2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Mobile Dynamically Reformable Formations for Efficient Flocking Behavior in Complex Environments

Turker Sahin and Erkan Zergeroglu

Abstract— **In this work inspired by flocking of birds or fish communities traveling together in the nature, we have developed a novel dynamically reformable mobile formation algorithm for the navigation of wheeled mobile robot (WMR) groups operating in complex and/or obstacle dense environments. The proposed method is formed via the combination of simple and computationally efficient tools such as (i) a mobile network of a small number of WMR sensors for detecting obstacles; (ii) Cardinal Cubic splines or Least Squares fits for modeling the formation boundaries based on this small network; and (iii) reference frames to ensure uniform spacing and velocity profiles along the ensemble. By these components a simple geometrical formation is developed for real time flock path planning of relatively large groups of small agents. To our best knowledge the proposed approach is novel and its effectiveness is verified by simulations in complex environments.**

I. INTRODUCTION

In complex robotic tasks, instead of using a couple of dexterous, highly functional robots with extensive sensor and actuators that will require complex planning and control algorithms and intelligence to perform the tasks, it might be more appropriate to use large numbers of inexpensive robots equipped with elementary level intelligence. Some similar tasks involve search and rescue, coverage, mapping in vast remote operational spaces including outer space and under sea. In this respect research on gaining insight into behavior of bird or fish species in nature is critical, as they form highly effective swarms improving the performance of single individuals in tasks like hunting, defense, and migration to long distances. It is believed that in these groups autonomous agents do not follow commands from a leader, or some global plan [1]. Instead they re-adjust their behavior by factors such as low level reasoning based on their nearest neighbors. Swarm or flock methods in robotics aim to gain insight into collective and cooperative tasks by imitation of these natural swarms. In this respect, Reynolds introduced three heuristic rules termed “cohesion”, “separation” and “alignment” that led to creation of the first computer animation of flocking [2]. Among these rules cohesion is on flock centering (ie. attempt to stay close to nearby flockmates); separation is on collision avoidance with nearest neighbors; and alignment is also refereed to as attempting to match velocity with nearby flockmates. These three rules form a framework for the navigation of robot swarms.

Many alternative methods have been proposed for forming swarm/flock behavior as closely as possible, particularly for modeling multi-agent collective motion problems. In [3], [4] particle swarm optimization is applied for generating accurate flock patterns. In [5], a simple model of swarming in the presence of an attractant/repellent profile is presented for simulating stable social foraging of swarms in a profile of nutrients. There are also potential function based swarm planners [6], [7]. In [8], authors present alternative swarm approach via guidance of a high number of simple agents by a smaller number of more sophisticated robots similar to the shepherding behavior in sheep flocks. These robot guides may aid future transportation and leisure activities in air and sea. Some methods which integrate implicit polynomials (IPs) with swarm applications have also been presented [9], [10], [11]. As accurate polynomial fitting is computationally intensive for real-time swarm applications, these methods are limited to IP based mapping nearly stationary shapes such as undersea structures or oil spills. Swarm methods can also be applied to other fields than robotics such as computational graphics/geometry [12] and medicine [13].

The flock planner we propose is based on the coordination of a mobile dynamically reconfigurable formation and the single agent path planner we have introduced in [14]. The formation as a whole uses a small number of ensemble outer sensors as a network to detect the obstacles and concurrently reform its outer shape during navigation. It uses the acquired data to guide the agents of formation away from the detected obstacles. The agents are constrained to within the morphed formation by fitting discrete boundaries via Cardinal Cubic (CC) splines or alternatively continuous borders by Least Squares IP functions. Finally for even agent distribution and velocity alignment a readjustable reference frame moving with the formation is introduced. The agents avoid collisions with their neighbors or any small obstacles that may enter the formation zone undetected by the sensor network by an efficient agent planner. To the best of our knowledge the proposed method is novel. The most notable benefits of the method are guidance of agents by smoothly varying outer boundaries, and effective agent margins by the attached mobile reference frame. These propertied result in improved collective behavior and reduced entrapment/collision rates in complex environments. Moreover the proposed formation has a computationally-simple approach to modeling formation borders despite utilizing a point set based approach. Hence, real-time simulations of relatively large flocks (of 15 agents) is achievable with entry level PCs.

The rest of the paper is organized in the following manner: Section II presents the problem statement, while Section

The authors are with the Department of Computer Engineering, Gebze Institute of Technology, PK. 141, 41400 Gebze/Kocaeli-Turkey. E-mail: {htsahin, ezerger}@bilmuh.gyte.edu.tr

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

III revises a fixed reference frame based flock planner. In Section IV, the mobile formation based planner is explained in detail with supporting simulations presented in Section V. Finally concluding remarks are summarized in Section VI.

