General Dynamic Formations for Non-holonomic Systems Along Planar Curvilinear Coordinates

Athanasios Krontiris Sushil Louis Kostas E. Bekris

***Abstract*— This paper describes a general geometric method for planar formations of non-holonomic systems. The approach directly provides the feasible controls that each individual robot has to execute in order for the team to maintain the formation based on the controls of a reference agent, either a real leader-robot or a virtual one. In order to directly satisfy the non-holonomic constraints, the geometric reasoning takes place in curvilinear coordinates, defined by the curvature of the reference trajectory, instead of the typical rectilinear coordinates. The generality of the approach lies on the ability to define dynamic formations so as to smoothly switch between static ones, where the robots can change both of their relative coordinates as they move, and the ability to acquire a desired formation given an initial random configuration. Furthermore, it is possible to correct errors in the achieved configuration of the vehicles on the fly. Simulated experiments are presented to verify the correctness of the provided derivations.**

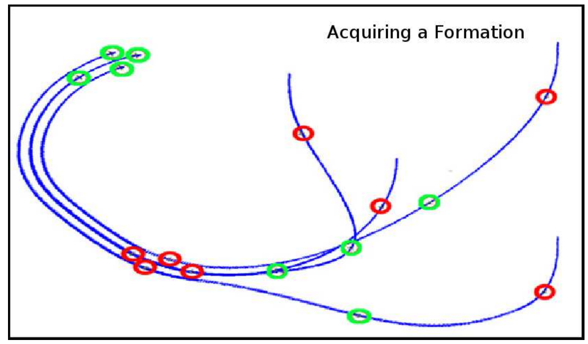

Fig. 1. Blue lines correspond to the trajectory of airplanes, while colored circles point to the location of airplanes at different points in time.

I. Introduction

Formations arise in robotic space exploration, where multiple satellites and spacecraft can provide improved reliability, and military applications, where formations were originally developed and can be utilized by unmanned vehicles. They can also provide reduction in induced drag to aircraft flying in a vee formation, thus, extending the range of a squadron or allowing solar powered airplanes to stay aloft longer. Beyond robotics, formations can be useful in simulations and games that mimic the behavior of real world agents. This paper is especially motivated by training simulations for military officers, which require the modeling of strategic maneuvers for opposing and friendly vehicles, aircraft and boats. The need to model and switch dynamically between formations in a computationally efficient manner arises as a primary requirement during the development of such simulations. Moreover, formations can be used to address the computational complexity of planning for multiple robots. Instead of planning for all the agents in a centralized manner, it is possible to plan for the team as a single larger-size agent. Formations also provide a way for multiple agents to act as a coherent group.

If the robots are holonomic, maintaining a formation is rather straightforward. The focus of this paper, however, is on systems with first-order non-holonomic constrains, which make the problem harder. Beyond maintaining a static formation, it is desirable to allow the shape of the formation to change over time. For instance, the formation might have to adapt to static obstacles in the environment, to the directions of a human operator, or a higher-level autonomous unit that specifies the type of formation. Dynamic formations can also be useful for assembling a static formation from a random configuration, as shown in Fig. 1. This work addresses the above challenges by extending approaches for the geometric computation of feasible trajectories for first-order non-holonomic systems [1]–[3]. It provides methods for acquiring a general planar formation, maintaining a static one and switching dynamically between different formations.

A. Background

Formation planning can be divided into three types: (i) Behavior-based approaches design simple primitives for each agent and combine them to generate complex patterns through the interaction of several agents [4]–[6]. Some behavior-based schemes have been shown to provably converge [7]. (ii) Leader-follower approaches designate one or more agents as leaders that guide the formation [2], [8], [9]. The remaining agents follow the leader(s) with a predefined offset. (iii) Rigid-body formations [10] maintain a constant distance between the agents' configurations. The idea of virtual leaders is a combination of the last two types [11].

Control theory has been used extensively to study formations with tools such as input-output feedback linearization for leader-follower formations, where the coontroller's stability is typically the issue [9]. Once a fixed formation is defined, it is interesting to compute the smoothest trajectory that minimizes energy, a problem cast as designing optimal curves in the SE(3) group [12]. Control-theory is also integrated with graph theory to define controllers that allow transitions between formations [8]. Graphical tools have also been used to study whether there exist non-trivial trajectories

This work is supported by the Office of Naval Research. Any opinions and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsor.

