the information obtained from sensors equipped on it. It moves in unknown environment without human assistance in order to reach its goal, while avoiding the collision with obstacles that may appear along its path [1, 2]. In the recent years, the use of multi-robot systems rather than just single robots for performing several complex tasks has raised growing interest. Multi mobile robots are used in several scenarios including: searching of an environment with unknown location [3], localization [4-6], object transportation [7], tasks constructions [8], robotics configuration [9], and motion coordination [10]. All of the above require multi robots system with some kind of group coordination and cooperation mechanisms.

The main goal of this paper is the design and the construction of a team of multi-robots and the algorithms for their formation and guidance in a complex environment with static obstacles. Robots formation aims at move each robot from current location to a target while obstacles need to be avoided. The robots formation is achieved when the relative distance between each element is kept constant at all times.

The process of multi-mobile robots formation can be performed either with centralized or decentralized approaches. The centralized approach is based on the use of some robots as supervisors of the trajectories of the other robots. A virtual reference on the desired trajectory controlled by remote supervisor is used to maintain the predefined positions of individual robots [11]. In centralized approaches a heavy computation and tight communication with other robots is required for the supervising robot. In contrast, the formation pattern in decentralized approaches is achieved through the individual robot's coordination decisions. One of decentralized approaches is achieved by using an algorithm with an unlimited amount of memory to accomplish regular polygonal shapes [12]. It depends on their past experience. On other hand, this algorithm is modified to an oblivious algorithm and used to circle formation [13].

Several approaches have been used to solve the formation control task. Among these approaches, the most common methods used for formation control are the behavior-based virtual structure and the leader-following. The effective formation control in the behavior-based approach results from a weighted summation of each behavioral output [14]. There is a clear difficulty to describe the group behavior and to evaluate the stability of the whole system. The formation in the virtual structure approach is achieved by assigning the desired position for every robot to the virtual structure that tracks the trajectories in the environment [15]. The stability of this technique is guaranteed and it is more robust, but it is difficult to control multiple mobile robot formation in a decentralized manner. In the leader- follower approach, each follower robot is controlled to track one leader with (l -Ψ) controller, or two leaders with (l -l) controller [16]. This method has a simple structure and is easy to implement by two controllers. It only relies on the information from local sensors. The drawback of this method is that the leader robot is a single unit whose fault may lead to failure of the whole formation. Other approaches to formation are also available like artificial potential field, graph theory, and synchronization approach. In the artificial potential field an artificial force is generated to drive robots and form definite formations. The use of the artificial potential field was improved to generate queues [17]. Robot queues formed by robots can be combined to generate some formations like columns, wedges and double columns. Many researchers used graph theory to deal with formation control among robots. The formation is defined by basic concepts in graph theory, and the representation of vertex-edge relationship is introduced to analyze the linear dynamics of a group of vehicles [18]. Two distinct properties: graph connectivity and graph rigidity in graph theory were used to stabilize the dynamics of a multi-vehicle system [19]. The control goal in synchronization method is derived according to the desired formation, which is based on the differential position errors between each pair of the adjacent robots. One of the synchronization methods is the cross-coupling control [20]. In this method, the motion control for each robot is divided into two parts: the first part is to realize the tracking control goal by moving each robot along the desired path. The second part consists of investigating the motion synchronization among two nearly robots. In this manner, the synchronization is achieved between all robots in the group. The synchronization approach can be designed as a decentralized scalable system with low complexity and computational power. However, this method still needs further research to perform formation for different types of mobile robot's dynamics. In other works, each robot will track its desired trajectory in the shape, while the center of the formation shape will move in a straight line, and instantaneously synchronizing its motion with the two nearby robots maintain the wanted time-varying formation [21]. This work differs from [22], where the formation shape center is fixed and robots only switch between different formations shapes in a time-varying formation

The main aim of this paper is to perform the polygon shape formation of multi-robot in global knowledge environments. Several static and dynamic strategies for polygon shape formation are implemented. This paper is focused on produces strategies which guarantee simplicity in system design; aim at saving the energy spent through the formation construction and reduces the time to complete the formation.

II. Polygon Shapes Formation Strategies

In this paper, different strategies of polygon shape formation generation are performed by considering the effect of the connectivity between robots (global knowledge environment), and the type of formation to investigate (static and dynamic formations). In global knowledge environment, each robot has unlimited sensing of the obstacles, other mobile robots and target position in environment. The static formation means that the goal position is fixed in time while the dynamic formation occurs when the goals change their positions with time.

A. Static polygon shape formation

In this strategy a static polygon formation is achieved by assigning a specific goal to every robot as shown in Fig. 1.

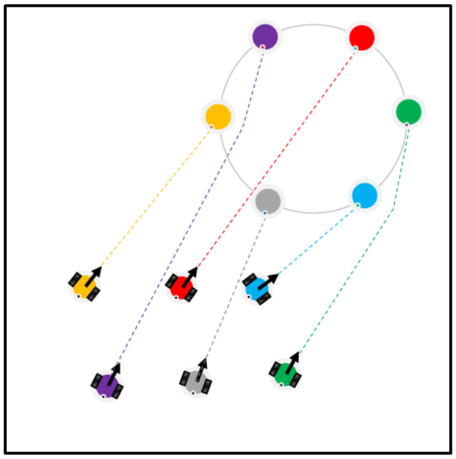

Fig. 1. The static polygon formation strategy.

The cluster matching algorithm is used to calculate the initial location and orientation of each robot, which is assumed to be known by other robots [23]. The formation is achieved by driving every robot to its own goal through a straight line trajectory. The path planning with obstacle avoidance is achieved by using the visibility binary tree algorithm. However, in this strategy, these trajectories may cause collision among robots and then break the whole system. The robots use the reciprocal orientation algorithm to help them plan trajectory to their goals [24]. The implementation process of this strategy is performed according to the following steps:

Step 1: Compute the position (x_i, y_i) of each robot by using the cluster matching algorithm. Fig. 2. shows the steps required to implement this algorithm. Fig. 2 (a) shows the initial position and orientation of each robot in the environment. Fig. 2(b) shows the estimation process of the location of visible robots by using a static anchor node at one corner of the environment. Fig. 2(c) shows the process of partition of robots into clusters. Each cluster contains one cluster head and some of neighbor's nodes. Fig. 2(d) shows the process of representing the robots into a unit disk graph. By matching this unit disk graph structure with cluster head structure, the localization information calculated in Fig. 2(b) is assigned to the correct robot.

