

Externally Excited Coordination of Autonomous Formations

Iman Shames, Changbin Yu, Barış Fidan, Brian D. O. Anderson
Australian National University and National ICT Australia, Canberra, Australia

Abstract— Inspired by the recent “Robot Sheepdog Project” in the literature, in this paper a new method for coordination of multi-agent systems moving in a 2-dimensional region with the obstacles along a collision free trajectory is presented. A two level decentralized control scheme is proposed. At lower level, for the agents within the formation, “the sheep”, the only concern is to keep that of their formation shape intact and avoid inter-agent collision. At the higher level, an intelligent agent, “the sheepdog”, is responsible for determining a collision free trajectory for both the formation and itself, and then coordinating the formation motion along this computed path sequence. This sheepdog utilizes a repulsive vector (acting as a source of repulsive force) to drive the formation along the planned path. The effectiveness of the proposed scheme is demonstrated via simulation results in the presence of multiple obstacles.

Key words — Multiagent System, Formation Control, Path Planning, A* Search, Sheepdog

I. INTRODUCTION

In recent years, the topics of cooperative control, control of multi-agent formations and flocking have gained attention [1-5] in parallel with interest in their real-life applications such as search and rescue operations, surveillance, scientific data gathering, exploration, etc. Many projects focused on multi-robot path planning, formation generation, formation keeping, and formation control. These projects differ in factors such as the types of agent dynamics, control strategies, formation control tasks and the information flow between the agents.

While all of the methods developed in these studies are based on having intelligence in one or some of the agents within the flock (swarm, formation, etc.), there exists another set of recent works that suggest use of a single extra and more intelligent agent called a sheepdog, which is originally borrowed from the Robot Sheepdog Project [6]. In [6] a sheepdog robot has been used to control a flock of biological ducks, which already had natural flocking behaviors. In [7] an external stimulator agent, namely a sheepdog, is introduced to shepherd a flock of agents, with modelled herding behaviour, to their goal position. However, in both [6] and [7], the agents form loosely coupled flocks (swarms) rather than formations.

This work is supported by National ICT Australia, which is funded by the Australian Government’s Department of Communications, Information Technology and the Arts and the Australian Research Council through the Backing Australia’s Ability Initiative.

In this work, we study the use of a sheepdog agent for control of persistent formations of mobile agents. The idea is to use an intelligent agent, with path planning ability, to control many low level agents whose main concerns are to maintain their inter-agent distances and as a result maintain the formation intact and avoid inter-agent collision, depending on the requirements of the problem. By the word “intact” we mean that the formation shape is not altered during the course of movement. Admittedly, though we designate the agents of the formation as, “sheep”, often real sheep lack the capability of keeping themselves in a formation in the way that is commonly defined in the literature. Here, we prefer to follow the literature and exclusively use the formation to mean those multiagent systems which collectively preserve the formation shape during the continuous move [8-13].

The structure of this paper is as follows; Section 2 provides the problem definition. In Section 3 the proposed control framework is discussed. In Section 4 some simulation results are presented. The paper ends with concluding remarks in Section 5.

II. PROBLEM DEFINITION

A. Preliminaries

In this paper, we consider the problem of moving a formation from a certain initial location to another certain (desired or target) location in a region with obstacles without deforming the formation geometry, which is named a *cohesive motion problem* in [10]. One way to maintain formation shape during the course of motion is to keep fixed inter-agent distance between certain pairs of agents. Agent pairs for which the inter-agent distance is actively constrained to be constant can be thought of as being joined by bars with lengths enforcing the inter-agent distance constraints. The system can be therefore modelled by a graph where vertices represent point-like agents and inter-agent distance constraints are abstracted as edges [8, 9-11, 13].