II. The Problem Formulation

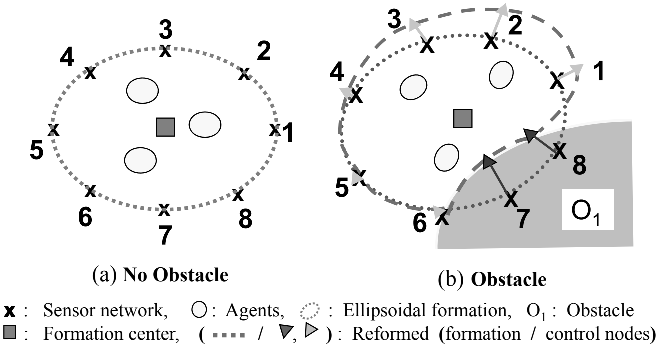
The main objective of this work is to design a mobile dynamically re-morphable formation, which can collectively steer a group of WMRs similar to flocking behavior in environments of highly complex obstacle distribution. Effective real-time operation with simple hardware and computational power is also another critical aspect. Hence the preferred approach is based on integration of various computationally simple algorithms. These components use position data from some agent sensors (in the form of a sensor network) to generate mobile formation borders and reference frames for satisfying the three main rules of flocking (ie. cohesion, separation and alignment) for a group of WMRs. The WMRs also use the an efficient agent planner for relatively minor path rearrangements.

III. Fixed Reference Frame Path Planner

The flock planner in [14] uses a simple yet effective reactive agent planner with a fixed reference frame (FRF) for navigation of a group of unicycle type WMRs in a loosely bound geometrical formation. The agents are normally directed towards their reference points in the formation. However, when they encounter obstacles, the avoidance mechanism of their individual planners are activated and temporarily divert them from the formation. After the obstacle avoidance is over, the agents are again directed to their reference locations in a smooth path. The method has a number of advantages such as enabling real-time path generation for relatively crowded agent groups, not requiring inter-agent communications during navigation and improved concave obstacle avoidance by adjusting two parameters. The main disadvantage of FRF planners is they cannot achieve collective navigation in complex environments such as tunnels. Many agents inevitably get marooned from the reference frame. The reactive agent obstacle avoidance performance falls as obstacle sizes increase with respect to the sensing range. This may cause some WMRs to be trapped. Even worse if an agent reverses direction in a tunnel, this may severely disrupt the trailing traffic. Moreover using expensive longer range sensors is not a viable option as this conflicts with the aim of swarm planning.

IV. Mobile Dynamically Reconfigurable Reference Frame Formation

In this work we propose a mobile flock planner, called "**M**obile **D**ynamically **R**econfigurable **R**eference **F**rame" (MDRRF) formations to address the collectivity limitation of FRF planners. The method uses obstacle data from a small network of range sensors to adopt the three flocking rules of Reynolds [2]. The sensor data is used to (i) maneuver the overall formation away from the detected obstacles and (ii) simultaneously reform the shape of the formation boundaries according to encountered walls. These components address the "cohesion rule" and reduce agent defragmentation. A mobile reformable reference frame is integrated into the MDRRF planner. The frame provides reasonably spaced reference points in environments with/without obstacles, thereby improving agent distribution and velocity profiles ("separation" and "alignment" rules).


3
4
2
1
5
6
7
8
3
2
4
1
5
8
6
7
O1
x : Sensor network, ○: Agents, ◌ : Ellipsoidal formation, O1 : Obstacle
■ : Formation center, (▪▪▪▪ / ▼, ▷) : Reformed (formation / control nodes)

(a) **No Obstacle** (b) **Obstacle**

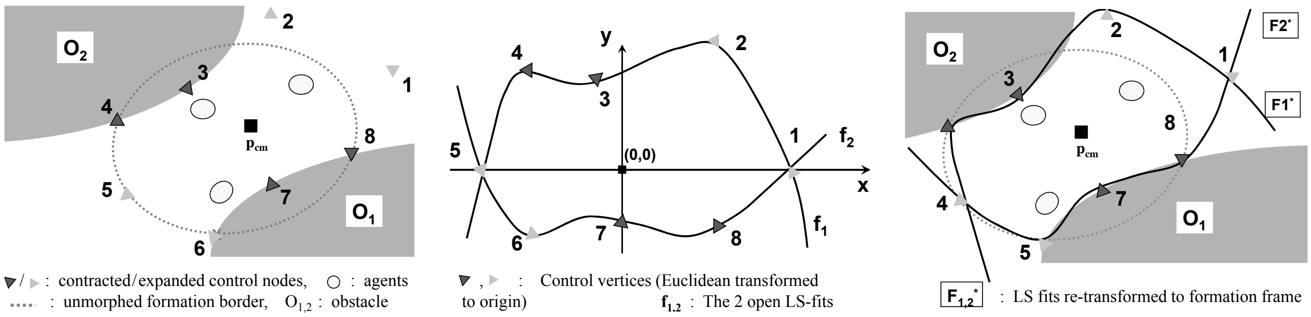
Fig. 1. Morphing of an MDRRF formation.