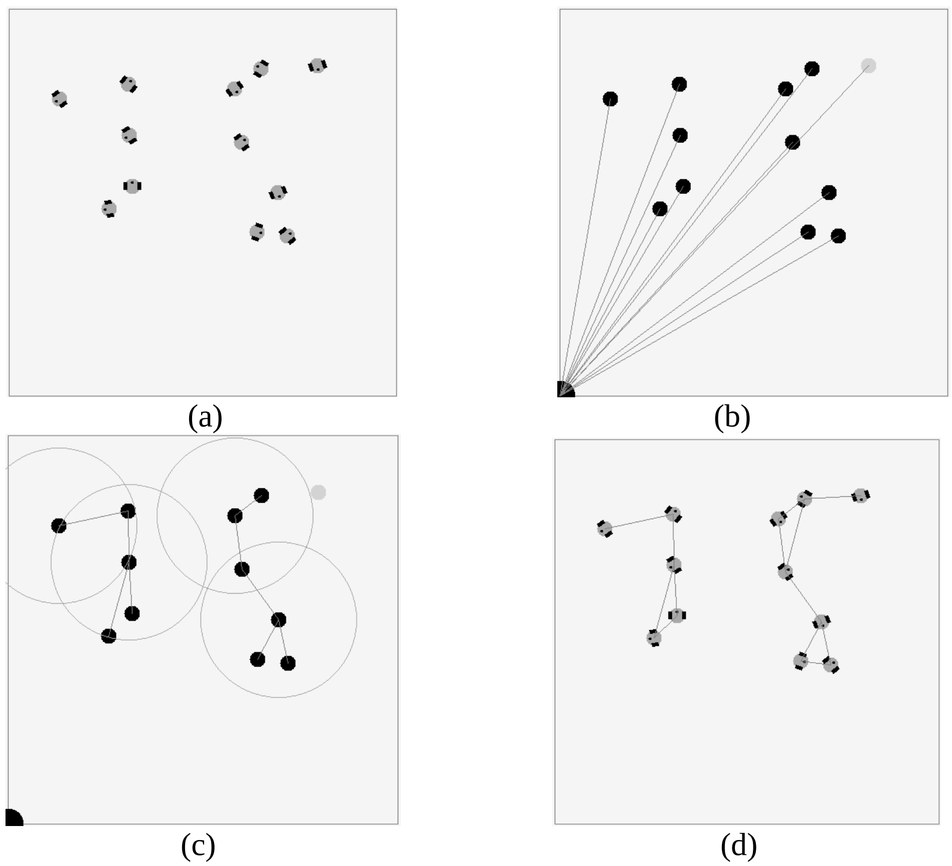

Fig. 2. The cluster matching algorithm.

Step 2: The shortest path with obstacle avoidance of each robot to its goal is computed by using binary tree tangent graph algorithm [25]. Fig. 3 shows the steps of implementation of this algorithm. Fig. 3(a) shows the direct path from green robot to its goal. Since this path intersects with gray robot, so that the binary tree tangent graph algorithm calculates new paths as a binary tree structure from robot location to goal (red dot lines) as shown in Fig. 3(b). This binary tree structure is optimized by solid red path as shown in Fig. 3(c). Finally, the searching algorithm is used to compute the shortest path from robot location to the goal (solid black line) as shown in Fig. 3(d).

Step 3: At each interval, each robot must check positions and orientations of all robots in environment. This process is done by using the reciprocal orientation algorithm [24]. The main idea of this algorithm is that the robots take into account the observed direction of the other robots in order to avoid collisions with them.

Moreover, each robot selects its own orientation taking into account the physical limits of the robot actuators. It also allows avoiding a collision between two robots by predicting the trajectories of the two robots in the near future and thus calculating if the two robots will collide or not. The term reciprocal refers to the fact that each robot takes half of the responsibility of the proper action to avoid the collisions, since it relies on the assumption that the other robot will also make some corrections to its trajectory. Fig. 4 shows the steps of implementation of this algorithm.

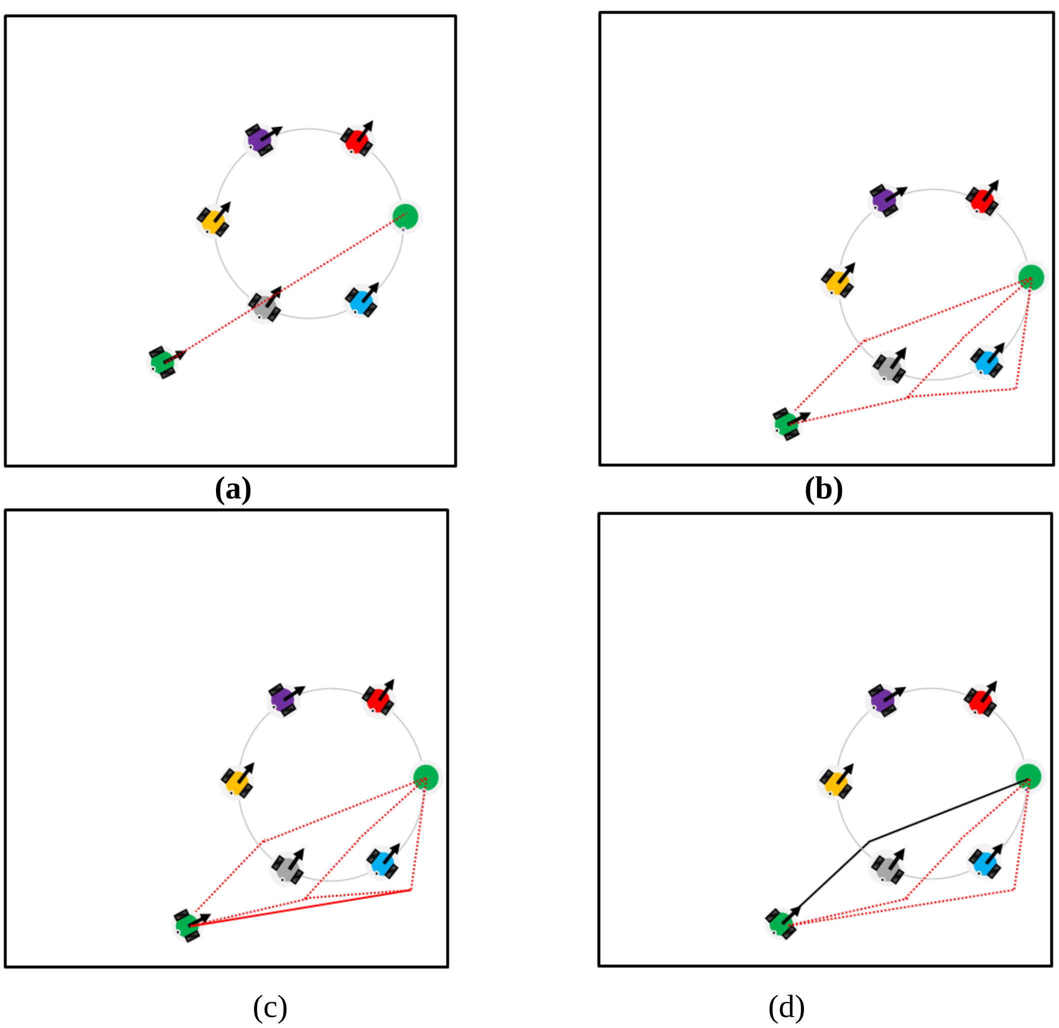

Fig. 3. The binary tree tangent graph algorithm.