The task of maintaining a prescribed distance between a nominated agent pair requires control action. One can conceive that the execution of the task could be the “responsibility” of both agents (symmetric control structure) or of one nominated agent of the pair (asymmetric control structure). Modelling using undirected graphs is appropriate in the former case. However in the latter case, it is important to recognise the distinction by assigning a direction to all edges in the

graph. A directed edge from vertex u to vertex v appears when agent u has the task of maintaining its distance constant from agent v , and agent v is unconstrained in its own motions with respect to the motion of u , i.e. it is “unconscious” of the task that agent u has to execute.

A formation F with an underlying graph $G_F(V_F, E_F)$ is called *rigid* if by explicitly maintaining distances between all the pairs of agents which are connected by an information link, i.e., whose representative vertices are connected by an edge in E_F , the distances between all other pairs of agents in F are consequentially held fixed as well, and hence F can move as a cohesive whole.

Typically the agent pairs in F whose inter-distances are explicitly maintained are the ones having information (i.e., sensing and communication) links in between, corresponding to the edges in the underlying graph G_F . Hence in (a geometric representation of) the underlying graph G_F , explicit maintenance of the distance between an agent pair (A_i, A_j) with an information link in between the two corresponds to keeping the length of the edge $(i, j) \in E_F$ constant.

For a formation F with asymmetric control structure, if each agent in F is able to satisfy all the constraints on it provided that all other agents within F are trying to satisfy their constraints (i.e., satisfy as many of their constraints as possible), then F is called *constraint-consistent*. A formation that is both rigid and constraint-consistent is called *persistent* [8].

Another notion found useful in characterizing persistent formations is the *number of degrees of freedom (DOF count)* of a vertex. In \mathfrak{R}^2 the vertices with zero out-degrees (the number of edges in G_F originating from vertex i) have 2 DOFs, the vertices with out-degree 1 have 1 DOF, the ones with out-degree 2 have 0 DOF.

For formal definitions of rigidity, persistence and constraint-consistence, and further information please refer to [8,9,11,12] and the references within.

For a persistent formation, it has been shown in [8, 9] that the sum of degrees of freedom (DOFs) of individual agents is at most 3 and for a minimally persistent formation, is exactly 3, which is the same as the DOF of a free rigid (non-vertex) object in \mathfrak{R}^2 (2 for translation and 1 for rotation). Based on the distribution of these 3 DOFs, minimally persistent formations can be divided into two categories: (i) Formations with the *leader-follower* structure where one agent has 2 DOF, another has 1 DOF and the rest have 0 DOF, and (ii) the *3-coleader* structure where three agents have 1 DOF and the rest have 0 DOF. In the *leader-follower* structure, the 2 DOF agent is called the *leader* and the 1 DOF is called *follower*. It's called the *first follower* if there is an information link between it and the leader. In the *3-coleader* structure, the 1-DOF agents are called the *coleaders*. In both structures the 0-DOF agents are called the (*ordinary*) *followers*. Figure 1 presents a 3-coleader structure with four agents as an example of class of persistent formations that are considered, in this case agents $A_1, A_2,$ and A_3 are coleaders with 1 DOF and A_4 is an ordinary follower with

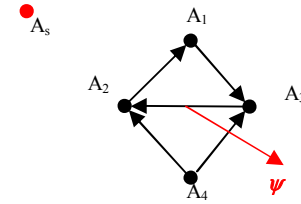


Figure 1. The formation $\{A_1, A_2, A_3, A_4\}$ and the sheepdog A_s studied in the simulations. Ψ is the vector responsible for keeping the formation away from the sheepdog

0-DOF. Here each agent tries to satisfy its distance constraint from that agent which the directed edge points to.

From [8], it can be seen that the underlying graph of a formation with the 3-coleader structure has at least one cycle. Due to the presence of a cycle, the motions of the three coleaders are cyclically dependent on each other [8] and hence the motion control for the formation requires a more implicit strategy.

For the main motives and obstacle-free version of the cohesive motion control problem together with a decentralized control design solution for 4-agent formations, the reader may refer to [10] and the references therein. A set of techniques to avoid collisions for the region-with-obstacle case have been introduced in [11]. This paper focuses on the region-with-obstacle case and collision avoidance as well, but differently from [11], an approach of external excitation using an extra agent is followed. This extra agent is used as a repulsive force source and is called a *sheepdog agent*, motivated by the recent “Sheepdog Robot Project” [6], where a “sheepdog robot” guides a group of biological ducks.

