Formation Control with Potential Functions and Newton's Iteration

Y. Sinan Hanay, H. Volkan Hünerli, M. İlter Köksal, Andaç T. Şamiloğlu, and Veysel Gazi

Abstract— **In this study, we analyze formation control (formation of a geometrical shape) of an autonomous multi-robot system with the use of artificial potential functions and Newton's iteration. The method is independent of the low-level vehicle dynamics of the robots and therefore it can be applied to different type of robots. We also perform numerical simulations to examine the performance of the method.**

I. Introduction

Coordinated behavior can be seen in many organisms in nature such as swarms of bacteria, flocks of birds, schools of fish, colonies of ants or bees, etc. In such animal groups usually there is no leader and the resulting coordinated behavior is emergent or self-organizing. As a result of such coordinated behavior complex tasks or structures can be performed by local interactions of relatively simple individuals.

Inspired by the efficiency and success of such animal groups recent robotics research has been focusing on multi-agent systems or basically groups of autonomous mobile agents (which could be ground, undersea, air, or space vehicles/robots, mobile sensors etc.). Such systems are of interest due to several inherent advantages: (i) Tasks may be too complex or sometimes impossible for a single agent to achieve; (ii) Performance of the system may be improved by using multiple agents; (iii) The agents of a multi-agent system may be easier to build, cheaper, more flexible, and more fault tolerant than a single agent designed for each separate complex task; (iv) The methods and algorithms developed for cooperative mobile robotics can also be beneficial in the problems of other sciences; especially for social sciences including organization theory, economics, cognitive psychology or life sciences like theoretical biology and animal ethology [1].

Biologists have been studying swarms in nature in order to construct mathematical models [2], [3], [4], [5] for swarm aggregations and coordination. Inspired by these works, a recent series of studies [6], [7], [8], [9], [10], [11], [12], [13] has provided rigorous stability and convergence analysis of swarm aggregations. One of the early literature surveys on the topic of multi-agent (multi-robot) systems is the study of Mataric [14] in 1995. There are also some recent books [15], [16] that may be useful references about the swarms in nature and engineering applications that inspire from these swarms. The references in [17], [18], [19], [20] are some relevant books and special issues of journals that include the recent studies on the swarm-robotics. In [21] the advantages and some applications of swarm-robotics are presented and principal definitions of some properties of these systems are stated. One very recent survey that considers multi-agent systems from the perspective of control engineering can be found in [22].

Formation of geometric shapes with autonomous robots is a particular type of the coordination problem of multi-agent systems. There are different methods for providing formation control. Most known of these methods are behavior based methods [23], Lyapunov theory and functions based and/or graph theory based methods [24], [25], [26], non-linear system theory based methods [27], [28] and artificial potential functions based methods [7], [29], [30], [31], [13], [32], [33], [34], [35]. Artificial potential functions are used widely for robot navigation and motion control [36], [37]. Another study on navigation and path planning of robots using artificial potentials is the work by Erkmen and her colleagues [38] in which the authors developed a path planning algorithm for an autonomous robot moving in an unknown environment. In their work the obstacles were fixed and the certainty of the knowledge the robot has about its environment is represented with an entropy.

In this study we will show that formation control can be achieved with the utilization of Erkmen's method [38] with some simple modifications. In particular, in this article for each robot we define the rest of the robots as mobile targets and generate the corresponding target function accordingly. Then the robots plan their motion according to their time varying target functions. We utilize Newton's update rule for uptating the positions of the robots.

The article is organized as follows. In Section II we discuss the mathematical preliminaries and in particular the Erkmen's method [38] and the necessary modifications to the method in order to fit it into the formation control problem. In Section III we verify the effectiveness of the method through numerical simulations. Then we conclude the article with final remarks and mention some future directions.

II. Mathematical Model

In this section, we present the mathematical model considered by Erkmen [38] together with some modifications

This work was supported in part by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant No. 104E170 and by the European Commission under FP6 contract No. 045269.

Y. Sinan Hanay, H. Volkan Hünerli, M. İlter Köksal, and Veysel Gazi are with the Dept. of Electrical and Electronics Engineering, TOBB University of Economics and Technology (ETU), Söğütözü Cad. No: 43, 06560 Ankara, TURKEY. {hanay,vhunerli,i.koksal,vgazi}@etu.edu.tr

Andaç T. Şamiloğlu is with Başkent University, Department of Mechanical Engineering, Bağlıca Kampüsü, Eskişehir Yolu 20.km, Bağlıca 06810 Ankara. He is also a Ph.D. student at Middle East Technical University, Mechanical Engineering Department, İnönü Bulvarı, Çankaya, Ankara, TURKEY and graduate research associate at TOBB ETU. andacsam@baskent.edu.tr

needed to utilize it for formation control of autonomous robots. In the problem we consider here each robot's objective is to achieve a previously given geometrical figure or a formation pattern in coordination with the other robots. The robots use a local planning strategy instead of a global strategy. Since the method is iterative, the robot updates its own motion plan at each step utilizing the new position information of other robots.