A. Mobile Sensor Network

The mobile sensor network $P_s = \{p_{s_i}\}_{i=1}^{8}$, is a set of 8 sensors from outer agents of formation as depicted in Figure 1. The main function of this network is to detect encountered obstacles and provide a measure of their impact on the formation boundaries. The flock planner will then use this data to re-adjust the direction and shape of the formation so that the agents are guided away from large and possibly concave obstacles.

In an obstacle free environment the MDRRF formation is assumed to start as an ellipsoidal zone and is steered non-holonomically for the rest of the simulation. The procedure is outlined in Figure 1(a). For re-adjusting the formation behavior nearby obstacles, the planner requires both the intrusion range of the obstacles into the formation sensing zone and an estimate of their size/position. Obstacle size and position estimates are computed by an efficient algorithm without resorting to complex mappings and these estimates are used to provide a steer force for maneuvering the overall ensemble similarly to the planner in [15].

The planner also reforms the shape of the formation outer boundary near obstacles to limit agent-obstacle interaction. A set of control nodes $P_c = \{p_{c_i}\}_{i=1}^{8}$ is defined to be one to one and onto the set of sensor range locations P_s when the formation is not influenced by obstacles. However when an obstacle is detected, the positions of the associated control nodes shrink towards the formation center to reflect the termination of formation boundary at the obstacle border (see the sensors 7 and 8 in Figure 1(b) shrunk by the dark gray arrows from the sensing range). At every time instant of the contraction periods, the control nodes corresponding with the free sensors expand to normalize the formation area for inter-collision free guidance of agents away from blocks (see the expansions from sensing range denoted by light gray arrows in Figure 1(b)).


O2
2
3
1
4
8
pcm
5
7
O1
6
y
4
2
3
5
(0,0)
1
f2
x
6
7
8
f1
O2
2
F2*
1
3
F1*
8
pcm
7
4
O1
5
▼/▶ : contracted/expanded control nodes, ○ : agents
····· : unmorphed formation border, $O_{1,2}$: obstacle
▼ , ▶ : Control vertices (Euclidean transformed to origin)
$f_{1,2}$: The 2 open LS-fits
$F_{1,2}^*$: LS fits re-transformed to formation frame

(a) Morphing of control vertices (b) Euclidean transformation to origin and LS fits (c) Re-transformation to the formation pose

Fig. 2. Continuous boundary modeling by 3 steps LS-fitting steps

The details of the mathematical model summarizing the above procedure is as follows. Let an obstacle detection occur by a number of n sensors at time $t = t_0$ [s] and let the obstruction continue until $t = t_f$ [s]. Let the index set of the all control vertices and the control vertices under obstacle effect denoted by $C_i = \{i\}_{i=1}^{8}$ and $O_i \in \mathbb{N}^n$, respectively. Thus the index set of the free control nodes is $F_i = C_i \setminus O_i \in \mathbb{N}^{8-n}$. The obstructed sensor nodes are shrunk towards the formation center as depicted in Figure 1(b) via the following the iterative model:

$$p_{c_i}[t + \triangle t] = p_{cm} + k_C(p_{c_i}[t] - p_{cm}),\ \forall i \in O_i. \quad (1)$$

In (1) iteration continues throughout the time period $t \in [t_0, t_f]$. $p_{c_i}[t]$ and $p_{c_i}[t+\triangle t]$ represent the current position of the contracting nodes and their next sample update. $\triangle t$ is the sample time, $k_C \in \mathbb{R}^+$ is the contraction coefficient and p_{cm} is the formation center. Contraction reduces the area of the formation in the direction of obstacles. The planner should compensate for this lost area to avoid agent deadlocks. This is carried out by expanding the free control nodes during $t \in [t_0, t_f]$ as follows:

$$p_{c_i}[t + \triangle t] = p_{cm} + k_E(p_{c_i}[t] - p_{cm})w_{E_i},\ \forall i \in F_i. \quad (2)$$

Here $p_{c_i}[t]$ and $p_{c_i}[t+\triangle t]$ are the current and next updates of the expanded nodes, and p_{cm}, $\triangle t$ are as previously defined. The expansion is governed by the expansion coefficient k_E and control node weights w_{E_i}. k_E is a measure that compensates for the lost area from contraction and is related to k_C by $k_E = e\,n\,k_C/(8-n),\ \ e \leq 1$. w_{E_i} is a weight for distribution of the expansion among the free sensor nodes. These weights emphasize direction of formation path so that:

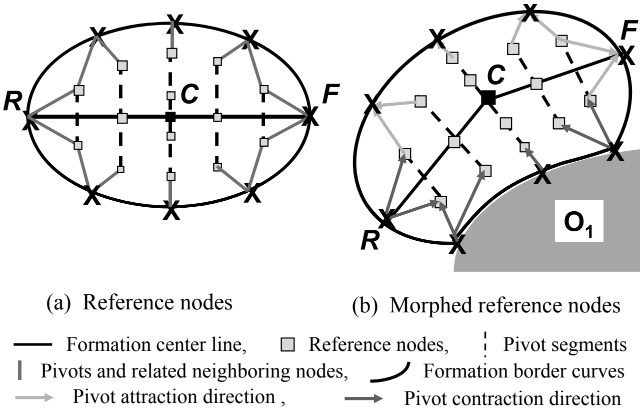
$$w_{E_i} = \left(1 + \frac{w_{sens_i}}{w_{FT}}\right),\ \ \forall i \in F_i, \quad (3)$$

where w_{FT} is the overall weight of the expanding control vertices and w_{sens_i} is weight of the i^{th} expanded node. The indexing of w_{sens_i} is according to the order in Figure 1 with:

$$w_{sens_i} = \{4, 3, 2, 1, 0, 1, 2, 3\},\ \ \forall i \in C_i. \quad (4)$$

B. Synthesis of the Formation Outer Boundary

The sensor network and morphing of the corresponding control nodes serve as a rough framework for reforming the formation shape near obstacles. But less sparse bounding mechanisms are necessary for keeping the agents together when the formation is affected by obstacles. Two alternative boundary models are applied for this purpose: (i) Cardinal Cubic (CC)-splines, and (ii) Least Squares (LS)-fits. CC-splines are 3^{rd} degree parametric curves. They are represented by piecewise cubic polynomials in the form $\sigma(s) = a_0 + a_1 s + a_2 s^2 + a_3 s^3$ with variation of their parameter in the range $s \in [0, 1)$. They interpolate through their control points called knots and satisfy C^1 continuity. Thus they form a discrete and smooth boundary of interpolated points between the formation sensor network knots by a selection of equidistant intermediate s values in the interval $[0, 1)$. The formula for CC-spline interpolation between the control nodes $p_{c_{i-1}}$ and p_{c_i} is:


X
X
X
R
C
F
X
X
X
X
X
F
C
X
X
X
R
X
O1
Formation center line, ▢ Reference nodes, ¦ Pivot segments
Pivots and related neighboring nodes, Formation border curves
Pivot attraction direction , Pivot contraction direction

(a) Reference nodes (b) Morphed reference nodes

Fig. 3. Readjustment of reference frame with formation morphing.

$$\sigma(s) = \begin{bmatrix} 1 \\ s \\ s^2 \\ s^3 \end{bmatrix}^T \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & \tau - 2 & \tau \end{bmatrix} \begin{bmatrix} p_{c_{i-2}} \\ p_{c_{i-1}} \\ p_{c_i} \\ p_{c_{i+1}} \end{bmatrix} \quad (5)$$

where $i \in C_i$ denotes the indices of knots set; and $\tau \in [0, 1]$ is the tangent of the spline at the knots. Lower τ values are preferable for providing smooth boundary pieces $\sigma(s)$, in between every neighboring knot pair $p_{c_{i-1}}$ and p_{c_i}.

The bounding mechanism of discrete CC-splines is similar to electrostatic potentials. The fitted set of points can be assigned same charges as the enclosed agents. Thus the spline border can be used as a repelling potential, ie. a set of same signed charges with the agents to redirect any nearby WMRs towards the formation center. Specifically, a bounding force $F_b(t) = [F_b(x), F_b(y)]^T$, is superposed to the agent steer force $F_s(t)$ to confine the agent to the formation boundary if the distance of a robot to the formation border L_a, is less than a threshold δL:

$$F_b(x) = K_r(p_{cm} - p_{ae})/(\|p_{cm} - p_{ae}\|L_a), \quad L_a < \delta L \quad (6)$$

Here (6) p_{cm} is the formation center, p_{ae} is the agent steer point and $K_r \in \mathbb{R}^+$ is a positive scalar constant. This mechanism only disrupts the paths of agents in vicinity of formation border, thereby improving agent collectivity.

