Dynamic Formation Control With Heterogeneous Mobile Agents*

Kadir Cimenci[1] and Aydan M. Erkmen[2]

Abstract— **Formation control in robotics is a growing topic where research works are mainly geared towards heterogeneous swarm colonies under either decentralized control or limited centralization. Swarm robotics where decentralization is applied, nevertheless assume that the agents are capable of getting global information about the whole swarm.Moreover in the literature, formation control is generally done for known fixed shapes that can be defined mathematically. However no dynamically changing shapes are envisaged and no shape transitions are clearly handled in those works. In this project, we attempt to bring a clear impact to the literature by focusing on tracking and realizing formation shapes under dynamically changing formation shape demands. We have proposed a novel method named Bubble Packing method for dynamic formation control and compared the performance of this method with Artificial Forces method which is generally used in literature in formation shape generation problems.**

I. INTRODUCTION

Formation control problem have different subproblems like formation shape generation, formation reconfiguration&selection and formation tracking [1]. In formation shape generation, agents are expected to get a formation shape which can be defined by external commands or with some mathematical constraint functions [2]. One general approach is to consider artificial potential functions. Ekanayake and Pathirana [3] have presented an artificial potential function method based on the consideration of the problem as controlling the position of a swarm into a shape, bounded by a simple closed contour. This approach results in deploying uniformly of swarm agents within the contour. Their work provides analysis about the stability and robustness of the proposed system based on Lyapunov like functions. In their work, desired formation shapes are defined with some analytical expressions and individual control laws for agents are composed with artificial potential functions by using these analytical expressions. Liao et al. at [6], have implemented potential function based formation control method by assigning positions of the agents in the formation to known locations. Chang et al. at [7], have implemented a combined method of potential field approach within a leader follower structure for a multi UAV swarm. In real world applications, it may not be possible to have the analytical expressions of the desired formation shapes. In our project, we have focused on designing control laws which do not depend on the analytical expressions of the formation shapes. Another disadvantage of the artificial potential based approaches is that , the control forces applied to individual agents are determined instantaneously in accordance with that agent's and the other agents' positions and they cannot guarantee the optimization of the total distance travelled by the agents. A common drawback related with this type of solution is that, there is a possibility to have local minimas in the solution where an agent reaches an undesired point in configuration space under the equilibrium of different types of artificial force components. In that case the total control input acting on the agent will be zero because of cancelling force vectors which has opposite directions to each other generated by formation shape and obstacles etc. In this strategy, the solution may converge slowly to the steady state due to absence of generalized goal states for individual agents in the final formation shape. Because, there are no specific goal states for the individual agents to reach at the final configuration and they are expected to get a global equilibrium with the help of different artificial force components.

One of the subproblems studied in formation control is formation tracking. The main objective of this problem is to maintain a desired formation with a group of robots, while tracking or following a reference trajectory. The most general strategy to provide a solution for this problem is leader-follower swarm structures [4], [5]. In leader follower strategy, some of the agents in the swarm are the leaders to manage the rest of the swarm to achieve a specific task and the rest of the agents act as followers. Sun et al. at [4], tried to solve the objective function which represents the operation of the team in an environment with obstacles. They have implemented a leader follower approach in which leader follows a trajectory and rest of the team follows this leader by keeping their optimal formation which is generated to maximize a given objective function. Xiao et al. at [5] have implemented a neural-dynamic optimization based nonlinear model predictive controller for a leader follower scheme. These leader-follower strategies have a common drawback related with the dependency on the leader agent, because it may not be possible for the rest of the swarm to continue mission in case of a failure on leader agent. This approach creates a single point of failure system and if this central unit fails during mission, swarm cannot achieve the desired task. In this work, it is aimed to implement a solution in which every agent is responsible for contributing on decisions and reaching a global consensus.

Some of works related with formation control define mathematical constraints and objective functions to be minimized with consensus of swarm [8], [9]. Mora et al. at [8], have implemented a formation control method among obstacles with a distributed method. They define a pre-defined set of $f \in N$ formations like square, line, diamond etc. rather than complex formation shapes. Yu et al. at [9], have implemented a distributed circular formation control in which a group of agents travel on a common circle by maintaining their positions evenly spaced. In real world applications, it may be needed to define complex formation shapes rather than simple geometrical ones. In this project, we have implemented a formation control algorithm which doesn't require the analytical expressions of desired formation shapes. Our approach doesn't depends on pre-defined set of formation shapes.

