TÜBİTAK

Turk J Elec Eng & Comp Sci, Vol.18, No.5, 2010, © TÜBİTAK
doi:10.3906/elk-0906-22

Formation control of nonholonomic mobile robots using implicit polynomials and elliptic Fourier descriptors

Yeşim Hümay ESİN, Mustafa ÜNEL
Faculty of Engineering and Natural Sciences, Sabancı University
İstanbul-TURKEY
e-mail: yesime@su.sabanciuniv.edu, munel@sabanciuniv.edu

Abstract

This paper presents a novel method for the formation control of a group of nonholonomic mobile robots using implicit and parametric descriptions of the desired formation shape. The formation control strategy employs implicit polynomial (IP) representations to generate potential fields for achieving the desired formation and the elliptical Fourier descriptors (EFD) to maintain the formation once achieved. Coordination of the robots is modeled by linear springs between each robot and its two nearest neighbors. Advantages of this new method are increased flexibility in the formation shape, scalability to different swarm sizes and easy implementation. The shape formation control is first developed for point particle robots and then extended to nonholonomic mobile robots. Several simulations with robot groups of different sizes are presented to validate our proposed approach.

Key Words: *Formation, Coordination, Autonomous, Nonholonomic, Implicit Polynomials, Elliptic Fourier Descriptors*

1. Introduction

The formation control problem for mobile robots has been a challenging research issue. Complicated mission tasks can be achieved more effectively using groups of robots in formation. Coordination and formation decrease the working time and provides more flexibility in the execution of the tasks. It is well known that robot groups are more successful than a single robot in many critical applications such as exploration [1], surveillance [2], search and rescue [3], mapping of unknown or partially known environments [4], distributed manipulation [5], and the transportation of large objects [6]. Certain tasks such as carrying a heavy load [7], hunting [8] and encircling an enemy [9] can be achieved more efficiently using a certain formation. Formation may increase the task performance tremendously.

Formation control consists of several subproblems such as determining the appropriate formation shape, achieving formation, maintaining formation and switching between different formation shapes. Among these

subproblems, achieving and maintaining formation are important research areas. The formation method should be flexible in terms of the formation shape and it should be applicable to robot groups of different kinds and numbers. There exist several approaches for modeling and solving these problems, ranging from paradigms based on combining reactive behaviors [10] to those based on leader-follower graphs [11] and potential field methods [12].

In this work, a new formation control method is developed. The proposed method utilizes both implicit polynomials (IP) and elliptic Fourier descriptors (EFD) to represent the desired formation shape [13]. These descriptions introduce flexibility into the formation shape. The implicit polynomial description is used in the first phase of the formation control to form the desired shape and the parametric description is used in the second phase of the control to maintain the formation. In addition to these controls, a coordination control to keep a desired distance between the robots is introduced as well. The main advantages of this new approach when compared with existing work are increased flexibility in the formation shape, scalability to large number of robots (including heterogeneous robots) and efficient implementation. For example, in some other methods ([14], [15]), there exist certain restrictions on the formation shape.

Formation control is first developed for point particle robots. Early results with point particle robots also appeared as a conference paper in [16]. Current work embellishes and extends that previous work to nonholonomic mobile robots. The simulation program developed in Matlab/Simulink environment is designed to be modular so that the simulations can be performed with any number of robots and with arbitrary desired formation shape. The performance of the proposed method has been verified by both simulations with point particle and nonholonomic robot groups of different sizes.

Organization of this paper is as follows. Representation of the free-form curves is reviewed in Section 2. Control of nonholonomic mobile robots is covered in Section 3. Coordination of robots, designing a control for shape formation, keeping the formation, and formation control for nonholonomic robots are presented in Section 4. Section 5 presents the simulation results and discussion. Section 6 concludes the paper with some remarks and indicates possible future directions.