In the Erkmen-Yeĝenoĝlu-Stephanou [38] method, artificial potential functions are used to define the positions of the stationary targets and obstacles in the state space. These functions are generated as discussed below. Let us define the target function as

$$F_A(X) = \left\{ \begin{array}{c} f_1(x_1,\ldots,x_n) \\ f_2(x_1,\ldots,x_n) \\ \vdots \\ f_n(x_1,\ldots,x_n) \end{array} \right\} = 0 \qquad (1)$$

Here $X = [x_1,\ldots,x_n]^\top \in \mathbb{R}^n$ is the position vector, n is dimension of the position space and $\mathbb{R}^n$ denotes the $n-dimensional$ Euclidean space. Let us represent the set of targets in the position space with

$$H_A = \{Z_i \in \mathbb{R}^n | i = 1,\ldots,m\}$$

Here Z_i denotes the position of the ith target and m is the number of targets. For any given set of targets H_A the corresponding nonlinear target function $F_A(X)$ is defined such that every $Z_i \in H_A$ is a root of $F_A(X)$. The first components of all Z_i target vectors z_{1i}'s are used to form the function $f_1(x)$.

$$\begin{aligned} f_1(x) &= x_1x_2 - \Big[x_1P_2^{m-1}(x_1) \\ &\quad -(x_1-z_{11})(x_1-z_{12})\ldots(x_1-z_{1m})\Big] \end{aligned} \qquad (2)$$

Similarly the jth components of Z_i vectors are used to create the function $f_j(x)$. In other words, jth components of the vectors Z_i are taken into account such that the zeros of the function $f_j(x)$ are given by

$$\big[(z_{1(j-1)}, z_{1(j)}), (z_{2(j-1)}, z_{2(j)}), \ldots, (z_{m(j-1)}, z_{m(j)})\big]$$

For example, let $x_j = P(x_{j-1})$ to be a $(m-1)$'th order polynomial. In this case

$$f_j = x_j - P(x_{j-1}), j = 2,\ldots,n,$$

verifies the required zeros and the target function constructed this way can be written as,

$$F_A(X) = \left\{ \begin{array}{c} x_1x_2 - P_1^m(x_1) \\ x_2 - P_2^{m-1}(x_1) \\ \vdots \\ x_n - P_n^{m-1}(x_{n-1}) \end{array} \right\} = 0 \qquad (3)$$

This target function is an artificial potential function that has zeros at the target points. For better understanding of this method let us give a numerical example. Let the dimension of the space be $n = 2$, the number of targets be $m = 3$ and the positions of the targets be given as $Z_1 = [1,1], Z_2 = [2,3], Z_3 = [3,8]$. In this case P_2 becomes a second order polynomial in the form of

$$\begin{aligned} P_2 &= c_1(x_1-z_{11})(x_1-z_{12}) + c_2(x_1-z_{12})(x_1-z_{13}) \\ &\quad + c_3(x_1-z_{11})(x_1-z_{13}) \end{aligned} \qquad (4)$$

and P_1 becomes a third order polynomial in the form of

$$P_1 = x_1P_2^{m-1}(x_1) - (x_1-z_{11})(x_1-z_{12})(x_1-z_{13}) \qquad (5)$$

For the values of Z_1, Z_2 and Z_3 above we have

$$\begin{aligned} P_2 &= c_1(x_1-1)(x_1-2) + c_2(x_1-2)(x_1-3) \\ &\quad + c_3(x_1-1)(x_1-3) \end{aligned} \qquad (6)$$

Our objective is to construct P_2 such that $(x_2 - P_2)$ has zeros at Z_1, Z_2, and Z_3. Solving the equations (4) and (5), we get coefficients $c_1 = 4$, $c_2 = 0.5$, and $c_3 = -3$ which result at the desired zeros. Then, by inserting P_2 into P_1 and for the above values of Z_1, Z_2 and Z_3 we have

$$P_1 = x_1P_2 - (x_1-1)(x_1-2)(x_1-3) \qquad (7)$$

and the corresponding target function $F_A(X)$ becomes

$$F_A(X) = \left\{ \begin{array}{l} x_1x_2 - (0.5x_1^3 + 3.5x_1^2 - 9x_1 + 6) \\ x_2 - 1.5x_1^2 + 2.5x_1 - 2 \end{array} \right\} \qquad (8)$$

Thus the target function $F_A(X)$ has its zeros at Z_1, Z_2, and Z_3. Next step after constructing target function $F_A(X)$ is to solve the equation $F_A(X) = 0$ by iterative methods. This will lead to a step by step motion towards the targets. As in [38] we use Newton-Raphson iteration method for this purpose.