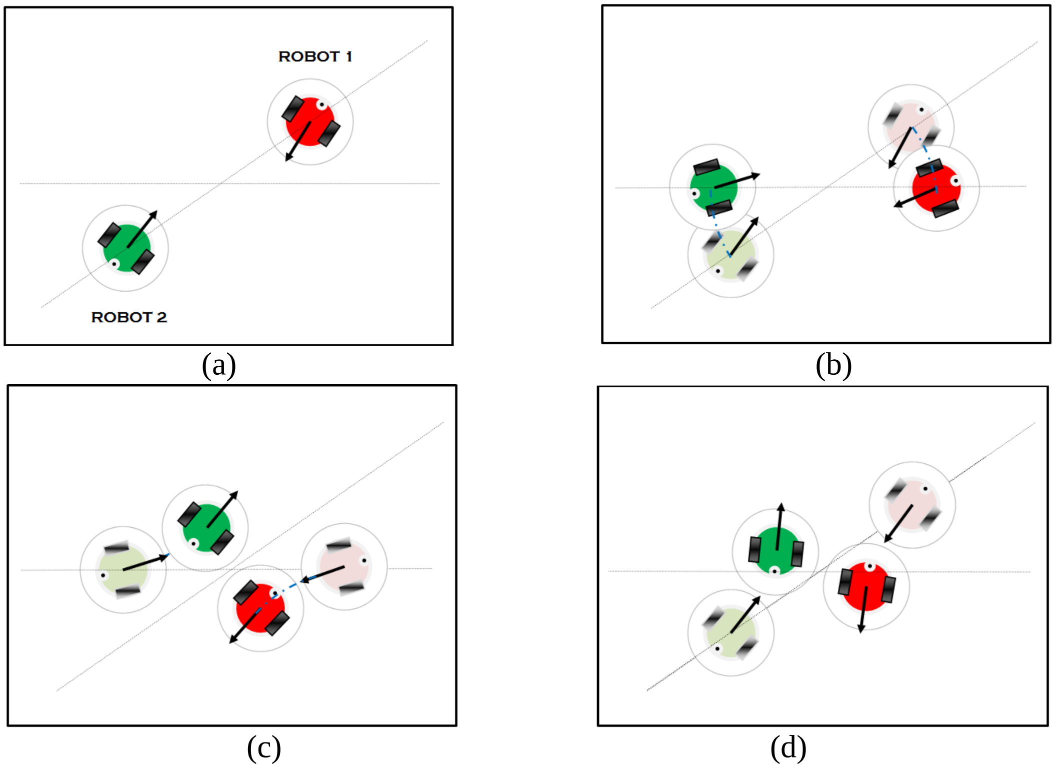

Fig. 4. The reciprocal orientation algorithm.

Fig. 4(a) shows a dynamic environment with two moving robots. Each robot considers the observed directions of the other robots in order to establish the trajectory to follow so that collisions are avoided. The first step of this algorithm is to align the straight line between these robots with horizontal axis. According to this, virtual locations of the robots are found by rotating the actual positions of the two robots to the horizontal direction as shown in Fig. 4(b). After that each robot predicts its collision avoidance trajectory in the near future and thus calculating if the two robots will collide or not. Fig. 4(c) shows the next virtual position and orientation in environment. The last step represents by returning the robots to their original location by rotating them in reverse direction as shown in Fig. 4(d).
The overall approach is summarized by the pseudo-code shown in Fig. 5.

B. Dynamic goal polygon shape formation

In this strategy, each robot must decide its own goal location dynamically in global knowledge environment. Many methodologies such as tabu search, variable depth search, and combination of greedy methods are developed to solve such problem in a centralized way [26]. Two cases for choosing the goal are discussed in this strategy. The first one is to save energy by making each robot to go to the nearest goal from its current location. In this case, two robots may decide to go to the same goal location during their motion. Hence, certain competition mechanism among these robots should be designed to prevent two or more robots approaching the same goal. The second one is to decrease the time of formation by rearranging the goals of the robots in a manner making the longest time of arrival as small as possible. In both cases the decision of arranging the goals is based on building an array contains information about distances between each robot and all goals. This strategy is achieved by searching this array and using the permutation algorithm to find the best arrangement between goals and robots.

1. Saving energy case

This case is implemented in the global knowledge environment, where each robot knows its location including the distances to the other robots. By using permutation algorithm, all the probabilities of distributions of the robots on goals can be tested to choose the best arrangement for this case. The permutation algorithm is applied on information stored in the two dimensional array containing the distances between robots and goals. The best arrangement occurs in the case when the sum of the distances between robots and goals is the smallest. The steps of implementation of this algorithm are:

Input *n*: number of robots
R: Maximum detection range of infrared sensors
$P(x_s, y_s)$: Laser sensor position.
g1(x_1, y_1).. gn(x_n, y_n): Goals positions from 1 to n.
For each interval *t* **do**
For each robot *i* **do**
Use cluster matching algorithm to estimate robot position (x_i, y_i) and orientation(θ_i).
Assignment robot *i* to its corresponding goal *i*.
Compute robot *i* trajectory to its goal using binary tree tangent graph algorithm.
Estimate the future position of robot *i* using reciprocal
Orientation algorithm to avoid collision with other robots.
Next robot *i*
Next time interval t

Fig. 5. Static polygon formation strategies.