2. Parametric and implicit representation of closed curves

Coordinates of a point on a closed curve satisfies the relations

$$\begin{aligned} x(s) &= x(s+L) \\ y(s) &= y(s+L), \end{aligned} \tag{1}$$

where L is the total length of the curve, $x(s)$ and $y(s)$ are functions of arc length parameter s. Note that these equations are periodic with period L. Because of periodicity, a closed curve can be represented with an elliptic Fourier descriptor (EFD) [17] as

$$\begin{aligned} x(\theta) &= a_o + \sum_{k=1}^{n} (a_k \cos k\theta + b_k \sin k\theta) \\ y(\theta) &= c_o + \sum_{k=1}^{n} (c_k \cos k\theta + d_k \sin k\theta), \end{aligned} \tag{2}$$

where x and y coordinates of the points on the curve are written as functions of a normalized parameter θ defined as

$$\theta = \frac{s}{L} 2\pi \tag{3}$$

Clearly, $\theta \in [0, 2\pi)$ for $s \in [0, L)$.

Note that θ is an angle since $\sin(\cdot)$ and $\cos(\cdot)$ are defined only for angles. However, θ can be related to time t. Recall that an angle can be written as $\theta = \omega t$, where ω is the angular velocity of the moving point on the curve. For example, letting $\omega = 1$ rad/sec implies that $\theta = t$. We know from differential geometry that curves can be parameterized in infinitely many different ways. The parameter can be either arc length s, or time t, or something similar. There is always a mapping between different parametrization of the same curve. As a result, x and y are periodic functions of t with period of $2\pi/\omega$.

In equation (2), n is a positive integer, total number of harmonics in the representation. It is well-known that the shape information is contained in the low frequency components, i.e. first few harmonics. The accuracy of the representation increases with the number of harmonics used in the description. On the other hand, the computation time increases with more harmonics. Usually 5–10 harmonics are sufficient to represent moderately complex shapes.

The coefficients in EFD representation can be determined using the formulas

$$\begin{aligned} a_0 &= \frac{1}{N}\sum_{i=1}^{N} x_i \qquad b_0 = \frac{1}{N}\sum_{i=1}^{N} y_i \\ a_k &= \frac{2}{N}\sum_{i=1}^{N} x_i \cos(k\omega t) \qquad b_k = \frac{2}{N}\sum_{i=1}^{N} x_i \sin(k\omega t) \\ c_k &= \frac{2}{N}\sum_{i=1}^{N} y_i \cos(k\omega t) \qquad d_k = \frac{2}{N}\sum_{i=1}^{N} y_i \sin(k\omega t) \end{aligned} \tag{4}$$

where N is the number of samples taken from the closed curve, and x_i and y_i are the coordinates of the points.

Curves can also be represented using implicit polynomials (IP). The implicit polynomial representation, $f(x, y) = 0$, of a closed curve can be found by several methods. In this study, the implicit polynomial representation of the closed formation shape is obtained using the *implicitization method* detailed in[13]. The implicit polynomial equations can be very useful for generating potential fields to be used in the shape formation phase. The parametric equations such as EFD described above can then be used for maintaining the desired formation.

The representation of a closed curve with an implicit polynomial equation takes the form

$$F(x, y) = \sum_{0 \le i+j \le d} a_{ij} x^i y^j = 0, \tag{5}$$

where a_{ij} are the coefficients and d is the degree of the polynomial. As detailed in[13], the degree of the polynomial is $d = 2n$ where n is the number of harmonics used in EFD representation.

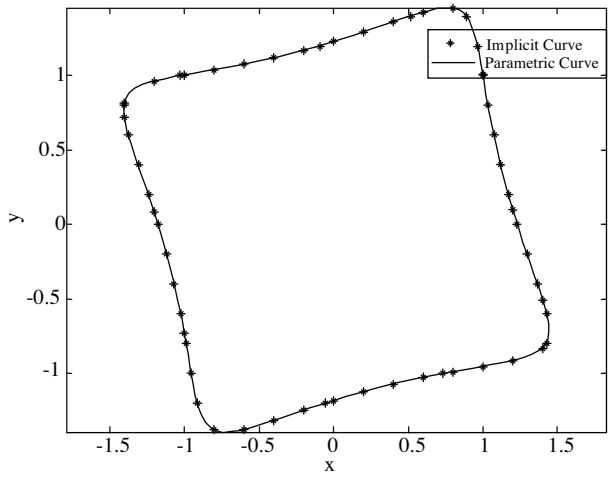

Figure 1. Representation of a quadrangle by EFD and IP.