CC-splines based formation borders have a number of drawbacks: (i) They are interpolating functions, thus their output is a discrete set of positions. The number of elements in this set cannot be selected arbitrarily high for computational cost concerns. As the formation is morphed, there may occasionally be sparser regions in the set, allowing the agents to leave the formation. (ii) Another difficulty of the CC-splines is their boundary force $F_b(t)$ integrated with the agent obstacle avoidance force mechanism via superposition. Hence the selection of the constant K_r in equation (6) involves heuristics. This may arise problems in obstacle and agent dense scenarios. We have used IP curves which model boundaries by continuous functions to address these weaknesses models. The general formula for IP curve models (also refereed to as algebraic curves) is

$$f(x, y) = \sum_{i,j \geq 0, i+j=n} a_{ij} x^i y^j \quad (7)$$

where a_{ij} and n are the coefficients and the degree of the curve, respectively; and i and j are the powers of x and y terms. IP curves are defined at the zero-set of these functions such that $f(x, y) = 0$. Algebraic curves have a computationally efficient point-boundary classification mechanism by evaluation of their polynomials. For example to determine the position of an agent with respect to an algebraic curve based formation boundary, we only evaluate the IP at the agent sensor position values $p_c = (x_c, y_c)$. Thus from (7), we can conclude that the sensor point is outside the curve if $f(x_c, y_c) > 0$, on the curve if $f(x_c, y_c) = 0$, and inside the curve if $f(x_c, y_c) < 0$. As a result any IP based outer boundary can be directly integrated into the obstacle avoidance mechanism of the employed agent planner as a virtual obstacle similarly to the approach in [15]. This is an effective formation border generation mechanism if the IP fits can be applied in real-time.

However, directly attempting to fit closed-bounded 2-D algebraic curve fits has numerous drawbacks. IP fits have the effect of smoothing their data point sets, thus poor shape representation is common when lower degree IPs are utilized as basis sets [16]. A more important and common problem of general IP methods is curve instability (ie. open curves with branches leading to infinity rather than closed bounded fits), which renders point classification useless. In this paper an alternative approach is used for outer bounding the formations by IP fits to achieve improved stability and point classification: The set of control nodes P_c, is Euclidean transformed to origin with $T : P_c \to P_t$ to form a normalized set of control nodes P_t (see the transition from Figure 2(a) to 2(b)). The transformation is so that the front and rear control vertices intersect with the x-axis as depicted in Figure 2(b); however, the formation center p_{cm} need not exactly coincide with the origin. The new set of 8 control nodes P_t is subdivided to two data subsets of 5 points each corresponding to the upper and lower half planes bounded by x-axis. These two subsets are fit by two separate 4^{th} degree open IPs. These two fits are then re-transformed to the formation pose as in Figure 2(c) thereby forming the outer boundary of the robot ensemble. This approach has numerous advantages: (i) the resulting curves intersect each other near the very front and rear nodes of control points set P_c, thereby offering similarly efficient point classification as closed bounded IP fits; (ii) utilization of two IPs have higher redundancy for accurate shape representation; (iii) the method uses a small number of points and is suitable for real-time operation.

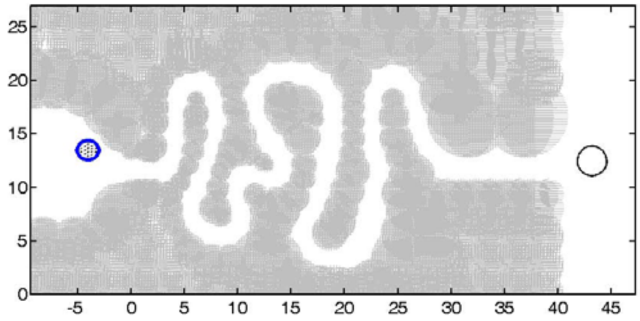


Fig. 4. Overview of the complex tunnel simulation environment.

Least Squares IP fitting method is used for being more efficient than other IP techniques. The utilized polynomial basis functions are in the form

$$y = f(x) = a_0 + a_1 x + \ldots + a_4 x^4 = \mathbf{a}^T \cdot \mathbf{x}. \quad (8)$$

Here the LS curve parameter vector $\mathbf{a} \in \mathbb{R}^5$ is estimated by:

$$\mathbf{a} = (\chi^T \chi)^{-1} \chi^T \upsilon, \quad (9)$$

where χ and υ are the VanDerMonde matrix of x-axis and vector of y-axis components of transformed data subsets, in the form

$$\chi = \begin{bmatrix} 1 & x_{t_1} & x_{t_1}^2 & x_{t_1}^3 & x_{t_1}^4 \\ & & \cdots & & \\ 1 & x_{t_5} & x_{t_5}^2 & x_{t_5}^3 & x_{t_5}^4 \end{bmatrix}, \quad \upsilon = \begin{bmatrix} y_{t_1} \\ \cdots \\ Y_{t_5} \end{bmatrix}. \quad (10)$$