Specific tasks including different missions, requires agents with different capabilities and this kind of tasks may not be achieved with swarms composed of homogeneous agents [10]. In our work, one of the objectives is to implement a formation control system with heterogeneous agents.

In this paper, we propose a solution named Bubble Packing method, to achieve the objectives discussed in this section. We have compared the performance of this method with the Artificial Forces method which is commonly used in formation control problems. We have implemented algorithms which can adapt themselves to dynamically changing formations for both of these methods.

This paper is organized as follows. Section 2 provides information related with our methodology. Section 3, describes the artificial forces methodology. Section 4, introduces the implementation details of our proposed Bubble Packing method. Section 5, compares the simulation results of Artificial Forces and Bubble Packing methods in total displacement and dynamically changing formations. Section 6 presents our conclusions.

II. METHODOLOGY

In our project, we have reduced down the problem of dynamical formation control of heterogeneous mobile robots into two subproblems such as local positioning and formation control. Local positioning subsystem proposes a solution for the localization of agents in the workspace. On the other hand, formation control subsystem implements algorithms to cover the desired formation shape homogeneously with the agents in the swarm. In this paper, we have focused on our solution to the formation control problem in details.

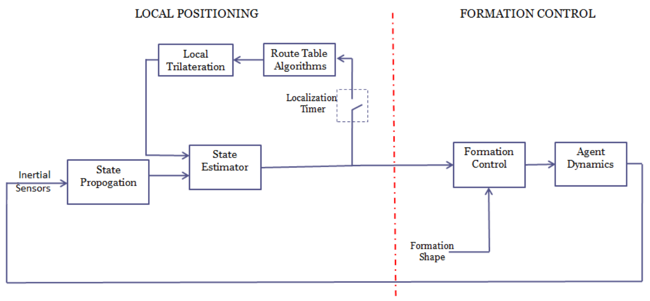


Fig. 1. General Block Diagram of the Provided Solution

A general block diagram of the system design is illustrated in Figure 1. According to this diagram, local positioning system provides a basis for the formation control problem with agents state vectors composed of translational positions and velocities. Basically, this subsystem provides position and velocity data to agents in the swarm. Formation control subsystem proposes a solution to achieve desired complex shapes with heterogeneous agents in the swarm. Formation control system gets the requested formation shape externally and creates individual control laws for agents to cover the desired shape homogeneously.

III. ARTIFICIAL FORCES METHOD

In Artificial Forces method we have implemented potential fields over each agent arised from the interactions between agents, formation shape and environment. The final positions of the agents at the formation shape are determined with local equilibrium of the swarm in which every agent is at balance under the total force acting from the environment. Basically we have implemented four different kinds of artificial forces named; intermember forces representing the forces created by other agents in the swarm to achieve collision avoidance; attractive forces representing the forces created by desired formation shape to attract the agent into the shape; repulsive forces created by formation shape to keep agents inside the desired formation shape; obstacle forces to provide collision avoidance with obstacles.

IV. BUBBLE PACKING METHOD

In Bubble Packing, we reduced the formation control problem into two subproblems. 1)Partitioning the desired formation shape into potential goal states to cover the desired formation shape homogeneously. 2)Decision process to assign agents to these goal states continuously in order to minimize the total displacement of swarm while achieving the desired formation shape. During this decision process, the cost of reaching different goal states is defined with the displacement on the shortest path while reaching that goal state which is reduced by assigning each agent to proper goal states.

A. PARTITIONING FORMATION SHAPE INTO GOAL STATES

This shape partitioning process basically depends on covering the formation shape with a proper number of bubbles by packing them tightly. [12]. The algorithm places bubbles with their initial conditions and apply interbubble forces to distribute the bubbles homogeneously inside the shape. Here, the main idea is to generate a mesh for a surface with identical bubbles to mimic a regular Voronoi diagram with the vertices represented by the centers of these bubbles. We define coverage circles as circles with minimum radius which covering the whole collision surface of an agent in 2D workspace. We implement the algorithm by representing the agents in the swarm as bubbles with the radius of their coverage circles and create a mesh by using these bubbles.