Next we give the formal definition of the problem together with descriptions of the agent characteristics and the formation model assumed.

B. Agents and Formation Model

Consider a formation F with N autonomous mobile agents A_1, \dots, A_N and another agent A_s , which is called the sheepdog agent. Assume each of the agents A_1, \dots, A_N, A_s are point agents whose positions on the 2-dimensional (xy) cartesian coordinate system at a particular time $t \geq 0$ are given by $p_1(t) = (x_1(t), y_1(t)), \dots, p_N(t) = (x_N(t), y_N(t)), p_s(t) = (x_s(t), y_s(t))$, respectively.

For each agent, a single integrator kinematic model has been used; hence agent motion is given by,

$$\dot{p}_i = v_i \quad i = 1, \dots, N, s \quad (1)$$

where v_i denotes the velocity of agent A_i .

Putting the sheepdog agent A_s aside, we assume the same formation model and agent kinematics and constraints as in [10]. Each sheep agent A_i ($i \in \{1, \dots, N\}$) knows its position, $p_i(t)$, and the position of any agent A_j , $p_j(t)$, which it follows at any time $t \geq 0$. In addition, the leader or co-leaders within

the formation know the position of the sheepdog, $p_s(t)$, provided that its position satisfies the following inequality:

$$\|p_s(t) - c_F(t)\| < 2R_F \quad (2)$$

where $c_F(t)$ is the position of the center of mass of the formation at time t , and R_F is the radius of the smallest circle which surrounds the formation.

The sensing range of each sheep agent is assumed to be large enough to guarantee that the sheepdog will be sensed if its position satisfies the previous Equation (2).

The sheepdog agent A_s on the other hand is assumed to have the latest information from the environment, i.e. a global map, including, its exact co-ordinates, $p_s(t)$ the position of obstacles, the position of the formation center of mass, $c_F(t)$, and the desired final destination of the formation center of mass, c_{FF} . No limitation has been assumed for the sensing range of the sheepdog agent.

C. Cohesive Motion Task

For the sake of simplicity, in this paper we do not deal with the task of securing the formation's orientation and the relative positions of the agents with respect to each other. Only the position of the center of mass of the formation is focused on. The cohesive motion task can formally be stated as follows:

Problem 1: Consider a persistent 2-dimensional formation F with $N \geq 3$ autonomous mobile agents A_1, \dots, A_N and an extra sheepdog agent A_s , whose models and characteristics are given in Section II.B. The control task is to design an individual controller for each agent A_i to generate v_i , in order to move F (the center of mass c_F), to a given final position, c_{FF} , cohesively, i.e. without deforming the shape or violating the distance constraints of F during motion, using the extra sheepdog agent, A_s , as a source of repelling force.

D. Design Constraints

The following design constraint is assumed in this paper as a tool for securing time-optimality: Any agent A_i will move at the constant maximum speed $v_i > 0$ at any time instant $t > 0$ unless it is impossible to do that at that particular instant t due to, e.g., initial and final zero-velocity constraints, absence of the sheepdog, etc.

III. PROPOSED CONTROL FRAMEWORK

We use a two level control scheme: In the higher level, the sheepdog agent A_s generates a map of the environment, plans a path for the formation F to follow in order to reach the desired destination without hitting the obstacles, and generates a sequence of waypoints for the center of mass of F , c_F . In the lower level, A_s iteratively positions itself such that the agents of F move in a way that c_F moves to the next waypoint, in each

step. Hence the upper level control design task is design of the guidance algorithms of the sheepdog agent for map generation, path planning, and waypoint generation; while the lower level control design task is the design of the individual motion controllers of the agents, A_1, \dots, A_N , A_s to generate v_1, \dots, v_N , v_s . Details of the control design for each of the agents follow.