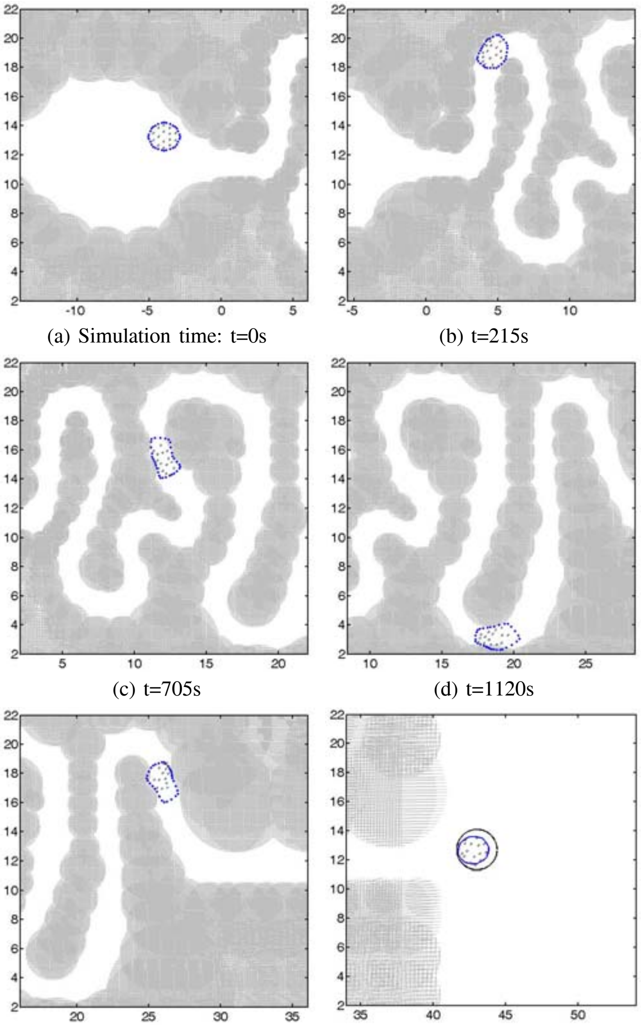
The proposed IP fit method is not interpolation, but approximation based. Hence the resulting fits may be inaccurate if some data set points are very near each other (ie. near singularity condition for the $\chi^T \chi$ matrix in (9)). This problem can be cured by reducing the e coefficient of the expansion ration k_E in equation (2) or by modification of (9) via ridge regression regularization in the form [16]:

$$\mathbf{a}_{RR} = \left(\chi^T \chi + \epsilon I_{5\times5}\right)^{-1} \chi^T \upsilon, \quad (11)$$

with $I_{5\times5}$, being the 5×5 identity matrix, and ϵ is a scaling factor in range $0.001 - 0.01$.

C. Reference Frame

For even inter-agent spacing and smooth velocity profiles in the ensemble, a mobile reference frame is applied (see Figure 3). To avoid computationally costly reference point