The bubbles are distributed homogeneously with this process under two kinds of forces, interbubble forces and shape repulsive forces. The interbubble forces are proximity-based forces so that a system of bubbles is in equilibrium when bubbles are distributed over the whole formation shape. The implemented force equation between agent i and j is given as

$$f_{i,j}(l) = \begin{cases} -0.1l^3 - 0.2l^2 - 0.3l + 4; & \text{when } 0 \leq l \leq 2.5 \\ 0; & \text{when } l > 2.5 \end{cases} \quad (1)$$

where l is the distance between the centers of the related bubbles.

The shape repulsive forces have the same characteristics with the repulsive artificial forces in [3].

The bubbles are distributed homogeneously under the influence of these two forces and they get an equilibrium state in which the total net forces acting on individual bubbles reaches zero. Figure 2 shows a simulation output of shape partitioning algorithm. The coverage circles are homogeneously distributed in the formation shape with the help of interbubble and shape repulsive forces. The final equilibrium states of the bubbles determine the potential goal states $g_i \in G$ of the agents in the swarm to cover the formation shape.

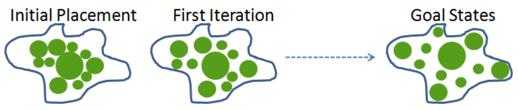


Fig. 2. Shape Partitioning Algorithm

B. DECISION PROCESS ON GOAL STATES

Each agent decides on where to position in a given set of possible goal states $g_i \in G$. Cost of reaching different goal states is the main criteria for each agent during this decision process. Given goal states and cost values to these goal states, each agent decides where to position in the formation. This process is held to optimize the utility of swarm with a collaboration. Our solution for this problem is to make each agent to calculate its own costs to reach the goal states $g_i \in G$ and then reach a global consensus with the other agents to minimize the overall displacements of the swarm.

The algorithm which handles the assignment process of the agents to the goal states, is implemented at four stages. At the first stage, each agent calculates its own free configuration space. Secondly, agents calculate their visibility graphs by using their free configuration space. At the third stage, agents calculate and report their costs to reach each of the goal states $g_i \in G$ with the help of Dijkstra's algorithm. These cost values are defined as the displacements to reach a goal state on shortest path. Finally, agents are assigned to the goal states with the help of Hungarian algorithm which uses the cost values reported by the agents. Hungarian algorithm handles the assignment process minimizing the overall cost(i.e. total displacement) of the swarm.

1) *Configuration Space:*

We have defined the shortest paths to the goal states of $g_i \in G$ in the free configuration space to avoid collisions with the workspace obstacles [13]. In our implementation, each agent calculates its free configuration space by extracting their forbidden spaces from the configuration space itself.

Assume an environment with set of obstacles $S = \{P_1, P_2, ..P_t\}$. Configuration for agent i can be described with the position of the center of its coverage circle with $R = \{x_i, y_i\}$. Configuration space of i^{th} agent is the workspace itself and represented by $C(R_i)$. This configuration space is composed of two subspaces; free configuration space and forbidden configuration space of agent i [13].

$$C(R_i) = C_{free}(R_i, S) + C_{forb}(R_i, S) \quad (2)$$

In our implementation each agent calculates its forbidden space by augmenting the workspace obstacles with the help of Minkowski sum method [13]. Figure 3 shows a forbidden space related with an obstacle. Forbidden space for agent i, $C_{forb}(R_i, S)$, is the sum of the forbidden spaces calculated for each obstacle in the environment. Agents extract their forbidden space from the configuration space itself to calculate their free configuration spaces [13].

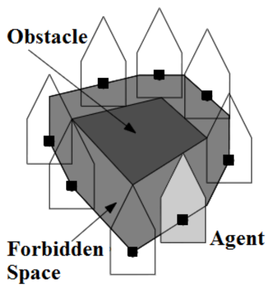


Fig. 3. Forbidden Space Caused by an Obstacle [13]

2) *VISIBILITY GRAPHS AND DIJKSTRA'S ALGORITHM:*

Collision avoidance is active while travelling towards to the goal states $g_i \in G$ by defining the shortest paths in the free configuration spaces of the agents. In [13], an additional constraint for the shortest path is defined as follows :

> The shortest path between p_{start} and p_{goal} among a set S of augmented polygonal obstacles consists of the arcs of the visibility graph $\gamma_{vis}(S^*)$ where $S^* := S \cup \{p_{start}, p_{goal}\}$