The authors are with the Computer Science and Engineering Department of the University of Nevada, Reno, 1664 N. Virginia St., MS 171, Reno, NV, 89557. Correspondence: bekris@cse.unr.edu

for specified formations given the agent's kinematic constraints [13]. Formations have applications in the areas of robotic sensor networks, where it is important to consider limited in sensing and communication capabilities [14]. More recently, there have been methods for distributed task assignment so that robots can achieve a desired formation [15]. From a geometric point of view, a "formation-space" abstraction, equivalent to the configuration space abstraction, has been defined for permutation-invariant formations of robots that translate in the plane [16].

This paper builds upon motion planning methods for formations [1], [2], which are geometric and exact in nature. This framework allows the robots to track the trajectory of a reference agent that moves independently. Instead of a rigid distance to the leader, the approach adapts the shape during turns to satisfy non-holonomic constraints. When a formation is turning, the method automatically plans for followers on the "outside" to speed up and robots on the "inside" to slow down, resulting in formations that respect the constraints. Moreover, the framework supports almost any arbitrary geometric formation, dynamic adaptation of formations, and is homogeneous, since each robot can utilize the same parametrized algorithm. A related work has focused on optimizing the shape of the formation given the reference agent's trajectory by using a kinetic energy metric in SE(2) [17]. Recently the authors of the current paper have shown how to compute the curvature and velocity limits for the reference agent on the fly so that the satisfaction of similar limits for all the robots is guaranteed [3].

B. Contribution

The approach described here allows a staged process, where planning takes place for a single reference agent and then the team members move in response to maintain the formation. The formation parameters are expressed as the difference between the reference agent and each robot in curvilinear coordinates, such as the coordinates (p, q) in Fig. 2. The curvature of the curvilinear coordinate system is the instantaneous curvature of the reference agent K_L. A static formation implies constant p and q. A dynamic formation allows robots to smoothly switch between static formations. Reasoning in this coordinate system allows the direct computation of controls for the robots relative to the controls of the reference agent while satisfying the non-holonomic constraints. Beyond what has been shown before [1]–[3], this paper contributes the following:

1. It provides a new derivation that allows robots to change both their horizontal p and vertical curvilinear coordinate q during a dynamic formation. The previous approach allowed only changes in the vertical direction q.
2. The current paper shows how to address the case where the follower has different initial orientation from the leader. This allows to solve the problem of the robots getting into a formation from a random configuration, which was not addressed before. The solution to this problem can also be utilized in providing robustness to errors in the execution of the selected trajectories.

II. Problem Setup

The simple car is a model for non-holonomic robots:

$$\dot{x} = u \cdot \cos\theta, \qquad \dot{y} = u \cdot \sin\theta, \qquad \dot{\theta} = u \cdot K, \tag{1}$$

where x, y are Cartesian coordinates for the robot's position and θ is the system's orientation. The controls correspond to the velocity u and the curvature K, which is directly related to the steering angle ϕ of the robot ($K = \tan(\phi)$). The controls often have to respect certain limits:

$$|K| \leq K_{max}, \qquad 0 < u_{min} \leq u \leq u_{max}. \tag{2}$$

Different non-holonomic systems can be casted to the above formulation (e.g., differential-drive robots). The planar motion of aircraft and boats is often modeled as a simple car.

There is a reference agent, the leader L, which selects its controls u_L, K_L independently. The leader can correspond to one of the robots or to a virtual agent. L's trajectory could be either precomputed or computed on the fly. If precomputed, then there is no constraint to the type of formations considered here. If computed on the fly, then the leader should be defined ahead of all the follower robots. If the leader is a virtual agent, then every kind of formation is allowed.