A. Sheepdog Agent

The sheepdog agent A_s is required to generate the waypoints for the formation F to follow, as well as to move in a way that it repels F to track these waypoints. Each of the three subtasks of this guidance scheme are explained in the next three subsections, separately. In the last subsection of this section, the motion control law design for A_s is presented.

Grid Map Generation: In order to define various subregions in the area of interest, we first introduce the following notation:

$$\Omega(\bar{x}, \bar{y}, l, w) = \{(x, y) \mid \bar{x} \leq x \leq \bar{x} + l, \bar{y} \leq y \leq \bar{y} + w\} \quad (3)$$

denotes the rectangular region on the xy -plane where (\bar{x}, \bar{y}) are the coordinates of the lower left corner of the rectangular region, l is its length parallel to the x -axis, and w is its width parallel to the y -axis.

The environment map (area of interest) is represented by a rectangle

$$\Omega_E = \Omega(x_0, y_0, L, W) \quad (4)$$

with an overlaid grid structure. In this study grid resolution is assumed to be 0.1 m, i.e. each of x_0, y_0, L, W are assumed to be integer multiples of 0.1 m. The obstacle regions are represented by a sequence of M rectangles

$$\Omega_{O_i} = \Omega(x_{O_i}, y_{O_i}, l_{O_i}, w_{O_i}) \quad (5)$$

for $i = 1, 2, \dots, M$, fitting in with the grid structure, i.e. $x_{O_i}, y_{O_i}, l_{O_i}, w_{O_i}$ are integer multiples of 0.1 m for each i . It is assumed that $\Omega_{O_i} \subset \Omega_E$ for each i . Further denote

$$\Omega_O = \Omega_{O_1} \cup \dots \cup \Omega_{O_M} \quad (6)$$

In addition $\bar{\Omega}_{O_i}$ is defined as

$$\bar{\Omega}_{O_i} = \Omega(x_{O_i} - \varepsilon, y_{O_i} - \varepsilon, l_{O_i} + 2\varepsilon, w_{O_i} + 2\varepsilon) \quad i = 1, 2, \dots, M$$

in which ε is

$$\varepsilon \geq R_F \quad (7)$$

and

$$\bar{\Omega}_O = \bar{\Omega}_{O_1} \cup \dots \cup \bar{\Omega}_{O_M} \quad (8)$$

Assuming that the sheepdog agent A_s knows the locations of all of the obstacles within Ω_E perfectly, the grid-map generation task can be characterized as of the coloring of each of the $\frac{L}{0.1} \frac{W}{0.1}$ 0.1m \times 0.1m grids of Ω_E , i.e., each

$$\Omega_g[i, j] = \Omega(x_0 + 0.1i, y_0 + 0.1j, 0.1, 0.1) \quad (9)$$

where $i \in \{0, 1, \dots, (L/0.1) - 1\}$, $j \in \{0, 1, \dots, (W/0.1) - 1\}$, with one of three colors: White, black, gray. For each grid $\Omega_g[i, j]$ the following rule is used for coloring:

1. If $\Omega_g[i, j] \subset \Omega_o$ then color $\Omega_g[i, j]$ black
2. If $\Omega_g[i, j] \subset \overline{\Omega_o} \setminus \Omega_o$ then color $\Omega_g[i, j]$ grey
3. Otherwise color $\Omega_g[i, j]$ white.

The aim in the above coloring is as follows: The path will pass only through white grid squares, the agents can move through grey and white grid squares. With such a strategy it is guaranteed that neither the formation nor the sheepdog collides into obstacles.

Path Planning Using A Algorithm:* Here the aim is to find a path between the start position and the goal position, if any exists. This path is the path which the center of the mass of the formation should follow. The optimal path to be obtained using the A* algorithm will be expressed by a sequence of grid squares, and will be called the A*-optimal path sequence denoted by S_{A^*} .