Let the location of the robot at the k'th step be given by $X(k)$ and updated based on the Newton's method

$$X(k+1) = X(k) + A(k) \qquad (9)$$

where the step vector is given by

$$A(k) = -\Big[\nabla F_A(X(k))\Big]^{-1} F_A(X(k)). \qquad (10)$$

Our objective is to force the robots to form a geometric shape using the above method. Given any initial positions and any desired geometrical formation the robots should locate themselves to the desired inter-robot distances so as to form the desired geometrical shape. With this objective in mind we define the target set for each robot as the set of points that consists of the points defined at the desired distances from the other robots. For instance assume that there are three robots which are required to form an equilateral triangle with edge lengths d as shown in Figure 1.

For this case there exist two non-stationary (moving) target points for each robot at each step and these target points are on the lines that connect the robot to the other robots and at a distance d from the these robots (Figure 2). For example for the robot in the lower left corner in Figure 2 the target set consists of the points labelled by stars. Note that at each step -after the motion of robots- the positions of these points change and therefore, the target sets will be time-varying and need to be updated.

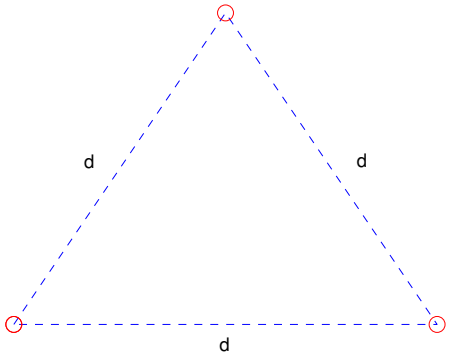

Fig. 1. The equilateral triangle that the robots should form.

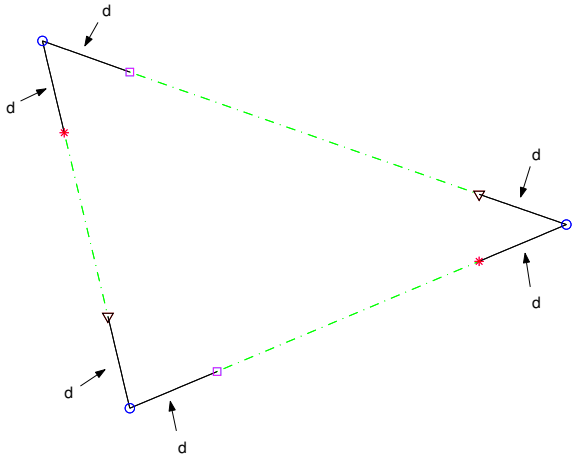

Fig. 2. The positions of the targets for each robot.

Let the number of robots in a robot group/swarm be N and the i'th robot's target set is shown with H_A^i. It is obvious that there is $N-1$ number of target points in the set H_A^i. These targets are defined by the robot's desired distance to the other robots in the desired formation, d_{ij}. (It is possible to define a unique formation by using graph-theoretic concepts and choosing less number of targets, but in this work for simplicity we assume that the graph is complete. The procedure should work for such cases as well.) To obtain the desired formation we can use two different approaches and these can be described as follows.

- Each robot (indexed by i) is directly generating a target function F_A^i using the set H_A^i and deciding the step-size according to the equation

$$A_i(k) = -\Big[\nabla F_A^i(X_i(k))\Big]^{-1} F_A^i(X_i(k)) \qquad (11)$$

  where $X_i(k)$ is the position of agent i at step k.
- Each robot (indexed by i) is generating a different potential function F_A^{ij} for the j'th robot ($j \neq i$) instead of a single target function F_A^i that covers all of the targets. Then the target set of robot i becomes

$$H_A^i = \bigcup_{j=1,\dots N, j\neq i} H_A^{ij},$$

where H_A^{ij} is the target set for robot i based on only robot $j \neq i$. Then robot i generates a target function F_A^{ij} according to each H_A^{ij} and defines its step according to the equation

$$A_i(k) = -\sum_{j=1, j\neq i}^{N} \Big[\nabla F_A^{ij}(X_i(k))\Big]^{-1} F_A^{ij}(X_i(k)). \qquad (12)$$

For the first case we need a total of N potential functions for N robots, whereas in the second case we need $N \times (N-1)$ potential functions for N robots. In the second method there are more potential functions to be calculated. However, in the first case, since the number of the targets for given potential is higher the polynomials in the potential function are of higher orders/degrees, whereas in the second case the polynomials are only of first order. In this study, we used the second choice and developed one target function for each target instead of a single target function for all of the targets. This results in significant decrease in complexity.