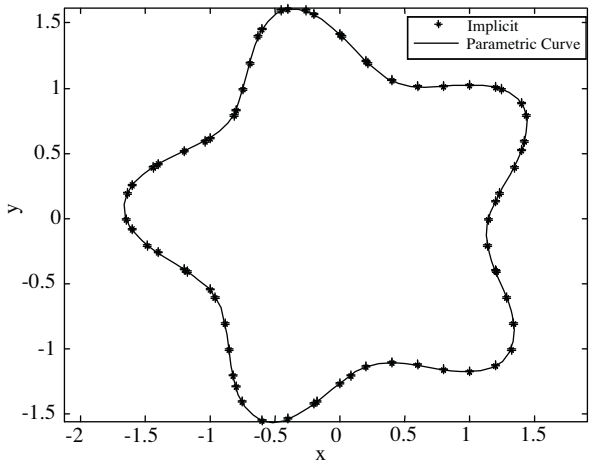

Figure 2. Representation of a star-like shape by EFD and IP.

Figure 1 depicts a closed curve with EFD and IP representations. In this figure, the curve is represented by EFD using 3 harmonics and by the corresponding IP of degree 6. In Figure 2, a more complicated curve is modeled by EFD using 6 harmonics and by the superimposed IP of degree 12. In these figures, the continuous lines depict EFD description while the dotted lines show the IP representations. As seen from the figures, EFD and IP representations exactly match each other, and therefore complicated shapes/curves can well be represented by using either representation. We'll adopt both representations in our development of formation control laws.

3. Control of nonholonomic mobile robots

Simple dynamic and kinematic equations are not sufficient for modeling nonholonomic robots. There is a widely known kinematic model for nonholonomic unicycle robots which is given as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta)u_1 \\ \sin(\theta)u_1 \\ u_2 \end{pmatrix}, \tag{6}$$

where x and y are the Cartesian coordinates of the center of mass of the robot, θ is the orientation angle with respect to the x axis. The controls u_1 and u_2 are related to the velocities of the right and left wheels. This equation can alternatively be written as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2. \tag{7}$$

As it can be seen from equation (7), this is an underactuated system where there are only two actuator inputs to control three outputs, i.e. pose variables ((x, y, θ)), of the robot. Control of such underactuated systems is a challenging task.

It is shown in [18] that although nonholonomic mobile robots are completely controllable in their configuration space, they cannot be stabilized to a desired pose by using smooth time-invariant state-feedback control.

However, the feedback stabilization of a point on a nonholonomic mobile robot was shown to be possible in [19]. In that work, it is proved that feedback stabilization of the robot's pose around the pose of a "virtual reference robot" is possible provided that the reference robot keeps moving.

For the time-varying reference tracking, the reference trajectory should satisfy the nonholonomic constraint. This is ensured by defining the trajectory using a virtual reference robot which moves according to the model

$$\begin{pmatrix} \dot{x_r} \\ \dot{y_r} \\ \dot{\theta_r} \end{pmatrix} = \begin{pmatrix} \cos(\theta_r) \\ \sin(\theta_r) \\ 0 \end{pmatrix} u_{1r} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_{2r}, \tag{8}$$

where (x_r, y_r) is position of the virtual reference robot with respect to the Cartesian coordinates and θ_r is its orientation. u_{1r} and u_{2r} are respectively its linear and angular velocities.