A static formation is defined relative to the leader in curvilinear coordinates. Curvilinear coordinates are defined for curved coordinate lines. The curvature K_L becomes the instantaneous curvature of the reference agent. For example, if airplane L in Fig. 2 is the leader, then the middle blue line corresponds to the "x" axis of the curvilinear system, and the line perpendicular to the airplane is the "y" axis. To maintain a static formation, each robot must maintain a constant p distance along the leader's curved trajectory and q distance along the perpendicular direction, as in Fig.2. The variable d_L denotes the distance the reference agent has covered and $s = d_L + p$ denotes the distance that the projection of the follower has covered along the leader's trajectory. Note that if the robot is already in a static formation, then $\theta = \theta_L(s_i)$, i.e., the orientation of the robot and the (past) orientation of the leader when it was at the projection is the same. Given the above definitions, the following problems are considered:

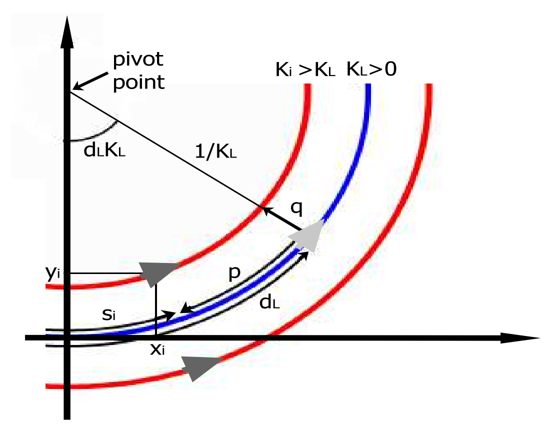

Fig. 2. An illustration of the curvilinear coordinates (p,q) defined by the reference agent moving with constant curvature K_L.

- What are the controls u, K that a follower should execute in order to maintain a static configuration p, q, $\theta(=\theta_L(s))$ in curvilinear coordinates relative to the leader?
- What controls u, K should the follower execute in order to switch from an initial configuration p_0, q_0, $\theta_0(=\theta_L(s_0))$ to a new one with parameters p_1, q_1, $\theta_1(=\theta_L(s_1))$ where $p_0 \neq p_1$ and $q_0 \neq q_1$?
- If a robot is in a configuration where there is no p_0 so that $\theta_0 = \theta_L(s_0)$ (i.e., the robot cannot be considered in formation with the leader), what controls should the follower execute so that it achieves a static configuration p_1, q_1, $\theta_1(=\theta_L(s_1))$ relative to the leader?

III. Dynamic Formations

The following section provides equations for the controls u, L of the follower given the leader's controls u_L, K_L and functions for p, q and their derivatives. If the functions for p and q are constant, then the static case is addressed, otherwise it is possible to define a switching maneuver that results in a dynamic formation. Section III.B addresses the challenge of acquiring a formation by considering the case $\theta_0 \neq \theta_L(s_0)$ for all reasonable s_0.

A. General Dynamic Formations

Consider a follower, which must retain distance p and q from the leader along curvilinear coordinates. Then, given Fig. 2 and trigonometry on angle $sK_L(s)$ at the pivot point, which is common for the follower and the projection point, the coordinates of the follower are:

$$\begin{aligned} x &= (\frac{1}{K_L(s)} - q) \cdot sin(sK_L(s)), \\ y &= \frac{1}{K_L(s)} - (\frac{1}{K_L(s)} - q) \cdot cos(sK_L(s)). \end{aligned} \tag{3}$$

To achieve dynamic formations, the approach allows the coordinates $[p,\ q]$ to be functions of time or distance along the leader's trajectory. Assume that the leader's controls u_L and K_L are piecewise constant. Then the first derivatives of the Cartesian coordinates in Eq.3 are:

$$\begin{aligned} \dot{x} &= -\dot{q} sin(sK_L) + \dot{s}(1 - qK_L(s)) \cdot cos(sK_L(s)) \\ \dot{y} &= \dot{q} cos(sK_L) + \dot{s}(1 - qK_L(s)) \cdot sin(sK_L(s)) \end{aligned} \tag{4}$$

Then, in order to compute the follower's velocity it is sufficient to compute $u(s) = \sqrt{\dot{x}^2 + \dot{y}^2}$, which will result into the following expression:

$$u(s) = \dot{d}Q = u_L \cdot Q \tag{5}$$

where

$$Q = \sqrt{\left(\frac{\partial q}{\partial d}\right)^2 + \left(1 + \left(\frac{\partial p}{\partial d}\right)\right)^2 \cdot N^2} \tag{6}$$

and $N = 1 - qK_L$. The result implies that the follower's velocity is split into two terms: (a) One direction along the horizontal curvilinear coordinate p. This depends on the change in p during the dynamic formation (i.e., $\frac{\partial p}{\partial d}$) and is also adapted according to the difference in curvature between the follower and the leader (i.e., the term N). (b) The other direction is along the perpendicular coordinate q and depends upon the change in q (i.e., $\frac{\partial q}{\partial d}$).