The A* algorithm, is a graph search algorithm that finds a path from a given initial grid square to a given goal grid square. It was introduced by Nilsson [14], is an effective heuristic improvement of Dijkstra's algorithm[15] and provides a better average performance, with respect to searched nodes, when one only needs the optimum path, with respect to traveled distance, between two nodes in a directed graph with non-negative weights [9].

Here the maximum number of movements from each grid square is restricted to eight directions, or less when the adjacent nodes are placed in a grey region or are the borders of the environment. See Figure 2 for the possible grid squares to be chosen for being the next element in the path sequence.

In each step, a weight function f is used to choose the best move. It is defined as $f = g + h$, where g is the actual cost (distance traveled) from the starting grid square, $(x(0), y(0))$, to the current grid square, and h is the heuristic cost from the current position to the goal.

Here h is defined as the Euclidean distance from the current position to the goal position. Thus h for grid square n , is defined as,

$$h(n) = \sqrt{(x(n) - x_g)^2 + (y(n) - y_g)^2} \quad (10)$$

where $(x(n), y(n))$ are the coordinates of, $S_{A^*}[n]$ the center of the n th element of S_{A^*} , and (x_g, y_g) are the coordinates of the center of the goal square grid.

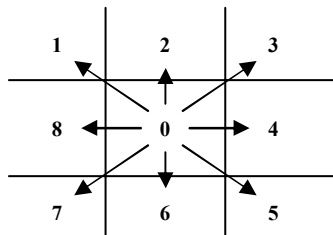


Figure 2. Eight possible selections for the next point in path sequence

Waypoint Generation: To make a suitable path for our system to follow, a set of waypoints are generated from S_{A^*} obtained in the previous step. In order to do so, an algorithm similar to the one presented in [9] is used. The algorithm is presented in the following, where it is assumed that the length of S_{A^*} is N .

1. Construct a reverse waypoint list $W_R[.]$ and set $W_R[1] := S_{A^*}[N]$.
2. set $k := N - 1$ and $l := 2$.
3. Do until $k = 1$;
 - a. Check whether there is a non-obstructed path between $W_R[l-1]$ and $S_{A^*}[k]$.
 - b. If no, set $W_R[l] := S_{A^*}[k+1]$ and $l := l + 1$.
 - c. Set $k := k + 1$ and go to 3
4. Reverse the order of $W_R[.]$ and store it as the waypoint list, $W[.]$.

Control Laws for the Sheepdog: In order to control the sheepdog, a combination of two laws below, which are obtained from the *virtual vector field* approach described in [12]. The first law is used for positioning the sheepdog agent when the formation has reached its waypoint and is waiting to be guided towards its next waypoint and the second law controls the agent's behaviour when it is repelling the formation towards the next waypoint.

Note that the first and the second control laws are applied in an alternating fashion, i.e. the first control law is applied in the intervals, $[0, t_1)$, $[t_2, t_3)$, \dots , and the second control law is applied in $[t_1, t_2)$, $[t_3, t_4)$, \dots , where $0 < t_1 < t_2 < \dots$

The first law is given by,

$$v_s(t) = k\alpha(t) \quad (11)$$

where $k > 0$ is a design constant and

$$\alpha(t) = \frac{(\mu(t) + \lambda(t))}{\|\mu(t) + \lambda(t)\|} \quad (12)$$

$$\lambda(t) = \begin{cases} 0 & \text{if } \|p_s - c_F(t)\| \geq 2R \\ -\frac{(c_F(t) - p_s)}{\|c_F(t) - p_s\|} & \text{otherwise} \end{cases} \quad (13)$$

$$\mu(t) = \frac{\bar{p}(t) - p_s(t)}{\|\bar{p}(t) - p_s(t)\|} \quad (14)$$

Here λ keeps the sheepdog away from the formation in order to prevent it from deforming the formation by exerting too much repulsion force on the formation while \bar{p} is the suitable position for the sheepdog in order to guide the formation to the next waypoint and is calculated by the following equation:

$$\bar{p} = 2R \frac{c_F(t) - p_d}{\|c_F(t) - p_d\|} + c_F(t) \quad (15)$$

where p_d is the next waypoint.