Step 1: Compute the position (x_i, y_i) of each robot by using the cluster matching algorithm as shown in Fig. 6(a).
Step 2: The shortest path from each robot to all goals is computed by using binary tree tangent graph algorithm. The distances d_{ij} between robot i and goal j as shown in Fig. 6(b) is

$$d_{ij} = \left(\left(y_i - y_j\right)^2 + \left(x_i - x_j\right)^2\right)^{1/2} \quad (1)$$

where (x_i, y_i) position of robot i, and (x_j, y_j) position of goal j.
Step3: Store these distances in two dimensional matrix *d*, where rows of this matrix are the robot numbers and columns are the goal numbers.

$$d=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & & d_{2n} \\ \vdots & & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{n} \end{bmatrix} \quad (2)$$

Step 4: scan all values of permutation algorithm, where each value represents the assignment of one robot to one goal. At each scan value compute the sum of all distances between robots and their goals. Now the lowest value obtained in permutation scanning corresponds to the best case in terms of energy saving as shown in Fig. 6(c).
Step 5: At each interval, each robot must check positions and orientations of all robots in environment to avoid the collision with other robots. This process is done by using the reciprocal orientation algorithm.

2. Shortest time of formation case

This case is implemented in global knowledge environment. Also, by using permutation algorithm, all the probabilities of distributions of the robots on goals can be test to choose the best arrangement. The best arrangement in this case occurs when the maximum distance in this arrangement is the minimum with respect to the other arrangements. The steps of implementation of this strategy are:
Step 1: Perform steps one, two, and three of the saving energy case.
Step 2: Scan all values of permutation algorithm, where each value represents the assignment of one robot to one goal. At each scan value compute the maximum of all distances between each robot and its goals.
Step 3: Choose the minimum value of all the maximum values of all the possible arrangements to represent the case of the shortest time of formation, as shown in Fig. 6(d).
Step 4: At each interval, each robot uses the reciprocal orientation algorithm to check the positions and orientations of all robots to avoid the collision with them.

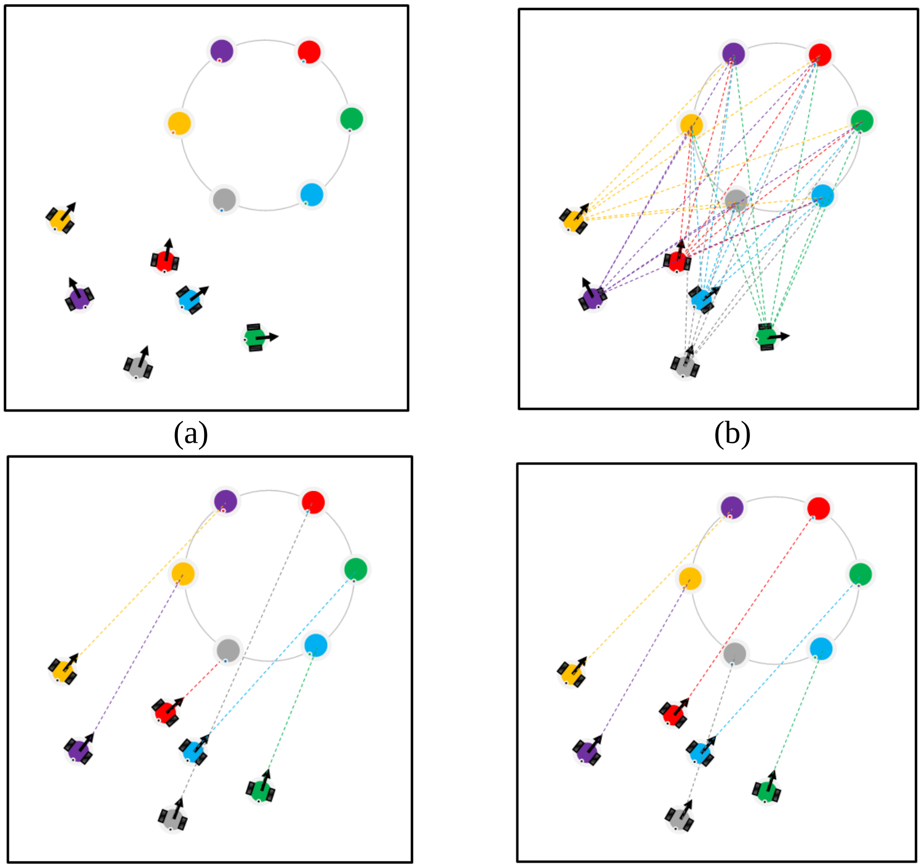

Fig. 6. Dynamic goal formation strategies.

C. Dynamic polygon shape formation

In this strategy, a group of robots moves in global knowledge environment to form a dynamic polygon. Each robot in polygon formation follows a virtual goal. All virtual goals have relative positions to one leader virtual point, which may be a moving point along a regular trajectory, or a fixed point in the environment, as shown in Fig. 7. This fixed leader point is a center of circular paths, and the virtual goals have relative positions to this point. At each time interval the virtual goals change their angle by a fixed increment angle α. The steps of implementation of this strategy are as follows:

Step 1: Compute the initial position (x_i, y_i) of each robot by using the cluster matching algorithm.

Step 2: By using the virtual leader location *(Lx, Ly)*, radius of circular paths (r_1, r_2), and new increment angle α, compute the virtual goal locations of four robots. The virtual goal location of robot i (Gx_i, Gy_i) in the inner circle path is

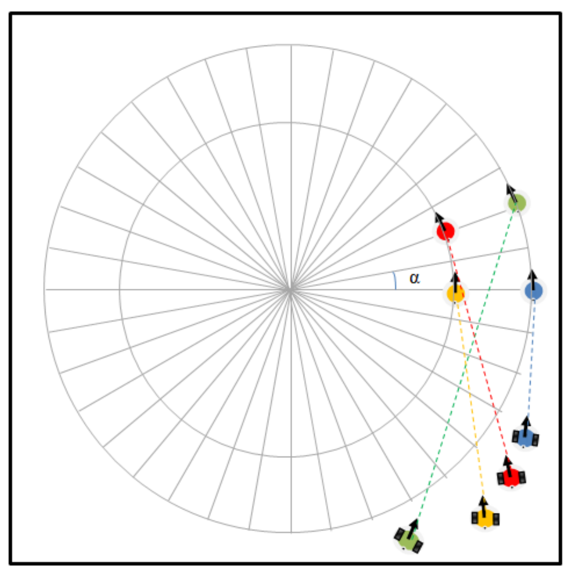

Fig. 7. Circular path with virtual goal points.

$$Gx_i = Lx + r_1 * \cos\alpha$$
$$Gy_i = Ly + r_1 * \sin\alpha \qquad (3)$$

Step 3: The shortest path from each robot to its virtual goal is computed by using binary tree tangent graph algorithm.
Step 4: Each robot uses the reciprocal orientation algorithm to check the positions and orientations of all robots to avoid the collision with them.
Step 5: At each interval, increase angle α and repeat steps 2, 3, and 4 until the formation is completed as shown in Fig. 8.