To compute the follower's curvature, it is necessary to compute the second order derivatives of the Cartesian coordinates, since the typical expression for curvature is:

$$K = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{\left(\sqrt{\dot{x}^2 + \dot{y}^2}\right)^3} \tag{7}$$

The derivation results in the following expression:

$$K = \frac{\left(1 + \frac{\partial p}{\partial d}\right)}{Q} \left(K_L + \frac{K_L \left(\frac{\partial q}{\partial d}\right)^2 + N \left(\frac{\partial^2 p}{\partial d^2}\right)}{Q^2} \right) - \frac{N}{Q^3} \frac{\partial^2 p}{\partial d^2} \frac{\partial q}{\partial d} \tag{8}$$

The advantage of considering curvilinear coordinates is that it is possible to exactly compute a follower's controls given Eqs. 5 and 8 as functions of the reference agent's controls. Note that the K_L term in the above equations corresponds to the leader's curvature at the projection point: $K_L(s)$. This means that the follower has to keep track of its projection onto the leader's trajectory. Furthermore, it has to be that $p < 0$, which is the reason why the reference agent is selected to be ahead of every other robot. Otherwise, a follower requires the curvature of the reference agent into the future, and if the leader's trajectory is computed on the fly, then this is not available. In contrast to the curvature, the velocity u_L is the current velocity of the leader $u_L(d)$.

The above equations require functions for dynamic p, q and their derivatives over the arc length d. If p and q are constant and their derivatives are 0, then the equations for static formations can be acquired from Eqs. 5, 8:

$$\begin{aligned} u(s) &= u_L(d)(1 - q \cdot K_L(s)) \\ K(s) &= \frac{K_L(s)}{1 - q \cdot K_L(s)} \end{aligned} \tag{9}$$

Alternatively, p and q can be defined so as to achieve a switching maneuver between two static formations. One such maneuver is a third order curve interpolating the curvilinear coordinates of the follower in the initial static formation (p_0, q_0) with those in the final formation (p_1, q_1). For instance, the perpendicular curvilinear coordinate can be defined as follows:

$$\begin{aligned} q(d) &= q_0 + (q_1 - q_0) \cdot \tau^2 \cdot (3 - 2\tau) \\ \frac{\partial q}{\partial d} &= 6 \cdot \frac{q_1 - q_0}{\Delta d} \cdot \tau \cdot (1 - \tau) \\ \frac{\partial^2 q}{\partial d^2} &= 6 \cdot \frac{q_1 - q_0}{(\Delta d)^2} \cdot (1 - 2 \cdot \tau) \end{aligned} \tag{10}$$

where $\Delta d = d_1 - d_0$ and $\tau = \frac{d - d_0}{\Delta d}$. For the horizontal curvilinear coordinate it is sufficient to use a linear interpolation:

$$p(d) = (p_1 - p_0) \cdot \tau \tag{11}$$

The above maneuver is defined for $d_0 \leq d \leq d_1$, where the range (d_0, d_1) expresses the portion of the leader's trajectory during which the dynamic formation is taking place. Outside this range, $\frac{\partial q}{\partial d} = \frac{\partial^2 q}{\partial d^2} = 0$. For $d < d_0$, $q(d) = q_0$ and for $d > d_1$, $q(d) = q_1$.

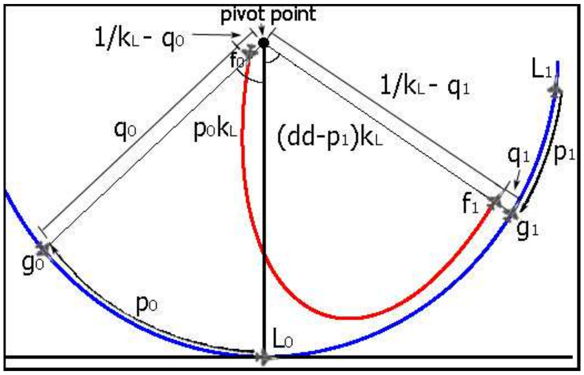

Fig. 3. An illustration of the general case where $\theta \neq \theta_L(s)$. Hermite interpolation in curvilinear coordinates is utilized to define a curve (in red) that will bring the follower to a desired static formation with the leader.