The above points and vectors are illustrated in Figure 3. The second law is given by

$$v_s(t) = k\alpha'(t) \quad (16)$$

where

$$\alpha'(t) = \frac{(\mu'(t) + \xi(t) + \lambda(t))}{\|\mu'(t) + \xi(t) + \lambda(t)\|} \quad (17)$$

in which we have:

$$\mu'(t) = -\frac{p_d(t) - p_s(t)}{\|p_d(t) - p_s(t)\|} \quad (18)$$

$$\xi(t) = \bar{\eta} \frac{(c_F(t) - p_s(t))}{\|c_F(t) - p_s(t)\|} \quad (19)$$

where $\bar{\eta} > 1$ is a design coefficient. μ' keeps the sheepdog away from the goal position, ξ attracts the sheepdog towards the formation. When the sheepdog deviates from the straight line connecting c_F and p_d interaction of μ' and ξ results in driving the sheepdog back to this straight line.

The vectors, μ' and ξ , are illustrated in Figure 4.

B. Control laws for the Regular (Sheep) Agents

In this section the control laws to control agents A_1, \dots, A_N of F , are briefly presented. These rules are the modified versions of the control laws presented in [10, 11]. These control laws are composed of two separate motion vector types of an agent, (i) to maintain a distance constraint with each of the agents it follows and (ii) to move towards a final destination.

The rules to control the coleaders in the formation, e.g. A_1, A_2, A_3 in Figure 1 are given by,

$$v_i = \beta_i(t)\gamma_i(t)\tilde{\delta} + \phi(t)\sqrt{k^2 + \beta_i(t)}\delta_{ji}^\perp \quad (20)$$

in which v_i is the velocity of i -th coleader, and we have:

$$\tilde{\delta}(t) = \frac{(\bar{\delta}_{ji}(t) + \psi(t))}{\|\bar{\delta}_{ji}(t) + \psi(t)\|} \quad (21)$$

$$\bar{\delta}_{ji}(t) = (p_j(t) - p_i(t)) / \|p_j(t) - p_i(t)\| \quad (22)$$

$$\delta_{ji}^\perp(t) = (-\delta_{jy}(t), \delta_{jx}(t)) \quad (23)$$

$$\bar{\delta}_{ji}^\perp(t) = \psi(t) \quad (24)$$

where, ψ is the vector responsible for keeping the formation away from the sheepdog and as a result driving the formation to its desired position, and it is defined as following:

$$\psi(t) = -\eta\lambda(t) \quad (25)$$

whit $\eta \in [0,1]$ being a design coefficient.

For the motives behind the above rules for controlling ordinary followers the reader may refer to [10].

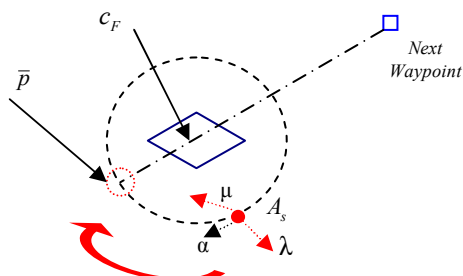


Figure 3. The vectors and points in the first control laws of sheepdog

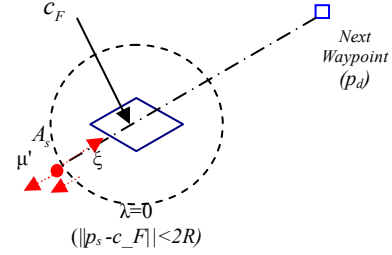


Figure 4. The vectors and points used to define second set of rules for control of sheepdog

Remark: Instead of knowing the whole map of the region of interest, the sheepdog may generate local maps periodically, e.g. every 10 seconds. Such an approach will relax the assumption that the sheepdog knows the global map.