A visibility graph, $\gamma_{vis}(S^*)$, is a graph which is set of interior nodes representing the vertices of the set of obstacles, S, in the environment and edges which represents visible connections between these nodes(which are not intersecting interior region of an obstacle) [13]. We insert all of the goal states $g_i \in G$ in the visibility graphs of each agent to calculate the shortest paths to these goal states. In the implementation phase, we use obstacles augmented with the Minkowski sums. Let these set of augmented polygonal obstacles represented with $S_i \subset S$. Algorithm to calculate the visibility graph of $S^* := S \cup \{g_i \in G\}$

where $VISIBLE\_VERTICES(v, S)$ algorithm checks whether line segments drawn from v to all vertices in S is intersecting interior area of any obstacle in the environment and returns the non-intersecting edges. By using this visibility graph, each agent calculates its own shortest path to reach

```
Data: Set of Vertices Included in $S^*$
Result: Visibility Graph of $S^*$
Initialize a graph $\gamma = (V, E)$ where $V$ is the set of all vertices in $S^*$ and $E = \oslash$
for <all vertices $v \subset V$ > do
    W = VISIBLE_VERTICES($v$,S);
    Add edges W to list E;
end
```

Algorithm 1: VISIBILITY_GRAPH

all goal states $g_i \in G$ with the help of Dijkstra's algorithm by adding these goal states in visibility graph.

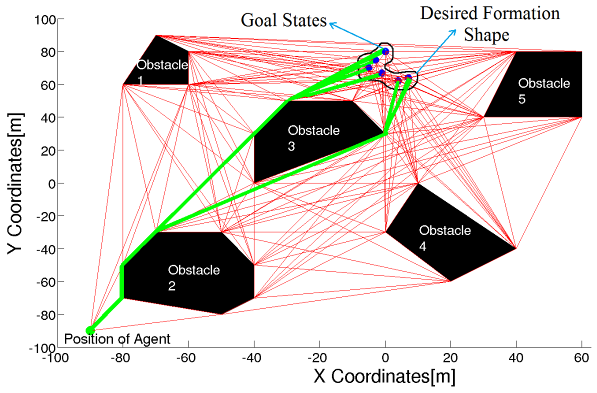


Fig. 4. Shortest Paths to Goal States $g_i \in G$ in a Visibility Graph

Figure 4 shows a simulation output executed with 5 workspace obstacles and 6 goal states in desired formation shape. In this algorithm, agent first calculates its own visibility graph by adding goal states $g_i \in G$ in graph as nodes. The visibility graph is illustrated with thin red edges in the figure. Our algorithm calculates the shortest paths to the goal states, which are given with thick green edges in the figure. We have used Dijkstra algorithm to compute the shortest path between two nodes in graph with multiple edges. In our work, the weights of the edges in γ_{vis} , are calculated with the Euclidian distance between nodes in the workspace. With the help of Dijkstra algorithm, agents calculate the shortest paths to all available goal states. These cost values are used to assign the agents to goal states by minimizing the total displacement.

3) DECISION PROCESS ON FINAL GOAL STATES: We provide an algorithm to calculate the costs to the goal states $g_i \in G$ with the help of Visibility Graphs and Dijkstra's algorithm. These costs are defined as displacements on the shortest paths to the goal states in the visibility graphs for each agent. Our aim is to minimize the total displacement of the individuals while achieving the desired formation shape. For this purpose we implement an algorithm to minimize the overall displacement of whole swarm while achieving a formation shape. The problem related with this process can be defined as follows:

We have n number of agents in our swarm and n number of goal states $g_i \in G$ placed in desired formation shape. Each agent has reported their costs (i.e. minimum displacements) to reach each of these goal states and there is a necessity to implement an algorithm to assign the agents to these goal states by minimizing the total displacement of the swarm. This is a generalized assignment problem and we use Hungarian algorithm which is a combinational optimization algorithm to solve this problem. To implement this algorithm, a complete bipartite graph $G = (S, T, E)$ with $n \in S$ agents and $g_i \in T$ goal states is constructed.In this graph, each agent has a cost for all goal states. Figure 5 shows an example Bipartite Graph with 5 agents.