Once the robots are in formation while using Eq. 9, they will maintain the formation regardless of the leader's trajectory. Moreover, if all the robots in the team execute a static formation, there is a guarantee for collision avoidance despite the formation not being rigid. During a dynamic switch between formations using Eqs. 5, 8, it is important for the leader to maintain constant curvature. Otherwise, at the point where the curvature changes, the follower will not have the same orientation as the projection state on the leader's trajectory ($\theta \neq \theta_L(s)$). This breaks the assumptions based on which Eqs. 5, 8 are derived and introduces errors. Consequently, a locking mechanism has to be enforced. When a follower switches between formations, the leader cannot change curvature.

B. Acquiring a Formation

This section addresses the general case, illustrated in Fig. 3, where the follower is not in formation with the leader. The objective is to come up with the equations of motion that will bring the follower into a desired static formation with parameters $p_1, q_1, \theta_1 (= \theta_L(s_1))$. The idea is to define the curve the follower must follow by employing interpolation in curvilinear coordinates between its current configuration and a configuration that satisfies the formation parameters. Such a process can be achieved with the Hermite curve interpolation, which interpolates between two points P_1 and P_4, while respecting the tangent vectors R_1 and R_4 at these points respectively. Given these points and vectors as input parameters, the interpolation curve is defined as follows:

$$\begin{aligned} H(\tau) &= (2\tau^3 - 3\tau^2 + 1)P_1 + (-2\tau^3 + 3\tau^2)P_4 \\ &+ (\tau^3 - 2\tau^2 + \tau)R_1 + (\tau^3 - \tau^2)R_4 \end{aligned} \quad (12)$$

For the problem at hand, P_1 will be the position where the dynamic formation will be initiated and P_4 is the point where the formation is achieved. R_1 will be along the current direction of the follower and R_4 will be along the direction specified by the formation at P_4. The parameter τ is as before: $\tau = \frac{d-d_0}{\Delta d}$. The parameter d_0 is the position (arc length) of the leader along its trajectory when the dynamic formation is initiated and d_1 is the leader's position when the formation is achieved. Thus, $d_0 \leq d \leq d_1$ and $0 \leq \tau \leq 1$.

In order to initiate the interpolation, it is necessary for the follower to first locate an appropriate projection point along the leader's past trajectory. This will allow the definition of curvilinear coordinates for its current position. A projection point is collinear with the follower and the leader's pivot point. Fig. 3 demonstrates the situation, where the projection point is denoted as g_0. If no such point exists, the follower can move towards the leader until one is found. If the follower can reduce and increase its distance with the leader, then the existence of such a projection point is guaranteed.

Once a projection point has been discovered, then it is possible to express the input parameters P_1, P_4, R_1 and R_4 of the interpolation procedure in curvilinear coordinates. Fig. 3 corresponds to the beginning of the dynamic formation. The input parameters are computed relative to the leader's coordinates. This means that the leader will be in position $(0, 0)$ and it's orientation is $\theta(d) = 0$. Then if:

- q_0 is the distance between the projection point and the follower's position,
- p_0 is the distance along the leader's trajectory between the projection point and the leader's current points and
- $\theta_0 = \theta - \theta_L$ (i.e., the difference in orientation between the follower and the leader)

the vectors P_1 and R_1 are defined as follows:

$$P_1 = \begin{bmatrix} \left(\left(\frac{1}{K_L} - q_0\right) sin(p_0 K_L)\right) / \Delta d \\ \left(\frac{1}{K_L} - \left(\frac{1}{K_L} - q_0\right) cos(p_0 K_L)\right) / \Delta d \end{bmatrix}$$

$$R_1 = \begin{bmatrix} cos(\theta_0) \\ sin(\theta_0) \end{bmatrix}$$

Given the curvilinear coordinates (p_1, q_1) that the follower will have at the end of the dynamic formation, the remaining input parameters are:

$$P_4 = \begin{bmatrix} \left(\left(\frac{1}{K_L} - q_1\right) sin((\Delta d + p_1) K_L)\right) / \Delta d \\ \left(\frac{1}{K_L} - \left(\frac{1}{K_L} - q_1\right) cos((\Delta d + p_1) K_L)\right) / \Delta d \end{bmatrix}$$

$$R_4 = \begin{bmatrix} cos(\theta_1) = cos((\Delta d + p_1) K_L) \\ sin(\theta_1) = sin((\Delta d + p_1) K_L) \end{bmatrix}$$

The orientation θ_1 is computed as $((\Delta d + p_1) K_L)$ because upon the completion of the curve, the follower robot will be in formation and it will have the same orientation as the projection at that point in time.