In this case, for each robot i we have $N-1$ number of target sets H_A^{ij} and equivalently for each of them there exists corresponding target function F_A^{ij}. The P_1 and P_2 polynomials that are used to form the target function are simplified significantly. In particular, the P_2 polynomial becomes a constant number and the P_1 polynomial turns into a first order polynomial. Let the j'th target set (due to the j'th robot) for the i'th robot $j \neq i$ be

$$H_A^{ij} = \{Z^{ij} \in \mathbb{R}^n\}$$

where $Z^{ij} = \Big[z_1^{ij}, z_2^{ij}, ..., z_n^{ij}\Big]^\top$. Note once more that Z^{ij} is not the position of robot $j \neq i$. t is the target point on the line connecting the positions of robots i and j on a desired distance d_{ij} from robot j (the distance d for the case in Figure 2. Then the corresponding polynomials become

$$P_l = z_l^{ij}, l = 2, \dots, n,$$

and

$$P_1 = x_1 z_2^{ij} - (x_1 - z_1^{ij})$$

and the corresponding target function given by

$$F_A(X) = \left\{ \begin{array}{c} x_1 x_2 - x_1 z_2^{ij} + (x_1 - z_1^{ij}) \\ x_2 - z_2^{ij} \\ \vdots \\ x_n - z_n^{ij} \end{array} \right\} = 0 \qquad (13)$$

is obtained. Although both methods seem very similar, the second method is more advantageous when the number of the robots increases.

Considering the second method, as stated above given, N agents at each step k, each robot generates $N-1$ number of target sets $H_A^{ij}, j = 1, \dots, N, j \neq i$ and the corresponding potential/target functions $F_A^{ij}(X_i(k)), j = 1, \dots, N, j \neq i$. Each robot determines its step size according to the relative difference

$$A_i(k) = \sum_{j=1, j\neq k}^{N} A_{ij}(k)$$

where we defined

$$A_{ij}(k) = -\left[\nabla F_A^{ij}(X_i(k))\right]^{-1} F_A^{ij}(X_i(k)). \quad (14)$$

In some situations robot's step size can be large due to the large relative difference between the agents and this may lead to convergence problems. Therefore, to reach the target or to achieve the desired formation some limitations should be applied on the relative difference obtained as the output of the Newton iteration. For this purpose, let us define the next position of the robot as

$$X_i(k+1) = X_i(k) + \lambda \Delta X_i(k) \quad (15)$$

where $\lambda > 0$ is the step size to be determined by the designer and $\Delta X_i(k)$ is the unit step vector determining the direction of motion and calculated as

$$\Delta X_i(k) = \frac{A_i(k)}{\|A_i(k)\|}. \quad (16)$$

The size of the difference or step size $\lambda > 0$ should be set adequately for a successful formation. Furthermore, robots should give more weight to the target/robot that is closer to itself while locating its next position. For this, the update difference should be inversely proportional with $\|A_{ij}(k)\|$. This condition can be achieved by choosing the step vector as

$$A_i(k) = \sum_{j=1, j\neq i}^{N} \frac{A_{ij}(k)}{\|A_{ij}(k)\|^2}, \quad (17)$$

Now, let us define the position error as

$$\xi_{ij}(k) = |\delta_{ij}(k) - d_{ij}| \quad (18)$$

where $\delta_{ij}(k) = \|X_i(k) - X_j(k)\|$ is the present distance between robots i and j, and d_{ij} is the desired distance between robots i and j.

To avoid sub-groupings problems in the swarm we choose the robots with higher position errors, $\xi_{ij}(k)$, to contribute more than the relatively accurately positioned ones. In our simulations we observed that by giving higher importance to the farther ones to the desired position, robots converge to the desired formation much faster. Therefore, the final constraint becomes

$$A_i(k) = \sum_{j=1, j\neq i}^{N} \xi_{ij}(k) \times \frac{A_{ij}(k)}{\|A_{ij}(k)\|^2}, \quad (19)$$

which together with (12) and (13) we utilized in the simulations below.