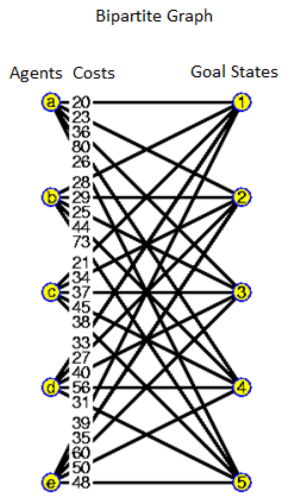


Fig. 5. Sample Bipartite Graph Used in Assignment Problem [14]

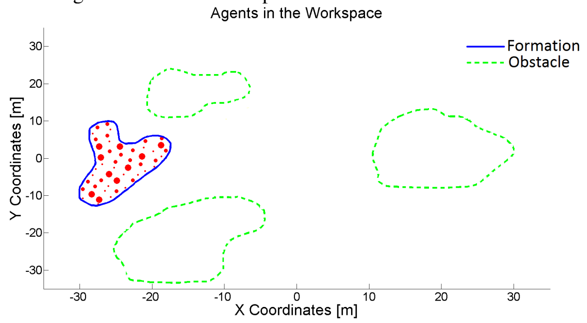
We have defined a cost matrix C to implement the Hungarian algorithm. The dimensions of the cost matrix is nxm in which each element represents the cost of assigning the goal state m to the agent n. Since we have equal number of agents and goal states, the cost matrix will be a square nxn matrix. Hungarian algorithm returns a vector of $nx1$ and each row in this vector represents the goal state that the related agent is assigned. With this assignment, the total cost of the swarm is minimized.

V. SIMULATION RESULTS

A. TOTAL DISPLACEMENTS OF AGENTS

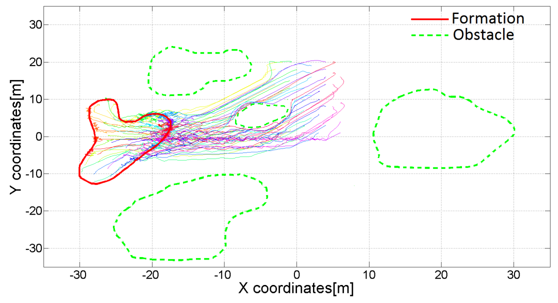
With Bubble Packing method, we aim to minimize the overall displacement of the agents in the swarm while getting the desired formation shape. To compare the total displacement of the swarm while getting the desired shape for two different methods, we have done Monte Carlo simulations with 1000 iterations. These simulations are handled for the same formation shapes with different initial conditions of the agents. Trajectories of the agents are recorded from the initial positions to the goal states for both Artificial Forces and Bubble Packing methods. Figure 6 shows a simulation output in MATLAB environment where red circles represent the coverage circles of agents.

Fig. 6. Formation Shape in MATLAB environment



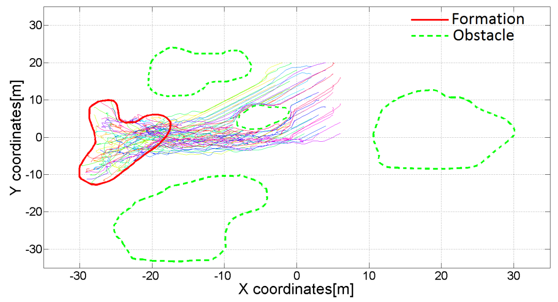
Trajectories of the agents towards to the desired shape in Artificial Forces method are shown at the Figure 7. It is possible to see that agents are not directed towards predetermined goal states, they are randomly placed in the desired shape. This approach increased the total displacements and settling times of the agents.

Fig. 7. Agent Trajectories with Artificial Forces Method



Trajectories of the agents towards to the desired shape in Bubble Packing method is shown at the Figure 8. In this method, each agent directly tries to reach its goal state. This approach reduces the total displacement and settling time.