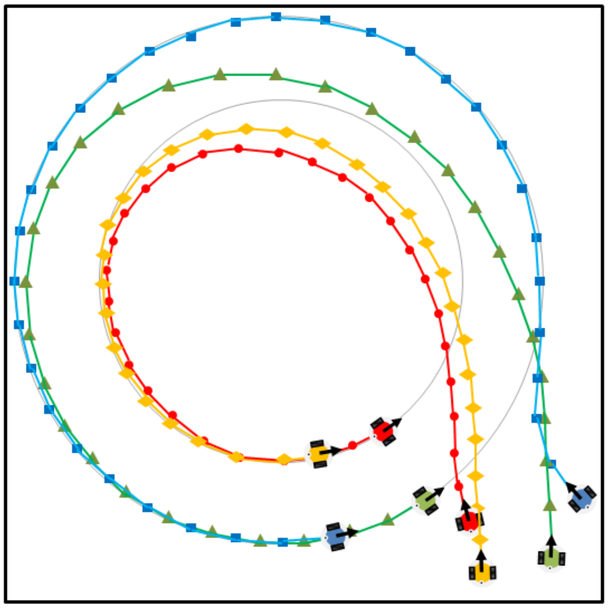
Fig. 8. Dynamic polygon formation in a circular path.

III. Simulation Results

In this paper the algorithms which proposed to investigate the formation of multi mobile robots are simulated and the results of these simulations are discussed. The different strategies of formation generation are simulated by considering the effect of the connectivity between robots (global knowledge environments), and the type of formation to investigate (static and dynamic formations). There were implemented using visual basic 2010. All simulations were performed over different topologies representing different network sizes (n) ranging from 4 to 8 robots. The robots were randomly placed on a 500x500 pixels area. There are two parameters used in this simulation:

- Network size (n): the number of robots in the network with a square area of side length l.
- Maximum detection range of infrared sensors (R): This parameter constraints the maximum graph distance between any two neighbor robots.

The purpose of these simulations is to evaluate the performance of these polygon formation strategies. The following performance metrics are used:

1. Percentage of accomplishment: this metric denotes the time consumed by multi-robots from their initial position to the final position. The time of accomplishment can be used to characterize the formation by using different networking topologies.
2. System Efficiency: The system efficiency [25] is defined as follows

$$\text{System efficiency} = \frac{1}{n}\sum_{i=1}^{n} \frac{|x_{R_i}(t) - x_{R_i}(0)|}{l_{R_i}(t)} \qquad (4)$$

where n is the number of robots, and $|X_{R_i}(t) - X_{R_i}(0)|$ is the straight-line distance of a robot from the initial position to the position at time t, and $l_{R_i}(t)$ is the total travelling distance of each robot at time t. By using this factor, the polygon formation of multi-mobile robots can be compared with different networking topologies.

Fig. 9 shows the first experiment simulation. Fig. 9 (a)–(e) represent the Screenshots of the simulation at different time steps. Fig. 9.f represents the trajectories of the robots to investigate the formation in global knowledge environment, where each robot has unlimited sensing of the obstacles, other mobile robots and target position. The goal positions in this experiment area assumed to be static.

The main goal of this simulation is to show that with a careful selection of the input parameter n, the proposed strategy is satisfactory. From Fig. 10, it is clear that increasing the number of robots n increases the accomplishment percentage with respect to time units, where time units of the simulator is the atomic unit of robot movement. That is the robot can make one movement decision by distribution and receiving information, in one time unit.

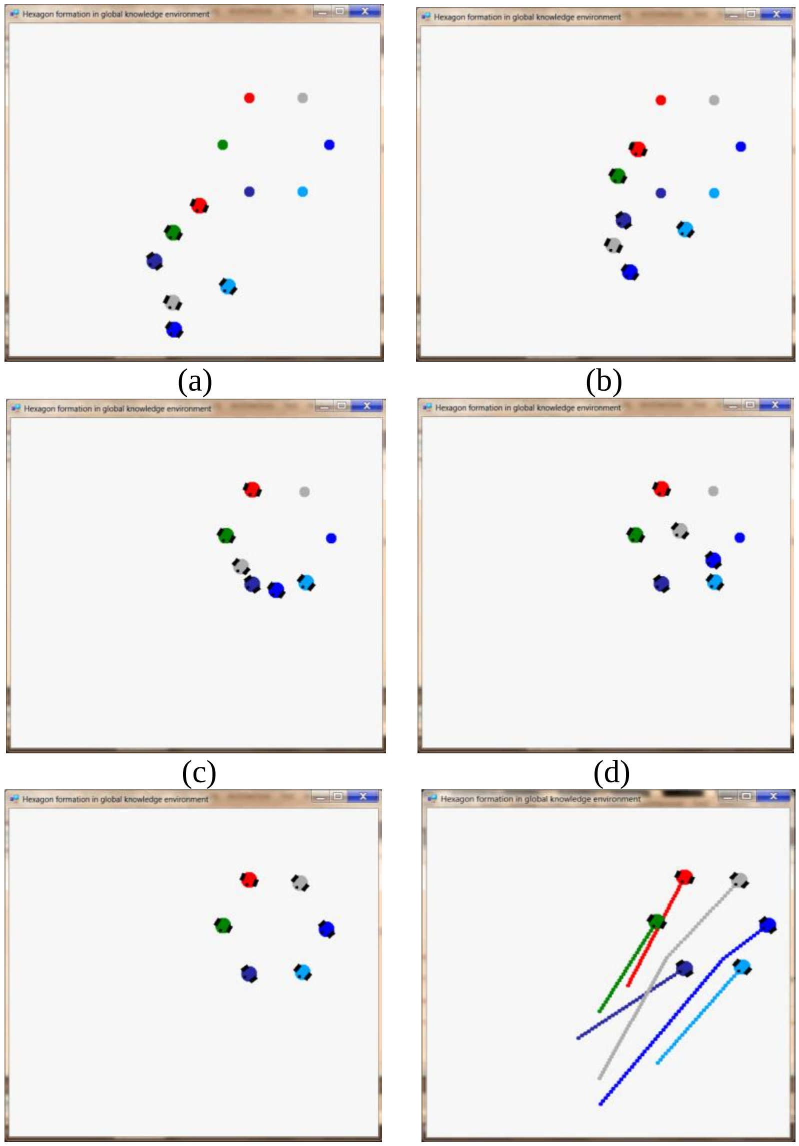

Fig. 9. Static polygon formation with global knowledge environment. (a)-(e) Screen shots of the simulations in different time steps. (f). Trajectories of the robots in the formation.