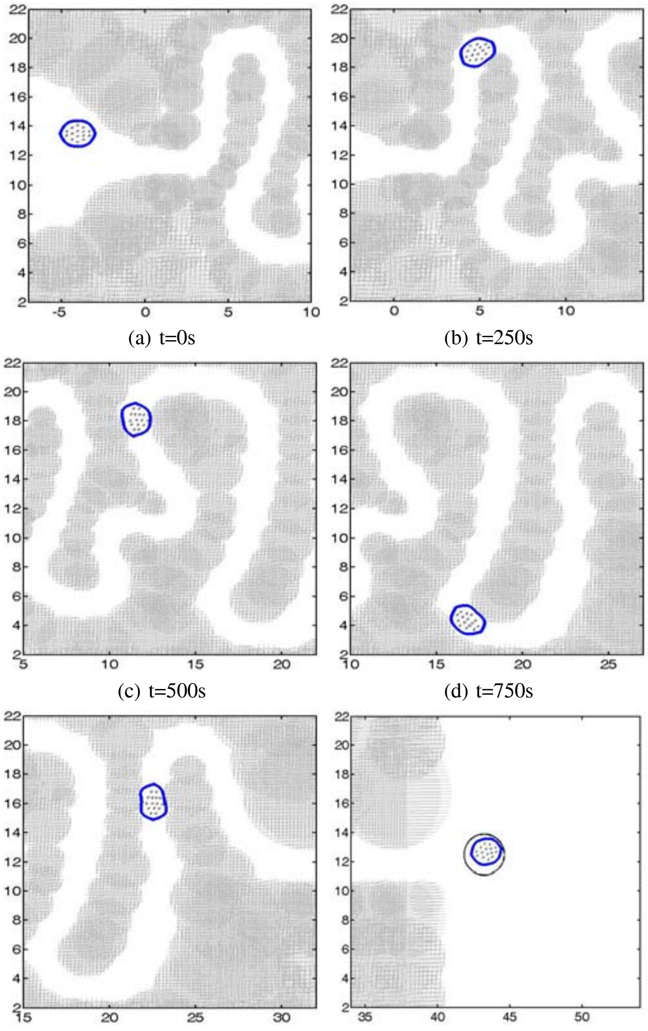


(a) Simulation time: t=0s (b) t=215s

(c) t=705s (d) t=1120s

(e) t=1450s (f) t=1775s

Fig. 5. MDRRF formation in a complex tunnel with CC-spline boundary



(a) t=0s (b) t=250s

(c) t=500s (d) t=750s

(e) t=1100s (f) t=1400s

Fig. 6. MDRRF formation in a complex tunnel with 2 LS-fits

distribution methods such as Voronoi tessellations [17], a simpler approach is preferred. Let the center line of the formation is divided to two segments, FC from front control node to center of formation CR from center to rear vertex as in Figure 3(a). Then the reference vertices are placed equidistantly on segments emanating from these center lines perpendicularly to the formation orientation. When the formation is morphed according to the method in Section IV-A, the norms of the pivot lines extend or shrink by the change of the nearby control vertex positions, thereby preserving the distances between the agents and the formation border as in Figure 3(b). If the line segments shrink causing the reference nodes to converge nearby, the path planners of the associated agents detect neighboring WMRs by their range sensors and activate their avoidance mechanism preventing collision.

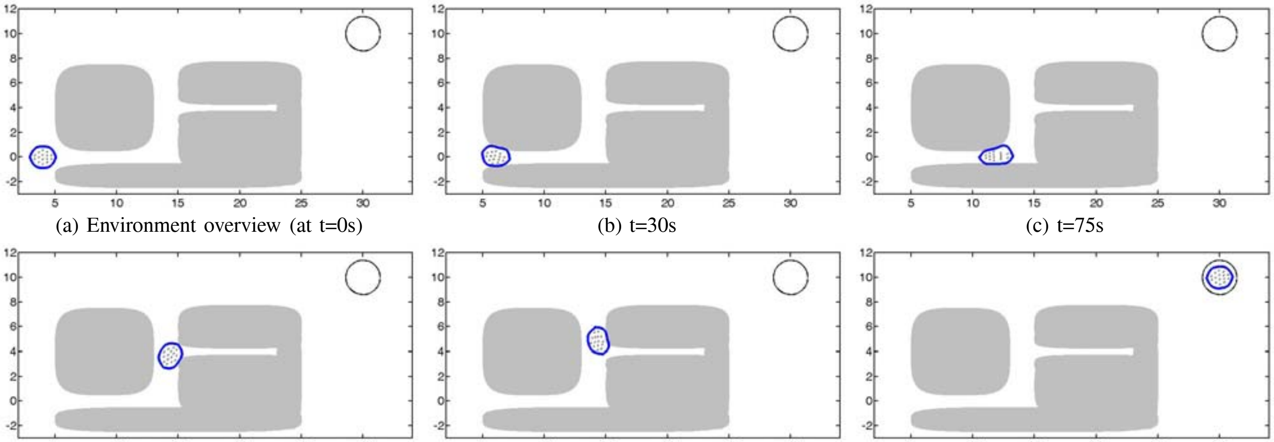
V. Simulation Results

The effectiveness of the MDRRF planner is verified simulations on tunnel environments with wide cross-sections and with U-gaps (see Figures 4 and 7). The aim of the simulations is steering a collection of 15 agents from their initial locations to the desired regions marked by circles in the right side clearance. The agents are assumed to be small ellipsoidal robots with minor and major axis lengths 5-5.5 $[cm]$. The major and minor axis lengths of the formation zone and the WMR sensing regions are:

$$a_f = 110, \quad b_f = 100, \quad a = 11, \quad b = 10 \ \ [cm].$$

The formation and agent reference velocities are $0.07, 0.075$ [m/s] for the CC-spline simulations, and $0.11, 0.117$ [m/s] for the LS-fit simulations. For spline interpolated formation borders, 5 intermediate points are placed between every neighboring knot pair. Ridge regression parameter is selected to be $\epsilon = 0.01$.

The first two simulations are on the collective steering of 15 agents, which utilize two alternative forms of MDRRF planners (see Figure 5 for CC-spline boundaries, and Figure 6 for the LS-fits bounds). In these figures the dotted and continuous envelopes depict the CC-spline and LS-fit based formation borders, respectively. From Figure 5 we can observe that some agents leave the formation of CC-spline boundaries (Figures 5(e) and 5(f)). This is in parallel with



(a) Environment overview (at t=0s)

(b) t=30s

(c) t=75s

(d) t=175s

(e) t=275s

(f) t=500s

Fig. 7. LS-fit based swarm planner in a tunnel with U-gaps.

discrete nature of splines. However, the LS-fits provide more effective bounding and hence improved agent collectivity as in Figure 6. Moreover, the LS-fit approach has a higher formation velocity. Another set of simulations are in Figure 7, where LS-fit based MDRRF planner helps a group of agents maneuver around the U-gap in a tunnel with a dead end corridor. The improved performance is mainly owing to the synergy of the multiple sensor readings from widely distributed sensors over the formation. These simulation environments are typical cases, where FRF based formations fail with the agents relying on their short range sensors. Thus we have repeated the simulations also with the FRF planner and included the results in the attached video file along with MDRRF clips. From the streams, we can observe the FRF planner fails with many agents getting trapped at the concave regions. We can also observe the advantage of MDRRF approach: CC-spline based simulations achieve more accurate and collective path plans than the FRF planners. LS-fits based planners achieve better agent and velocity alignment over the other two approaches resulting in more accurate flocking behavior.