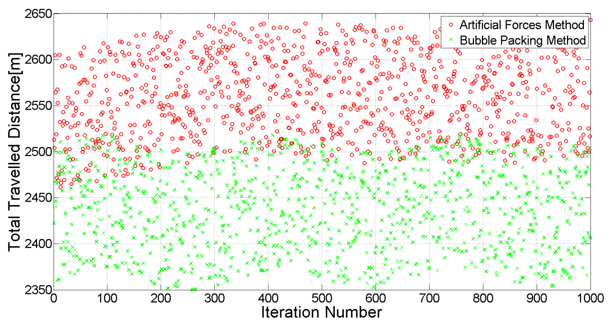
Fig. 8. Agent Trajectories with Bubble Packing Method



It can be seen that the trajectories of the agents are more chaotic and complex in the Artificial Forces method when compared to the Bubble Packing method. Main cause for this situation, the agents are under the effect of a total force which is instantly changing both in amplitude and direction with the local interactions with the environment in Artificial Forces method. The only force component which provides the distribution of the agents in the formation shape is the intermember forces which is dynamically changing so much with the local instant neighbors of the agents in the environment. On the other hand, Bubble Packing method implements an algorithm in which every agent is directed to a goal state where total displacements are minimized. This approach prevents the chaotic appearance of the trajectories and minimizes the displacements of the agents.

To compare the total displacements, we have done Monte Carlo simulations with 1000 iterations and the results are presented in Figure 9. According to this figure, Artificial Forces method has a higher mean value of total travelled distance while achieving the same formation shape with same initial conditions.

Fig. 9. Total Travelled Distances for Shape 1



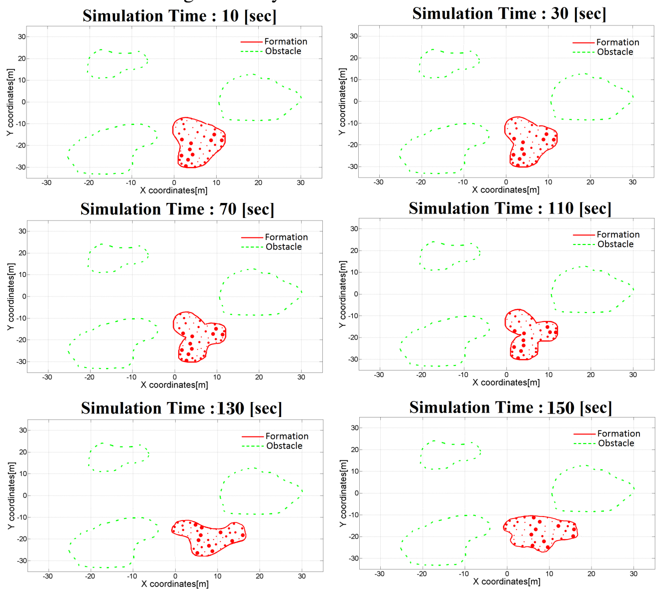
B. *DYNAMICALLY CHANGING FORMATIONS*

In dynamic formation control, desired formation shape is changed dynamically in a smooth fashion. Agents have to adapt themselves to this changing formation shape and cover the shape in a smooth continuous manner. Bubble Packing method determines goal states in a formation shape and it is possible to keep this shape partitioning process running in the background to dynamically adapt the goal states to the changing formation shape. Artificial Forces method calculates control inputs for the individuals according to the current environment conditions and the formation shape. There is no need to update the algorithm for dynamically changing formation shapes.

Figure 10 illustrates a simulation result of dynamically changing formation shape. In this figure, agents reposition in the changing formation shape dynamically.

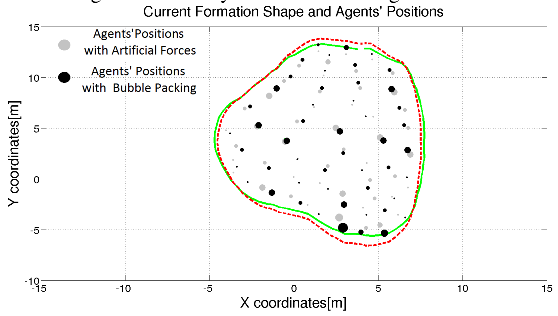
Since, agents directly adapts themselves to the changing formations shape in Artificial Forces method, it is expected to have a faster response than the Bubble Packing method. In Bubble Packing method, goal states are adapted to the changing formation shape and agents try to reach these goal states in runtime. This kind of approach introduces an additional latency to the response of the swarm to dynamically changing formation shapes. This situation is illustrated in Figure 11.We have compared the responses of these two algorithms to dynamically changing formation shapes. Bubble Packing and Artificial Force algorithms are executed with the same dynamical formation shapes and same initial positions of the agents. In these figures, black circles are representing