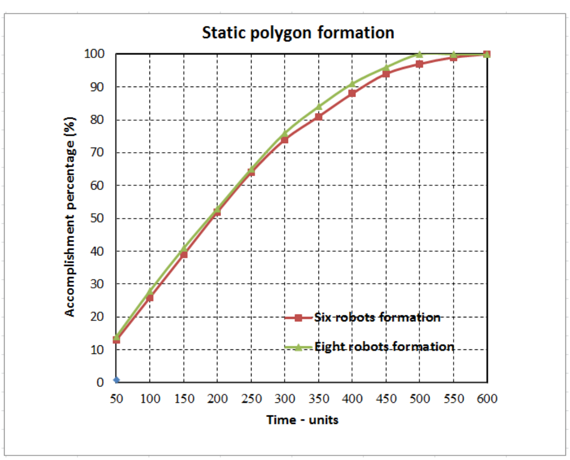

Fig. 10. Formation with different robot numbers.

Fig. 11 shows the second simulation. Fig. 10(a)–(e) represent the Screenshots of simulations at different time steps. Fig. 11.f represents the trajectories of the robots in global knowledge environment. The goal positions in this scenario are assumed to be dynamic with time. Two strategies are tested in this experiment. The first one is the saving energy formation, where each robot chooses its goal dynamically. The second strategy is the shortest time formation.

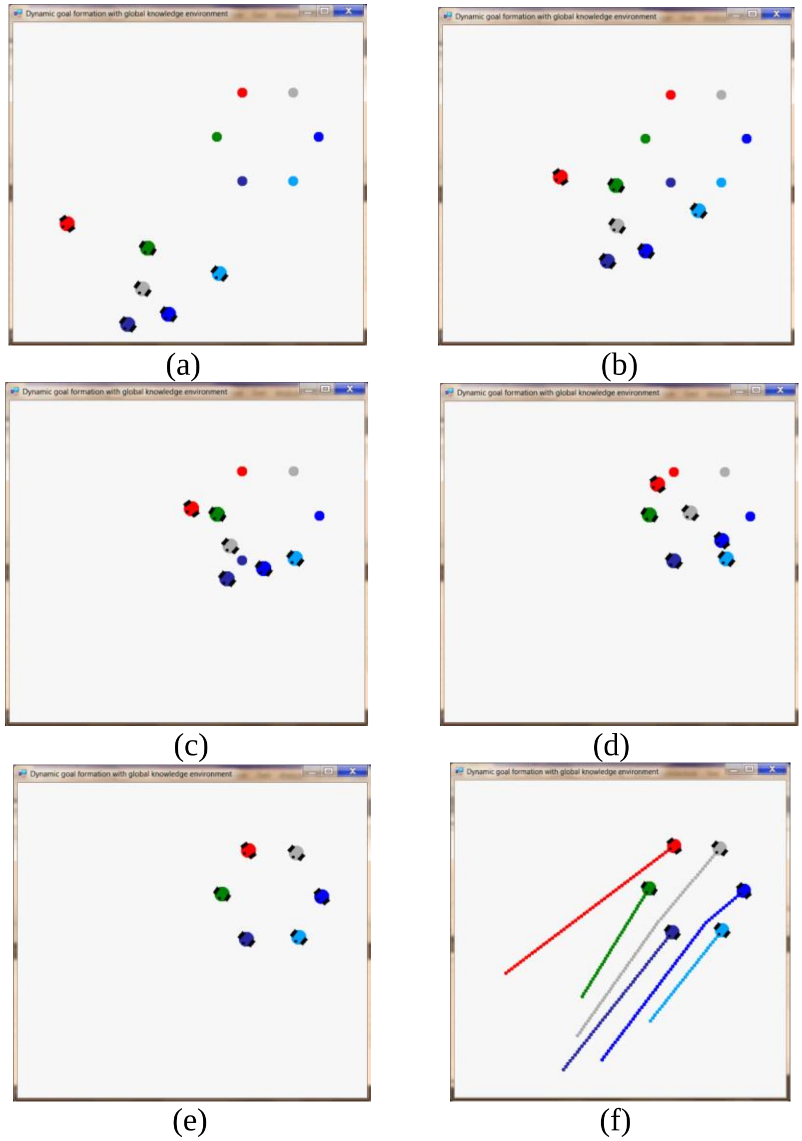

Fig. 11. Dynamic goal polygon formation with global knowledge environment. (a)-(e) Screenshots of the simulations at different time steps. (f) Trajectories of the robots.

The main goal of this simulation is to show that with a careful selection of input parameter n, and the desired goal, the proposed strategy is satisfactory. From Fig. 12, it is clear that in saving energy formation decreasing the number of robots n and increases the accomplishment percentage.

Fig. 13 shows the comparison between static polygon formation and two cases of dynamic polygon formation. The accomplishment percentage is equal for all cases since in these strategies all robots start from the same locations and have the same goals.

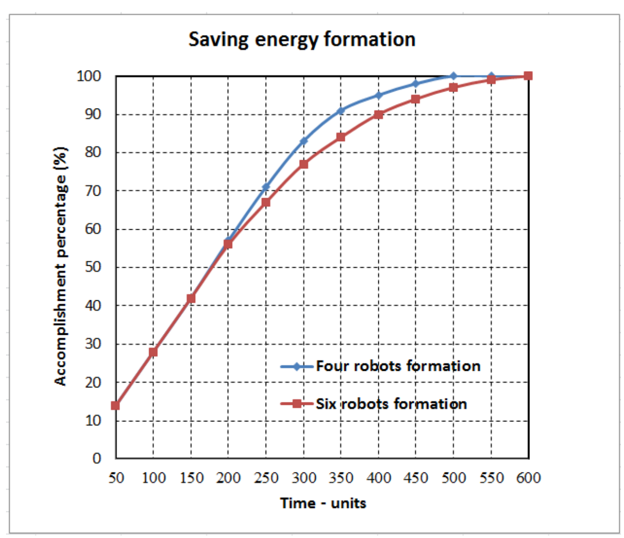

Fig. 12. Dynamic goal formation with different number of robots.