In order to compute the velocity and the curvature that the followers should acquire in order to trace the desired curve, it is first necessary to define the Cartesian coordinates x and y as functions of the curvilinear parameters computed through the Hermite curve interpolation. Note that the points P_1 and P_4 are defined in a coordinate system where $\Delta d = 1$ (since their coordinates are divided by Δd). Thus, the Cartesian position vector along the interpolated curve is defined as: $[x(\tau)\ y(\tau)]^T = \Delta d \cdot H(\tau)$. Then the velocity and the curvature of the leader can be computed, which require to first derive the first and second derivatives of $H(\tau)$ with respect to τ. The derivation results into the following expressions:

$$u(s) = u_L M \text{ where}$$

$$M = \sqrt{\left(\frac{\partial H_x(\tau)}{\partial \tau}\right)^2 + \left(\frac{\partial H_y(\tau)}{\partial \tau}\right)^2}, \tag{13}$$

$$K = \frac{\frac{\partial (H_x(\tau))}{\partial \tau}\frac{\partial^2 H_y(\tau)}{\partial \tau^2} - \frac{\partial^2 H_x(\tau)}{\partial \tau^2}\frac{\partial (H_y(\tau))}{\partial \tau}}{\Delta d \cdot M^3} \tag{14}$$

The smoothness of the curve can be controlled by changing the magnitude of the control vectors R_1 and R_4. The same vectors can be adapted so as to change the shape of the curve for collision avoidance purposes with other followers that might also be attempting to approach the leader.

The derivation in this section employed the assumption that the leader maintains constant curvature throughout the execution of the follower's transition into a formation. Nevertheless, this is now a less restricting assumption compared to the previous section, since the follower can always recompute the desired point P_4 and R_4 where the static formation will be achieved as soon as the leader changes curvature (more precisely: when the projection point's curvature changes). For the new equations, there is no issue with the fact that when the curvature changes the follower does not have the same orientation as the leader at the projection point.

IV. Experimental Validation

The above functions were tested in a simulation engine developed at the authors' institute as a training and educational tool. The engine is built over the Python-Ogre platform using Python as the programming language. This engine provides the opportunity to select different vehicles and direct them to a desired goal. The modeled vehicles in the attached experiments are visualized as aircraft but are moving in a plane as simple cars according to Eq. 1, which is integrated using the 4th order Runge-Kutta method. Each aircraft is modeled as a different entity and has a specific amount of time to complete its computations ($\frac{1}{60}$sec).

For these experiments, the reference agent is set to be one of the vehicles leading the formation (i.e., being ahead of every other vehicle). Since the focus of this work is not on the leader's trajectory, the leading vehicle is implementing a straightforward PID controller to compute its path on the fly. The PID controller is provided a target either from a human user or by loading a predefined sequence of target. The controller changes the steering angle ϕ_L of the leader ($K_L = tan(\phi_L)$) to minimize the difference between θ_L and the direction to the goal. This typically has the effect that the leader gradually turns towards its target and then executes a straight-line path. The leader has high velocity away from the target but as it approaches the target, the vehicle reduces velocity and initiates a circular trajectory. The circular trajectory depends on the maximum curvature and the original position from which the controller was initiated. The leader respects the control limits of Eq. 2.