Fig. 10. Dynamic Formation Control



the agents' positions with Bubble Packing method and gray circles are representing the agents' positions with Artificial Forces method at the same time instance relative to the beginning of the simulations. Formation shape with blue line illustrates the current formation shape. It can be seen that the gray agents are adapted to the current formation shape better than black ones. The shape with dotted red line, represents the estimated coverage area of black agents in the environment and this shape differs from the current formation shape locally at some boundary regions. This shows that agents' response to the dynamically changing formation shape is faster with Artificial Forces method. Bubble Packing algorithm has a slower response to the dynamically changing shapes. It is appropriate to use Artificial Forces method in rapidly changing dynamical shapes if it is critical to adapt the swarm to the formation with low latency.

Fig. 11. Latency in Bubble Packing Method-3



VI. CONCLUSION

In this project, we have implemented 2 different solutions, Bubble Packing and Artificial Forces methods for dynamical formation control problem. Bubble Packing method has better performance in total displacement metric due to the absence of predetermined goal states in Artificial Forces method. In this method, we have partitioned the desired formation shape into goal states and we assign the agents to these goal states continuously to minimize the total displacement of the swarm. This approach has reduced down the total displacement and settling time of the agents while covering the desired formation shape. On the other hand, Artificial Forces method has better response to the dynamically changing formations, because it directly applies individual control laws based upon the instant conditions of the workspace. In Bubble Packing method agents try to reach the goal states which are dynamically changing their positions. This cascaded process introduces additional latency to the response of the swarm. Both solutions has advantages in different phases of formation control. It may be useful to implement a hybrid solution, including both of these methods to cover formation shape generation and dynamical formation control problems.

REFERENCES

[1] K. Kanjanawanishkul, "Coordinated path following control and formation control of mobile robots," Ph.D. dissertation, Univ. of Tubingen, Tubingen, Germany, 2010.

[2] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions On Robotics And Automation*, vol. 17, pp. 947–951, 2001.

[3] S. Ekanayake and P. Pathirana, "Formations of robotic swarm: An artificial force based approach," *International Journal of Advanced Robotic Systems*, vol. 7, pp. 173–190, 2010.

[4] C. G. C. Xinmiao Sun, "Optimal dynamic formation control of multi-agent systems in constrained environments," *Automatica*, vol. 73, pp. 169–179, 2016.

[5] C. P. C. Hanzhen Xiao, Zhijun Li, "Formation control of leaderfollower mobile robots systems using model predictive control based on neural-dynamic optimization," *IEEE Transactions on Industrial Electronics*, vol. 63, 2016.

[6] J. L. W. X. D. F. L. Fang Liao, Rodney Teo and K. Peng, "Distributed formation and reconfiguration control of vtol uavs," *IEEE Transactions on Control Systems Technology*, 2015.

[7] K. C. Y. Xia and K. Huang, "Uav formation control design with obstacle avoidance in dynamic threedimensional environment," *Springer Plus*, vol. 5, 2016.

[8] L. L. Xiao Yu, "Distributed circular formation control of ring-networked nonholonomic vehicles," *Automatica*, vol. 68, pp. 92–99, 2016.

[9] M. S. Javier Alonso-Mora, Eduardo Montijano and D. Rus, "Distributed multi-robot formation control among obstacles: A geometric and optimization approach with consensus," in *International Conference on Robotics and Automation*, 2016.

[10] M. D. et al., "Swarmanoid: A novel concept for the study of heterogeneous robotic swarms," *IEEE Robotics & Automation Magazine*, vol. 20, pp. 60–71, 2013.

[11] M. L. Rasulov, *Methods of Contour Integration*, 3rd ed., W. T. K. H. A. Lauwerier, Ed. Elsevier, 2014.

[12] K. Shimada and D. Gossard, "Bubble mesh: Automated triangular meshing of non-manifold geometry by sphere packing," in *ACM Symposium on Solid Modeling and Applications*, 1995.

[13] M. Berg, O. Cheong, M. Kreveld, and M. Overmars, *Computational Geometry*. Springer, 1998.

[14] J. Sibeyn, "Graph algorithms," https://www8.cs.umu.se/ jopsi/dinf504/ chap14.shtml, last visited on April 2016.