Since the method we consider here is discrete, once the robots achieve the formation they start to oscillate around their target point with a deviation that almost equals to step size. By implementing step size as adaptive, this problem can be avoided. One possible straightforward implementation of adaptive step size could be as follows.

Adaptive Step Size: For each agent i,

$$\text{if } |\xi_{ij}(k)| < 2\lambda_i \text{ for all } j \neq i \text{ then } \lambda_i = \frac{\lambda_i}{2}.$$

This results in a decrease of the amplitude of the oscillations and closer convergence to the target points.

Finally, we would like to state that the method we described may suffer from the local minima problem present in the potential function based approaches and convergence can only be guaranteed locally.

III. Simulation Results

In this section we present numerical simulation results in order to test the effectiveness and viability of the proposed method. In the simulations we used $N = 6$ agents and the step size is $\lambda = 0.01$. Two types of formation problems are taken into account: equilateral triangle and parallelogram. For the equilateral triangle, the desired length of edges is $d = 6$. Therefore, depending on the relative positions in the triangle, the distances between robots should be 3, 6 and $3\sqrt{3}$ to form the equilateral triangle.

Figure 3 shows the trajectories of the robots during the simulation for the equilateral triangle formation. The initial

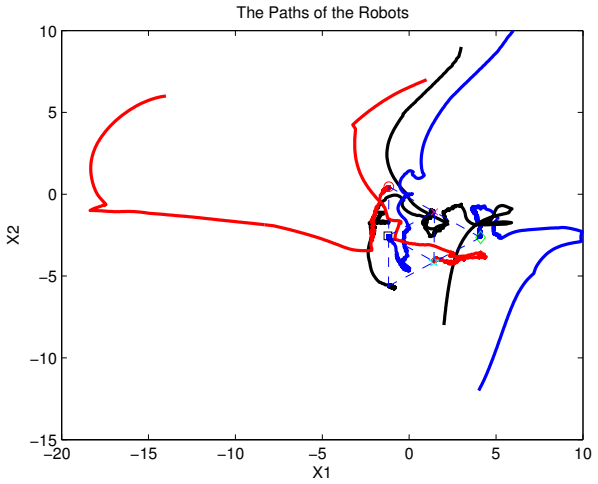

Fig. 3. The paths of the robots for the Equilateral Triangle formation.

positions of the agents were chosen randomly for this simulation. The figure reveals that the robots arrange themselves in the desired formation as predicted by the theory. Final positions of the agents - at the end of the simulation with 1400 steps - is shown in Figure 4.

Figure 5, shows the inter-agent distances with respect to time. As seen in this figure, agents converge to desired formation in about 400 steps. One important point to be mentioned is that, the inter-agent distances do not converge smoothly to the steady state values, there exist oscillations around the desired values. We can clearly see the oscillations of the inter-agent distances in Figure 6 which presents the zoomed portion of Figure 5. Distances oscillate around 6 and $3\sqrt{3}$. The oscillations are caused by the discrete time modeling of the proposed method. Robots perform oscillations around the corresponding target point, when their distance to the target point is smaller than the two times the step size. Figure 7 shows the inter-agent distances with respect to time with the

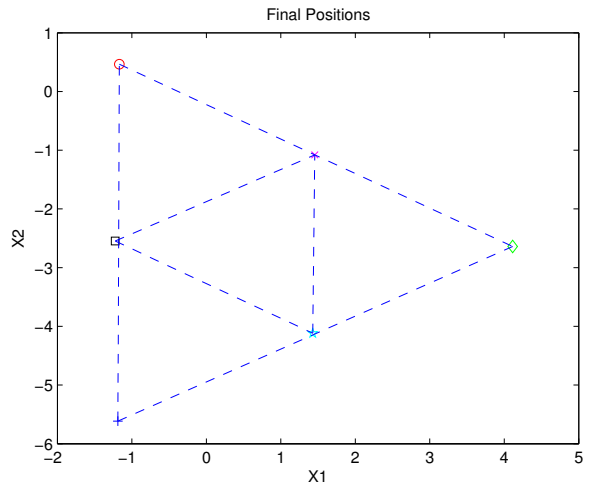

Fig. 4. Final positions of robots for the Equilateral Triangle formation.