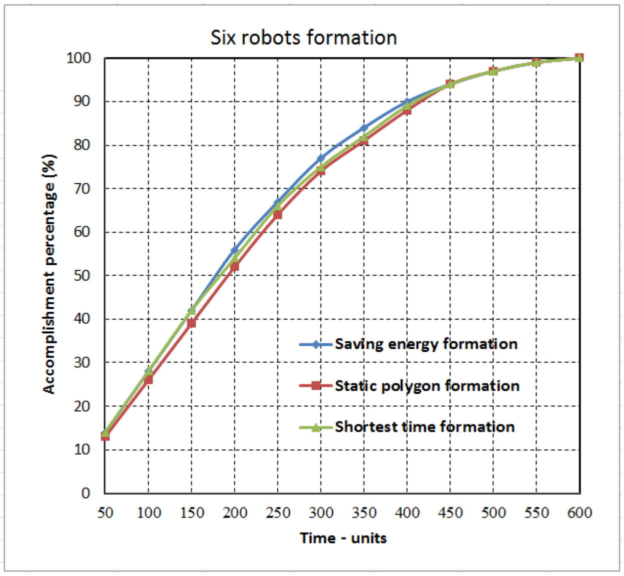

Fig. 13. Comparison between static polygon and dynamic goal formations.

The system efficiency of static polygon simulation is 97%, and for save energy dynamic goal simulation is 88.5%, and for shortest time simulation is 98%. The shortest time dynamic goal strategy has better efficiency, since it is the first strategy that completes the formation.

Fig.14 shows the third simulation. Fig. 14(a)–(e) represent the Screenshots of simulations at different time steps. Fig. 14.f represents the trajectories of the robots. Each robot follows a virtual movement goal. All virtual goals move along a regular trajectory. Fig. 15 shows accomplishment percentage for dynamic polygon formation of four mobile robots. The system efficiency of dynamic polygon simulation is 88.5%. The accomplishment time depends on the number of robots n, the initial position of these robots and the shape of the polygon to be formed.

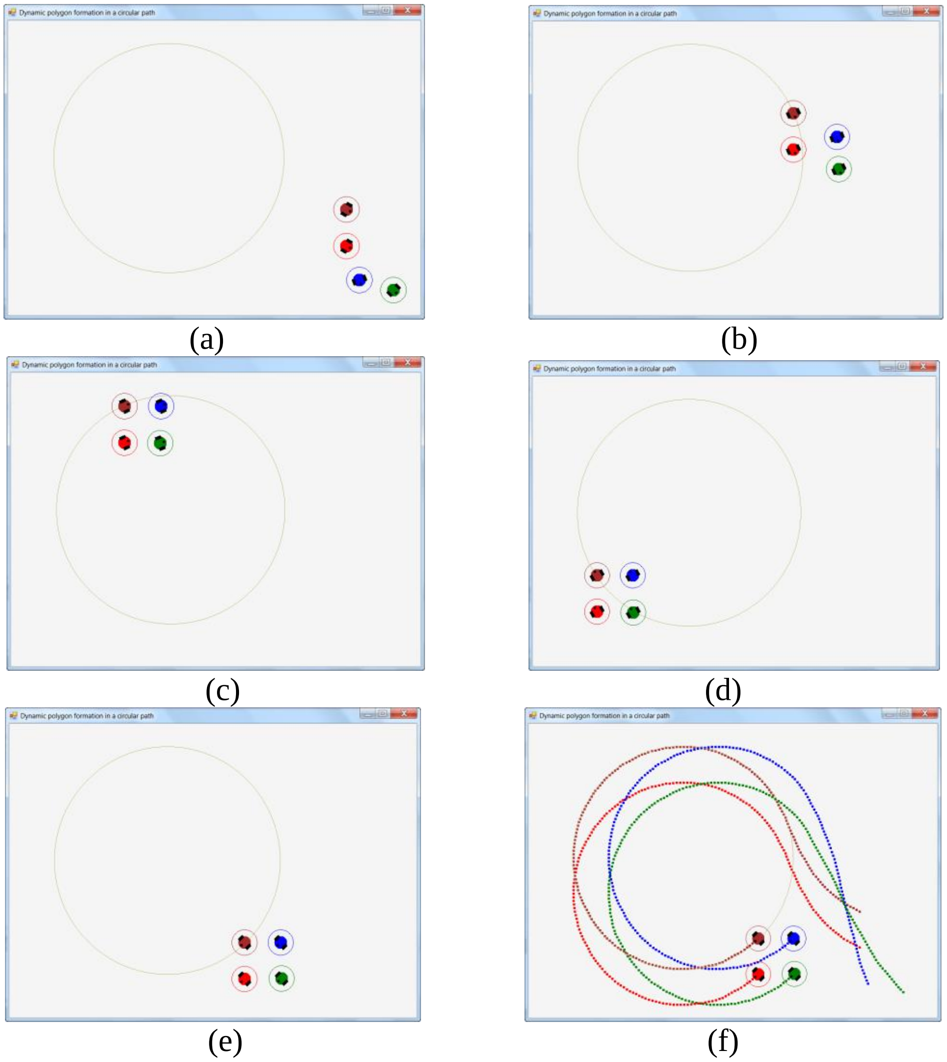
(a) (b) (c) (d) (e) (f)

Fig. 14. Dynamic polygon formation with global knowledge environment.
(a)- (e) Screenshots of simulations in different time steps. (f). Trajectories of the robots in the formation.

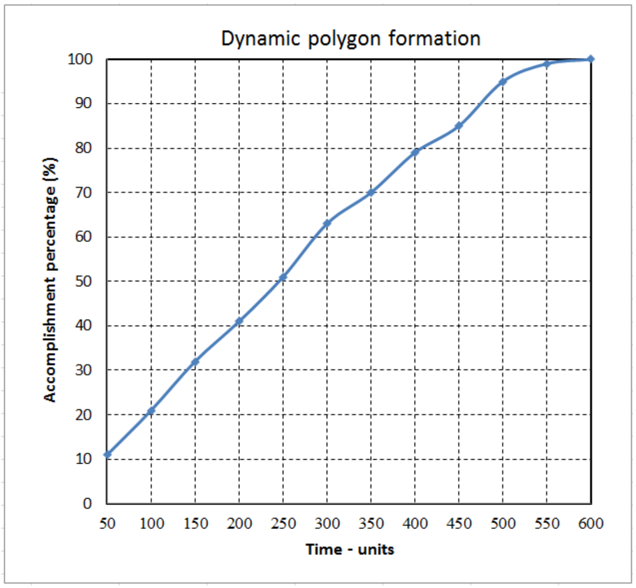

Fig. 15. Accomplishment percentage in dynamic polygon formation.