IV. SIMULATION RESULTS

In this section a set of various environment settings (with $\Omega_E = \Omega(0,0,15,15)$) are chosen to observe the performance of the proposed control method. A 3-coleader formation as in Figure 1 is studied, it is a 3-coleader structure and due to the complexity of the control of coleader formations has been chosen for test purposes.

In our simulation model, each obstacle is considered to be a rectangle, or a shape consisting of several rectangles. The grid coloring of the map is as described in Section III.A. Another assumption that has been made to run the simulation is that the speed of the sheepdog and the formation is considered to have a constant value of 1m/sec*.

In the first setting (Figure 5) the formation should travel from the lower left corner of the arena to the upper right corner, each waypoint along the way being designated by a square; these waypoints are calculated as described in III.A.

The trajectory plot shows c_F and A_s and the desired path of the formation in Figure 5. It took 79.2 seconds.

The second simulation (Figure 6) aims to test the ability of the formation to find and follow a clear path through a corridor in order to reach its final destination. There is a very narrow clear space between the grey regions of the walls. Figure 6 shows the desired path, the actual path and the sheepdog path. In the second set up, the sheepdog finds a collision free path through the corridor and herds the formation of sheep into its final destination behind the fence. As a result, the total path traveled by the sheepdog increases, as can be seen from the blue lines. However, this scenario took 46.1 seconds.

V. CONCLUSION

In this paper, we have introduced a new approach for control of a persistent/rigid formation of mobile agents by

* This assumption can be relaxed for the sheepdog, but for the sake of simplicity in simulations it has been assumed so.

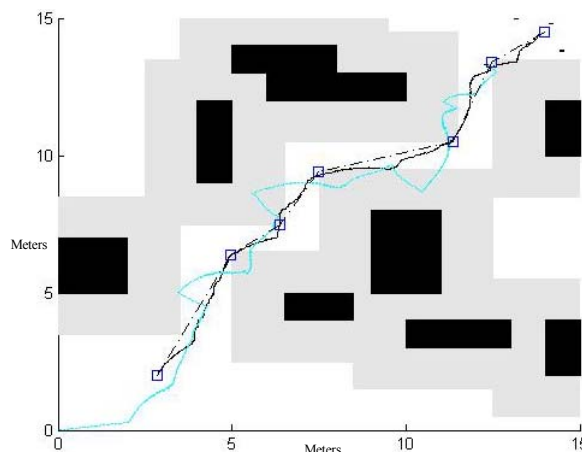


Figure 5. The black area denotes the obstacles, the grey area denotes the obstacle regions, the dashed line is the desired path, the squares are the waypoints, the black line is the formation trajectory (c_F) and the blue line is the sheepdog (A_s) trajectory

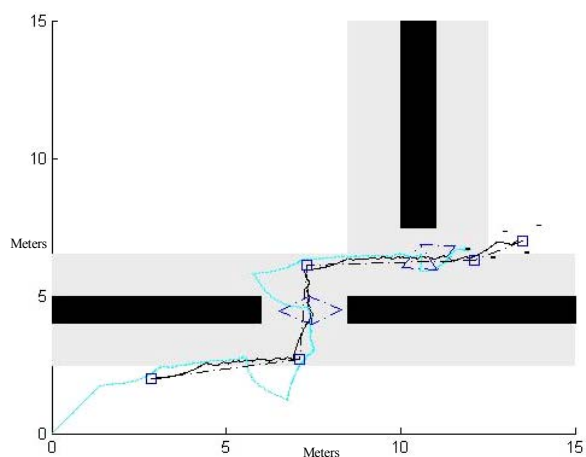


Figure 6. The black area denotes the obstacles, the grey area denotes the obstacle regions, the dashed line is the desired path, the squares are the waypoints, the black line is the formation trajectory (c_F) and the blue line is the sheepdog (A_s) trajectory.

using an extra intelligent agent to guide the formation of agents. The proposed approach utilizes the property that if there are a sufficient number of distances maintained between pairs of agents of a formation, so that the entire formation is kept as a rigid/persistent whole during any cohesive motion, then inter-agent collisions are avoided. The distortions occur in the shape of the formation, are kept negligibly small using this approach. Observing this fact, the problem reduces to motion control of the formation along with a path computed by another highly intelligent agent, in this case the sheepdog. This problem is tackled with the use of an extra intelligent agent, the sheepdog, which repels the formation towards the precalculated waypoints and eventually to the desired destination. A simple simulation model assuming point agents with velocity integrator dynamics and static rectangular obstacles is investigated to demonstrate the effectiveness of the proposed schemes, for cases with more complicated agent dynamics.