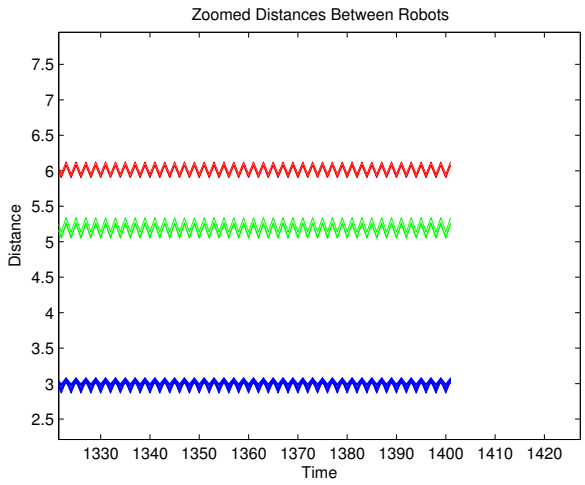

Fig. 6. Zoomed version of inter-agent distances.

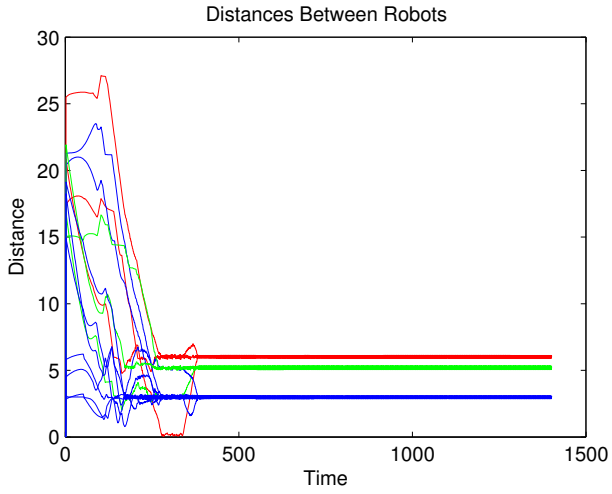

Fig. 5. Distances between robots for the Equilateral Triangle formation.

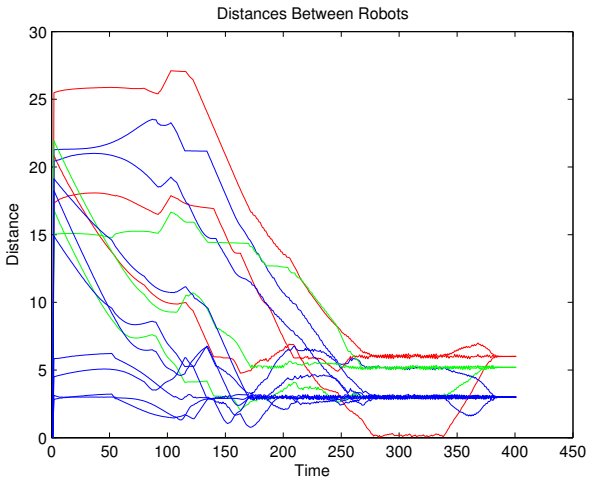

Fig. 7. Distances between robots for the Equilateral Triangle formation with Adaptive Step Size.

addition of adaptive step size. Recall that for this case the step size λ is halved once the distance of the robot to all the target points is less than 2λ. Figure 8 is the zoomed version of Figure 7 and note that the oscillations around the target point are not present in this simulation.

The second illustrative example consists of the formation of a parallelogram. In this formation, the desired inter-agent distances are set to 3, 6, $3\sqrt{2}$, and $3\sqrt{5}$ respectively. Figure 9 shows the trajectories of the agents for random initial positions. We used adaptive step size for this simulation. It can be seen from the figure that the agents converge to the desired formation as was the case in the previous example. The final positions are shown in Figure 10. The inter-agent distances are shown in Figure 11. The agents form the parallelogram after around 1800 steps for this particular simulation. Although it took longer time to form the parallelogram (compared to the case of equilateral triangle) for this particular simulation, this does not have to be always the case. The convergence speed may depend also on the initial configuration of the agents.

IV. Concluding Remarks

In this paper, we described a method for coordination and formation control of multi-agent systems, by making some modifications in the method of Erkmen, et al., which was developed for robots' motion planning. Note that here the method is a high-level method which is not based on any vehicle dynamics. The points generated by the method are discrete points which on implementations on real robots can be viewed as the way-points the robots have to move to. Then, in each robot, there should be a low-level control algorithm that guarantees that the robot will move to these (next) desired points in a finite time. The developed method is based on artificial potential functions and Newton-Raphson iteration, and since it is independent of low-level dynamics of robots, it can be applied to different types of vehicles. Future works may include the consideration of

Fig. 8. Zoomed version of inter-agent distances with Adaptive Step Size.