VI. CONCLUSIONS

We have proposed a new algorithm for flocking like behavior of relatively crowded WMR ensembles in obstacle ridden environments. Our method is based on the co-operation of a computationally efficient single robot planner with a simple mobile formation based on geometric evaluations rather than swarm intelligence. The algorithm enables more effective navigation than FRF based approaches in complex environments such as tunnels with variable cross sections, while retaining computational efficiency. Moreover improved navigation is obtained by utilization of data from a network of range sensors instead of resorting to expensive hardware. Therefore this method may reduce the necessity for detailed mappings or locally known milestones for flocks of inexpensive WMRs. The effectiveness of the proposed planner is verified by numerous simulations.

REFERENCES

[1] G. Flake, The Computational Beauty of Nature. Cambridge, MA: MIT Press, ISBN:0262062003, 1999.

[2] C. W. Reynolds, "Flocks, herds, and schools: A distributed behavioral model," Computer Graphics, 21:25-34, 1987.

[3] B. R. Secrest G. B. Lamont, "Visualizing Particle Swarm Optimization - Gaussian Particle Swarm Optimization," *Proc. IEEE Swarm Int. Symposium,* 198-204, 2003.

[4] A. Cervantes, I. Galvan and P. Isasi, "Building Nearest Prototype Classifiers Using a Michigan Approach PSO," *Proc. IEEE Swarm Int. Symposium,* 135-140, 2007.

[5] V. Gazi and K.M. Passino,"Stability Analysis of Social Foraging Swarms," *IEEE Trans. Sys., Man and Cybernetics,* 34(1):539-557, 2003.

[6] F. Zhang, D.M. Fratantoni, D.A. Paley, J. Lund and N.E. Leonard, "Control of Coordinated Patterns for Ocean Sampling", *Int. Journal of Control,* 80(7):1186-1199, 2007.

[7] L. Barnes, W. Alvis, M. Fields, K. Valavanis and W. Moreno, "Swarm Formation Control with Potential Fields Formed by Bivariate Normal Functions," *Proc. 14th Int. Mediterenean Conf. on Control and Automation,* 1-7, 2006.

[8] J.-M. Lien, O.B. Bayazit, R.T. Sowell, S. Rodriguez and N.M. Amoto, "Shepherding Behaviors," *Proc. IEEE Int. Conf. on Robotics and Automation,* 4159-4165, 2004.

[9] M.A. Hsieh and V. Kumar, "Pattern Generation with Multiple Robots," *Proc. IEEE Int. Conf. on Robotics and Automation,* 2442-2447, 2003.

[10] L. Chaimowicz, N. Michael and V. Kumar, "Controlling Swarms of Robots Using Interpolated Implicit Functions," *Proc. IEEE Int. Conf. on Robotics and Automation*, 2487-2492, 2005.

[11] S. Kalantar and U.R. Zimmer, "A Formation Control Approach to Adaptation of Contour-Shaped Robotic Formations," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* 1490-1497, 2006.

[12] N.R. Watson, N.W. John and W.J. Crowther, "Simulation of Unmanned Air Vehicle Flocking," *Proc. Theory and Practice of Computer Graphics,* 130-137, 2003.

[13] G. Dudek, M. Jenkin, E. Milios and D. Wikes, "A Taxanomy for Swarm Robots", *Proc. IEEE/RSJ Int. Conf. on Int. Robots and Sys.,* 441-447, 1993.

[14] T. Sahin and E. Zergeroglu, "A Computationally Efficient Path Planner for a Collection of Wheeled Mobile Robots with Limited Sensing Zones", *Proc. IEEE Int. Conf. on Robotics and Automation,* 1074-1079, 2007.

[15] T. Sahin and E. Zergeroglu, "Computationally Efficient Path Planning for Wheeled Mobile Robots in Obstalce Cluttered Environments", *Robot Motion and Control, Springer Verlag,* 360:259-267, 2007.

[16] T. Tasdizen, J.P. Tarrel and D.B. Cooper, "Algebraic Curves that Work Better,", *IEEE Trans. on Imag. Proc.,* 9(3):405-416, 2000.

[17] G. Varadhan, S. Krishnan, L. Zhang and D. Manocha, "Reliable Implicit Surface Polygonization Using Visibility Mapping,", *Eurographics Symp. on Geometry Processing,* 2006.