IV. Conclusions

Simulation results show that increasing the number of robots leads to an enhanced accomplishment time in static polygon formation. The comparison between static polygon formation and dynamic goal formation indicates that the accomplishment time depends on the initial position of robots, and also, the shortest time dynamic goal strategy has better efficiency, since it is the first strategy that completes the formation. The dynamic polygon formation is achieved with four robots move in circular path. The accomplishment time for this strategy depends on the shape of the polygon to be built, the number of robots and the initial position for these robots. Finally, the results show that the modified self-localization strategy has a better efficiency to complete the formation, since it uses the cluster matching algorithm instead of the triangulation algorithm to estimate the locations of robot.

V. References

[1] Z. Y. Ibrahim , A. T. Rashid, and A. F. Marhoon, " An algorithm for Path planning with polygon obstacles avoidance based on the virtual circle tangents", Iraq J. Electrical and Electronic Engineering, Vol. 12, No. 2, pp. 221-234 , 2016.

[2] Z. Y. Ibrahim , A. T. Rashid, and A. F. Marhoon, " Prediction-Based Path Planning with Obstacle Avoidance in Dynamic Target Environment ", Basrah Journal for Engineering Sciences, Vol. 16, No. 2, pp. 48 – 60, 2017.

[3] W. Burgard, M. Moorsy, C. Stachniss, and F. Schneider," Coordinated multi-robot exploration", IEEE transactions on robotics, vol.21, no.3, pp. 376-386, 2005.

[4] O. A. Hasan, A. T. Rashid, R. S. Ali, "Hybrid approach for multi-node localization and Identification", Basrah Journal for Engineering Sciences, vol.16, No 2, pp. 11-20, 2016.

[5] O. A. Hasan, R. S. Ali, A. T. Rashid, "Centralized approach for multi-node localization and identification", Iraq J. Electrical and Electronic Engineering, vol.12, No 2, pp. 178-187, 2016.

[6] A. T. Rashid, W. H. Zayer and M. T. Rashid, "Design and Implementation of Locations Matching Algorithm for Multi-Object Recognition and Localization "Iraq J. Electrical and Electronic Engineering, Vol.14 No.1, pp. 10-21, 20 1 8.

[7] R. Soares, E. Bicho, T. Machado and W. Erlhagen," Object transportation by multiple mobile robots controlled by attractor dynamics: theory and implementation", IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, pp. 937-944, 2007.

[8] Q. Lindsey, D. Mellinger and V. K. Lindsey," Construction of cubic structures with quadrotor teams", Proceedings of Robotics: Science and Systems, Los Angeles, USA, 2011.

[9] M. Yim, D. G. Duff, and K. D. Roufas, "Polybot: a modular reconfigurable robot", Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA , pp. 514-520, 2000.

[10] A. Yamashita, J. Ota, T. Arai, and H. Asama," Motion planning of Multiple Mobile Robots for Cooperative Manipulation and Transportation", Proceedings of the IEEE Transactions on Robotics and Automation, vol. 19, no. 2, pp. 1-15, 2003.

[11] M. Egerstedt and X. Hu, "Formation constrained multi-agent control", IEEE

Transactions on Robotics and Automation, vol. 17, no. 6, pp. 47–51, 2001.

[12] I. Suzuki and M. Yamashita, “Distributed anonymous mobile robots: formation of geometric patterns”, SIAM J. Computer, vol. 28, no. 4, pp. 1347–1363, 1999.

[13] X. Defago and A. Konagaya, “Circle formation for oblivious anonymous mobile robots with no common sense of orientation”, In Proceedings of the 2nd ACM international work principles of mobile computing, pp. 97–104, 2002.

[14] T. Balch and R. C. Arkin, “Behavior-based formation control for multirobot teams”, IEEE Transactions on Robotics and Automation, vol. 14, pp. 1-15, 1998.

[15] W. Ren and R. W. Beard, “Formation feedback control for multiple spacecraft via virtual structures”, IEEE Proceeding on Control Theory and Applications, vol. 151, no. 3, pp. 357-368,2004.

[16] V. Tran and S. Lee, “A Stable Formation Control Using Approximation of Translational and Angular Accelerations”, International Journal of Advanced Robotic Systems, vol. 8, no. 1, pp. 65-75, 2011.

[17] S. S. Ge and C. Fua, “Queues and artificial potential trenches for multi-robot formations”, IEEE Transactions on Robotics, vol. 21, no. 4, pp. 646–656, 2005.

[18] A. Krishnamurthy and R. Preis, “Satellite formation, a mobile sensor network in space”, Proceedings of the IEEE International Symposium on Parallel and Distributed Processing, pp. 243–249, 2005.

[19] A. Muhammad and M. Egerstedt, “Connectivity graphs as models of local interactions”, Proceedings of the IEEE Conference on Decision and Control, pp. 124–129, 2004.

[20] D. Sun, C. Wang, W. Shang and G. Feng, “A Synchronization Approach to Trajectory Tracking of Multiple Mobile Robots While Maintaining Time-Varying Formations”, IEEE Transactions on Robotics, vol. 25, no. 5, pp. 1074 - 1086, 2009.

[21] I. M. Sanhoury, S. H. Amin and A. Husain, “Switching Between Formation in a Moving Shape for Multi-Robots via Synchronization Approach”, Elsevier, International Symposium on Robotics and Intelligent Sensors, vol. 41, pp. 678 –684, 2012.

[22] I. M. Sanhoury, S. H. Amin and A. Husain, “Switching between formations for multiple mobile robots via synchronous controller”, in proceeding IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA), pp. 352–357, 2012.

[23] A. T. Rashid, M. Frasca, A. A. Ali, A. Rizzo, L. Fortuna, “Multi-robot localization and orientation estimation using robotic cluster matching algorithm”, Elsevier / Robotics and Autonomous Systems, vol.63, Part 1, pp. 108-121, 2015.

[24] A.T. Rashid, A.A. Ali, M. Frasca, L. Fortuna, Multi-robot collision-free navigation based on reciprocal orientation, Robotics and Autonomous Systems, vol. 60, no. 10, pp. 1221–1230, 2012.

[25] A.T. Rashid, A.A. Ali, M. Frasca, L. Fortuna, Path planning with obstacle avoidance based on visibility binary tree algorithm, Robotics and Autonomous Systems, vol. 61, pp. 1440–1449, 2013.

[26] J. Lee, S. Venkatesh and M. Kumar, “Formation of a geometric pattern with a mobile wireless sensor network”, Journal of Robotic System, Vol. 21, No. 10, pp. 517–530, 2004.