In real world applications, the scheme clearly has to be implemented to be robust to noise, and finite sensing radii

for all agents. Scalability and computational complexity are among other aspects to be considered.

Further extensions of this study may include use of several sheepdogs, implementation of this control method for controlling formations in 3-dimensional space, and optimization of the current approach from the aspect of the sheepdog trajectory and the formation trajectory.

REFERENCES

- [1] J. P. Desai, J. Ostrowski, and V. Kumar, "Controlling Formations of Multiple Mobile Robots", Proc. Of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, May 1998.
- [2] H. G. Tanner, G. J. Pappas and V. Kumar, "Leader-to-Formation Stability", *IEEE Transactions on Robotics and Automation*, vol 20, no 3, June 2004, pp. 433-455.
- [3] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Tra. On Automatic Control*, vol. 49, no. 9, pp. 520 - 1533, September 2004.
- [4] V. Gazi and K.M. Passino, "Stability analysis of swarms," *IEEE Tra. on Automatic Control*, vol. 48, no. 4, pp. 692-697, April 2003.
- [5] J. Baillieul and A. Suri, "Information patterns and hedging Brockett's theorem in controlling vehicle formations," in Proc. of the 42nd IEEE Conf. on Decision and Control, vol. 1, pp. 556-563, December 2003.
- [6] R. T., Vaughan, N. Sumpter, J. Henderson, A. Frost, and S. Cameron, "Experiments in automatic flock control", *J. Robot. and Autonom. Sys.* 31:109-117, 2000.
- [7] O. B. Bayazit, J.-M. Lien, and Nancy M. Amato. "Better group behaviors in complex environments using global roadmaps", In Proceedings of the Artificial Life (ALIFE), Dec 2002.
- [8] C. Yu, J. Hendrickx, B. Fidan, B. Anderson, and V. Blondel, "Three and higher dimensional autonomous formations: Rigidity, persistence and structural persistence," *Automatica*, Volume 43, Issue 3, March 2007, Pages 387-402
- [9] Y. Hao, S. K. Agrawal "Planning and control of UGV formations in a dynamic environment: A practical framework with experiments", *Robotics and Autonomous Systems* 51: Pages 101-110, 2005.
- [10] S. Sandeep, B. Fidan, and C. Yu, "Decentralized cohesive motion control of multi-agent formations," in Proc. 14th Mediterranean Conference on Control and Automation, June 2006.
- [11] C. Yu, B. Fidan, I. Shames, S. Sandeep, B.D.O. Anderson "Collision Free Coordination of Autonomous Multiagent System", to appear in European Control Conference, Athens, July 2007.
- [12] S. Drake, K. Brown, J. Frazier and A. Finn, "Autonomous Control of Multiple UAVs for the Passive Location of Radars", in Proc. 2nd Int. Conf. on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp.403-409, December 2005.
- [13] B.D.O. Anderson, C. Yu, B. Fidan & J. M. Hendrickx, "Control and Information Architectures for Formations", Proc. Of the IEEE conference on Control Applications (in Joint CCA/CACSD/ISIC), pp.1127-1138, Munich, Oct 2006
- [14] N.J. Nilsson, *Principles of Artificial Intelligence*, Springer-Verlag, Berlin, pp. 72-88, 1982.
- [15] E. W. Dijkstra, "A note on two problems in connexion with graphs.", *Numerische Mathematik*. 1 (1959), S. 269-271