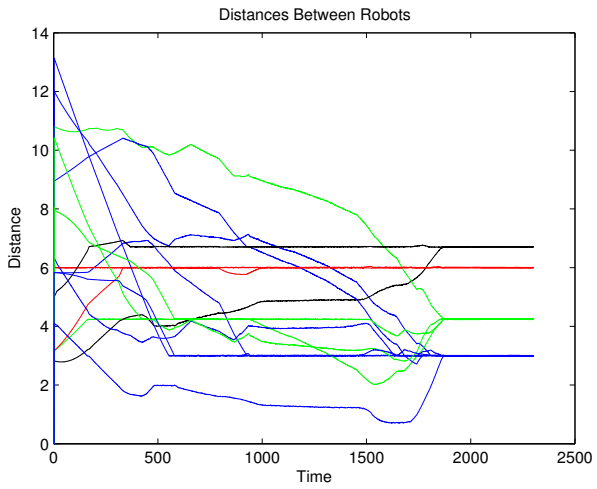

Fig. 11. Distances between robots for the Parallelogram formation.

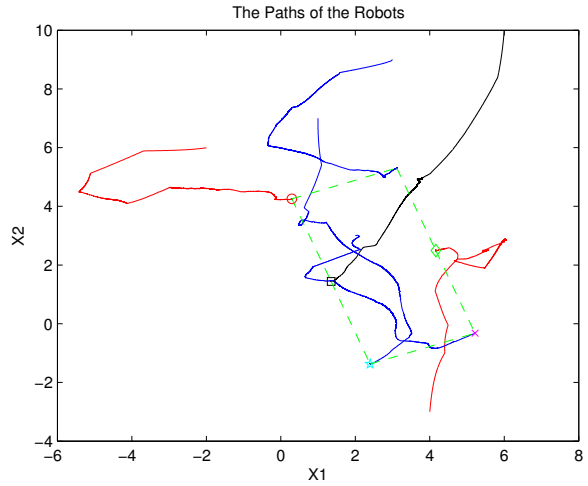

Fig. 9. The paths of the robots for the Parallelogram formation.

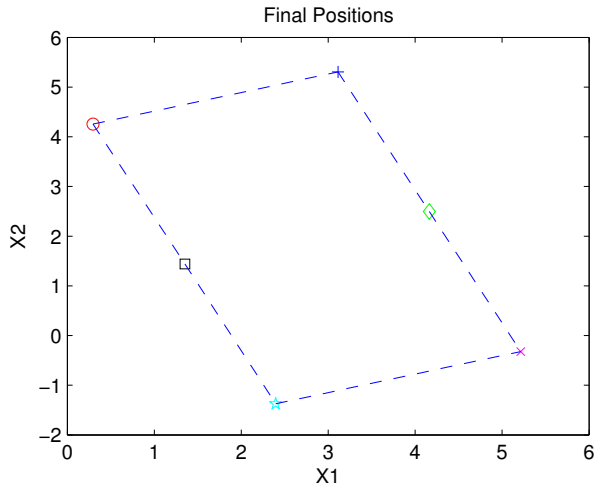

Fig. 10. Final positions of the robots for the Parallelogram formation.

environmental conditions and obstacles and/or collision-free navigation and formation maintenance during motion. Other issues that could be considered are possible time delays and uncertainties in the sensing of the position of the other robots as well as asynchronous motion of the agents.

REFERENCES

[1] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, no. 1, pp. 7–23, 1997.

[2] C. M. Breder, "Equations descriptive of fish schools and other animal aggregations," *Ecology*, vol. 35, no. 3, pp. 361–370, 1954.

[3] A. Okubo, "Dynamical aspects of animal grouping: swarms, schools, flocks, and herds," *Advances in Biophysics*, vol. 22, pp. 1–94, 1986.

[4] K. Warburton and J. Lazarus, "Tendency-distance models of social cohesion in animal groups," *Journal of Theoretical Biology*, vol. 150, pp. 473–488, 1991.

[5] D. Grünbaum and A. Okubo, "Modeling social animal aggregations," in *Frontiers in Theoretical Biology*, vol. 100 of *Lecture Notes in Biomathematics*, pp. 296–325. Springer-Verlag, New York, 1994.

[6] V. Gazi and K. M. Passino, "Stability analysis of swarms," *IEEE Trans. on Automatic Control*, vol. 48, no. 4, pp. 692–697, April 2003.

[7] V. Gazi and K. M. Passino, "A class of attraction/repulsion functions for stable swarm aggregations," *Int. J. Control*, vol. 77, no. 18, pp. 1567–1579, December 2004.

[8] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Trans. on Systems, Man, and Cybernetics: Part B*, vol. 34, no. 1, pp. 539–557, February 2004.