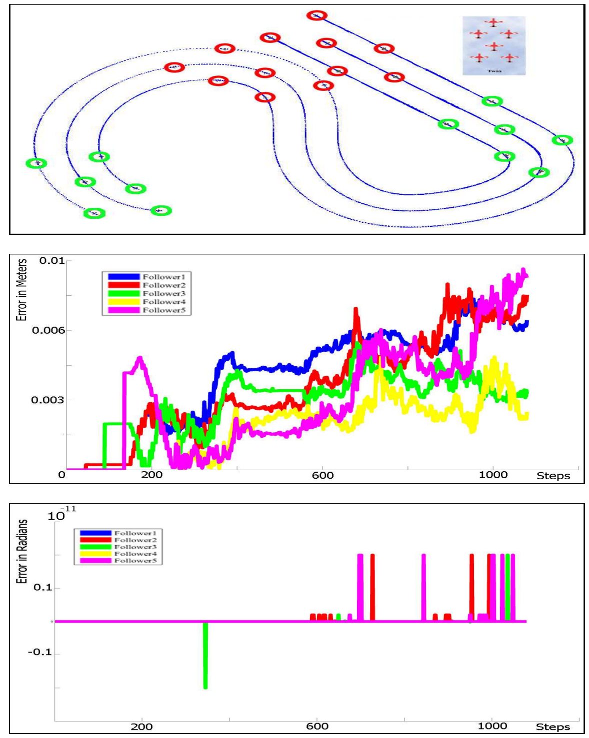

Fig. 4. (top) The path the vehicles execute in static formation. (middle) The errors the 5 followers had during the execution of their path. (bottom) the difference in orientation between the followers and the leader. Each color represents a different follower. The size of an aircraft in the top figure corresponds to 17m and the distance between vehicles is 60m.

A. Errors in Formation Maintenance

Due to numerical integration, small errors will appear over time, as well as because the equations derived assume constant controls for the leader. To compute the ground truth position for the follower robots a procedure was utilized that computed the correct Cartesian coordinates for each follower based on the desired curvilinear coordinates of the formation given the leader's position. The graphs in Fig. 4 show the errors in distance and orientation that the followers have during static formation. In order to maintain static formation the followers have to be parallel to the leader's path in curvilinear coordinates.

The vehicles execute the path in Fig.5 using the functions that are described in Section.III. Three different dynamic formations are executed in this path. The images next to the path represent the formation that the vehicles have at each specific point on the path.

B. Success of Formation Acquisition

In this section, the accuracy of the followers getting into a formation is tested. The error graphs are computed as soon as the followers get into a formation (i.e., the procedure in Section III.B is completed). The path is illustrated in Fig.1, while the errors in distance are shown in Fig. 6.

V. Discussion

This paper has outlined a framework for simulating dynamic formations along planar curvilinear coordinates for

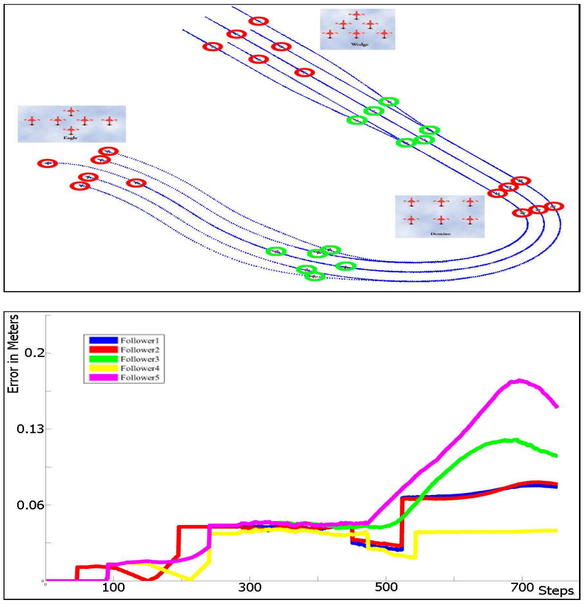

Fig. 5. (top) The path the vehicles execute while in dynamic formation. (bottom) The errors from all the followers. The size of the vehicles is 17m and the distance between vehicles is 60m.

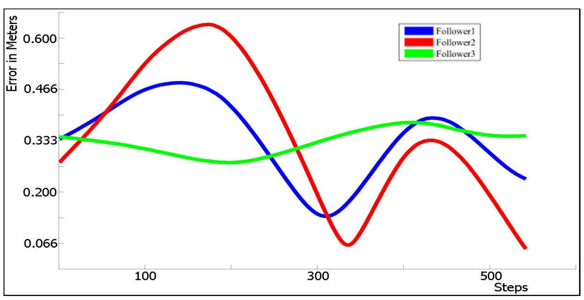

Fig. 6. Errors for the path in Fig.1 where the followers are acquiring a formation.

systems with non-holonomic constraints. In comparison to the related literature [2], this paper provided a derivation that allows changes in formations both along the horizontal and the perpendicular curvilinear coordinate. Moreover, it showed how to address the case where the vehicles coalesce in order to acquire a formation.