[9] Y. Liu and K. M. Passino, "Stable social foraging swarms in a noisy environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 1, pp. 30–44, 2004.

[10] Y. Liu, K. M. Passino, and M. M. Polycarpou, "Stability analysis of one-dimensional asynchronous swarms," *IEEE Trans. on Automatic Control*, vol. 48, no. 10, pp. 1848–1854, October 2003.

[11] Y. Liu, K. M. Passino, and M. M. Polycarpou, "Stability analysis of m-dimensional asynchronous swarms with a fixed communication topology," *IEEE Trans. on Automatic Control*, vol. 48, no. 1, pp. 76–95, January 2003.

[12] V. Gazi and K. M. Passino, "Stability of a one-dimensional discrete-time asynchronous swarm," *IEEE Trans. on Systems, Man, and Cybernetics: Part B*, vol. 35, no. 4, pp. 834–841, August 2005.

[13] V. Gazi, "Swarm aggregations using artificial potentials and sliding mode control," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1208–1214, December 2005.

[14] M. J. Mataric, "Issues and approaches in the design of collective autonomous agents," *Robotics and Autonomous Systems*, vol. 16, no. 2/4, pp. 321–332, 1995.

[15] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, New York, 1999.

[16] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*, Princeton University Press, New Jersey, April 2001.

[17] M. Dorigo and E. Sahin, Eds., *Special Issue on Swarm Robotics*, vol. 17, Autonomous Robots, September 2004.

[18] V. J. Kumar, N. E. Leonard, and A. S. Morse, Eds., *Cooperative Control: 2003 Block Island Workshop on Cooperative Control*, vol. 309 of *Lecture Notes in Control and Information Sciences*, Springer-Verlag, 2005.

[19] E. Sahin and W. M. Spears, Eds., *Swarm Robotics, A State of the Art Survey*, Lecture Notes in Computer Science 3342. Springer-Verlag, Berlin Heidelberg, 2005.

[20] E. Sahin, W. M. Spears, and A. F. T. Winfield, Eds., *Proceedings of the SAB06 Workshop on Swarm Robotics Swarm Robotics*, Lecture Notes in Computer Science (LNCS) 4433. Springer-Verlag, Berlin Heidelberg, 2007.

[21] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics: State-of-the-art Survey*, E. Sahin and W. Spears, Eds., Lecture Notes in Computer Science (LNCS 3342), pp. 10–20. Springer-Verlag, Berlin Heidelberg, 2005.

[22] V. Gazi and B. Fidan, "Coordination and control of multi-agent dynamic systems: Models and approaches," in *Proceedings of the SAB06 Workshop on Swarm Robotics Swarm Robotics*, E. Sahin, W. M. Spears, and A. F. T. Winfield, Eds., Lecture Notes in Computer Science (LNCS) 4433, pp. 71–102. Springer-Verlag, Berlin Heidelberg, 2007.

[23] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, December 1998.

[24] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc. of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 2864–2869.

[25] J. P. Desai, J. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 905–908, December 2001.

[26] P. Ögren, M. Egerstedt, and X. Hu, "A control Lyapunov function approach to multi-agent coordination," in *Proc. Conf. Decision Contr.*, Orlando, FL, December 2001, pp. 1150–1155.

[27] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 6, pp. 933–941, December 2003.

[28] V. Gazi, "Formation control of a multi-agent system using nonlinear servomechanism," *Int. J. Control*, vol. 78, no. 8, pp. 554–565, 20 May 2005.

[29] J. H. Reif and H. Wang, "Social potential fields: A distributed behavioral control for autonomous robots," *Robotics and Autonomous Systems*, vol. 27, pp. 171–194, 1999.

[30] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, December 2001.

[31] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *Proc. IFAC World Congress*, Barcelona, Spain, June 2002.

[32] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *Proc. Conf. Decision Contr.*, Orlando, FL, December 2001, pp. 2968–2973.

[33] P. Ögren, E. Fiorelli, and N. E. Leonard, "Formations with a mission: Stable coordination of vehicle group maneuvers," in *Symposium on Mathematical Theory of Networks and Systems*, August 2002.

[34] R. Bachmayer and N. E. Leonard, "Vehicle networks for gradient descent in a sampled environment," in *Proc. Conf. Decision Contr.*, Las Vegas, Nevada, December 2002, pp. 112–117.

[35] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Trans. on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, August 2004.

[36] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[37] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, October 1992.

[38] A. M. Erkmen, F. Yegenoglu, and H. E. Stephanou, "Entropy driven on-line control of autonomous robots," *Journal of Intelligent and Robotic Systems*, vol. 2, pp. 109–121, 1989.