An important issue is the satisfaction of the constraints in Eq. 2. In previous work [3], the authors have shown that it is possible to constrain the leader's valid velocities and trajectories on the fly in a way that allows the followers to always respect the same constraints during a dynamic formation. It is possible to extend this work so as to guarantee the satisfaction of these constraints during the initial achievement of a formation. If the satisfaction of the constraints forces the follower to stop traversing the curve computed by the Hermite curve interpolation, then a new curve can be computed on the fly over a longer distance.

It is interesting to investigate how the formation algorithm provided here can be integrated with motion planners so as to achieve collision avoidance, both with static and dynamic obstacles [18]. The planner has to automatically select the type of formation the team must follow and plan for transitions between static formations. Moreover, the current work assumed that the agents have access to the controls of the reference agent. An implementation on real robots would require broadcasting this information or extracting the controls by sensing the reference agent's trajectory. Beyond integrating the approach with motion planners, future work will investigate decentralized methods for deciding the reference agent's trajectory and the type of formation that the team should achieve. Other extensions, include work on a 3D version, especially for modeling aircraft formations, as well as an extension to higher-order aircraft so as to allow continuous or smooth velocity and curvature.

REFERENCES

[1] T. D. Barfoot, C. M. Clark, S. M. Rock, and G. M. T. D. D'Eleuterio, "Kinematic path planning for formations of mobile robots with a non-holonomic constraints," in *IEEE/RSJ Intern. Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002.

[2] T. D. Barfoot and C. M. Clark, "Motion planning for formations of mobile robots," *Robotics and Autonomous Systems*, vol. 46, no. 2, pp. 65–78, February 2004.

[3] T. Krontiris, S. Louis, and K. E. Bekris, "Simulating planar formations for non-holonomic systems along curvlinear coordinates," in *Third International Conference on Motion in Games*, Zeist, Netherlands, November 2010.

[4] T. Balch and R. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, Dec 1998.

[5] J. Fredslund and M. J. Mataric, "A general, local algorithm for robot formations," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 873–846, Oct 2002.

[6] L. E. Parker, B. Kannan, X. Fu, and T. Y., "Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations," in *Proc. of the IEEE/RSJ Intern. Conference on Intelligent Robots and Systems (IROS)*, 2003.

[7] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile agents using nearest neighbor rules," *IEEE Trans. on Automatic Control*, vol. 8, no. 6, pp. 988–1001, 2003.

[8] J. Desai, J. Ostrowski, and V. J. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *Transactions on Robotics and Aut.*, vol. 17, no. 6, pp. 905–908, 2001.

[9] H. G. Tanner, G. J. Pappas, and V. J. Kumar, "Leader-to-formation stability," in *IEEE Transactions on Robotics and Automation*, 2004.

[10] M. Lewis and K.-H. Tan, "High-precision formation control of mobile robotis using virtual structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, Oct 1997.

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

[12] C. Belta and V. J. Kumar, "Motion generation for formation of robots: a geometric approach," in *IEEE International Conference on Robotics and Automation*, Seoul, South Korea, May 21-26 2001.

[13] P. Tabuada, G. J. Pappas, and P. Lima, "Motion feasibility of multi-agent formations," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 387–392, 2005.

[14] M. Ji and M. Egerstedt, "Distributed coordination control of multi-agent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, Aug. 2007.

[15] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Distributed multi-robot task assignment and formation control," in *ICRA*, May 2008, pp. 128–133.

[16] S. Kloder and S. Hutchinson, "Path planning for permutation-invariant multirobot formations," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 650–665, 2006.

[17] R. M. Bhatt, C. P. Tang, and V. N. Krovi, "Formation optimization for a fleet of wheeled movile robots: A geometric approach," *Robotics and Autonomous Systems*, vol. 57, no. 1, pp. 102–120, January 2009.

[18] P. Ogren and N. E. Leonard, "Obstacle avoidance in formation," in *IEEE Intern. Conference on Robotics and Automation (IEEE)*, Taipei, Taiwan, 2003.