If x_r, y_r and θ_r are continuously differentiable and bounded as $t \to \infty$, it can be shown that

$$\begin{pmatrix} u_{1r} \\ u_{2r} \\ \theta_r \end{pmatrix} = \begin{pmatrix} \dot{x}_r \cos(\theta_r) + \dot{y}_r \sin(\theta_r) \\ (\ddot{y}_r \dot{x}_r - \ddot{x}_r \dot{y}_r)/(\dot{x}_r^2 + \dot{y}_r^2) \\ \arctan(\dot{y}_r / \dot{x}_r) \end{pmatrix}. \tag{9}$$

Note that computation of u_{1r}, u_{2r} and θ_r by Equation (9) requires first and second derivatives of x_r and y_r. If the mathematical forms of these signals are available, direct differentiation is used. However, we usually don't have mathematical forms since these are the Cartesian position of the virtual robot which in turn moves under some external potential field. Thus, we have only numerical values for x_r and y_r that need to be differentiated. In order to compute these derivatives numerically, we first filter x_r and y_r by a low-pass filter and then use Euler's backward difference. Suppose we want to compute derivative of x_r. The reason that we use low-pass filter is to smooth out the signal to eliminate high frequency noise. Passing the signal x_r through a low-pass filter amounts to the following discrete update rule in the time domain:

$$x_f[k] = \eta x_f[k-1] + (1-\eta) x_r[k], \tag{10}$$

where $x_f[k]$ is the filtered position signal at time k and it is computed as a convex combination of previous filtered value, $x_f[k-1]$, and the present value of the actual signal, $x_r[k]$. η is the parameter of the filter which adjusts relative importance of these two contributions, and usually chosen as a value in the interval $0.8 \leq \eta \leq 0.95$.

Once x_f is obtained, its derivative is computed using approximate derivative based on Euler's backward difference, namely

$$\frac{dx_f}{dt} \approx \frac{x_f[k] - x_f[k-1]}{T}, \tag{11}$$

where T is the sampling time. Second derivative can be computed similarly.

The tracking errors $\tilde{x}$, $\tilde{y}$ and $\tilde{\theta}$ are defined as the difference between the pose of actual robot and the virtual reference robot, namely

$$\begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} - \begin{pmatrix} x_r \\ y_r \\ \theta_r \end{pmatrix} \tag{12}$$

New definitions for tracking errors, e_1, e_2 and e_3, can be obtained based on the kinematic model of nonholonomic robot in equation (6) as

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{pmatrix}. \tag{13}$$

Since the above transformation is invertible, it follows that

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} \to 0 \;\Rightarrow \begin{pmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\theta} \end{pmatrix} \to 0 \quad \text{as } t \to \infty. \tag{14}$$

In [20], it is shown that with a proper selection of constant control gains, $k_1 > 0$ and $k_2 > 0$, all tracking errors can be regulated to zero using the controls u_1 and u_2 for time-variant reference trajectories:

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -k_1 e_1 + u_{1r}\cos(e_3) \\ -u_{1r}(\sin(e_3)/e_3) - k_2 e_3 + u_{2r} \end{pmatrix} \tag{15}$$

4. Formation control of mobile robots

In order to simplify the formation control problem, we first treat robots as point particles. This is a useful modeling assumption to start with because it eliminates the complex geometry and the nonholonomic dynamics of the mobile robots from the analysis. The kinematic model for the i^{th} robot can therefore be written as

$$\begin{pmatrix} \dot{x}_i \\ \dot{y}_i \end{pmatrix} = \begin{pmatrix} u^i_{\text{total}} \\ v^i_{\text{total}} \end{pmatrix}, \tag{16}$$

where $\dot{x}_i$ and $\dot{y}_i$ are the velocities of the particle i in the x and y directions, respectively, with respect to the world coordinate frame, and controls $\begin{pmatrix} u^i_{\text{total}} \\ v^i_{\text{total}} \end{pmatrix}$ are defined as the sum of the control components designed for the formation control and coordination control, namely

$$\begin{pmatrix} u^i_{\text{total}} \\ v^i_{\text{total}} \end{pmatrix} = \begin{pmatrix} u^i_{\text{Form}} \\ v^i_{\text{Form}} \end{pmatrix} + \begin{pmatrix} u^i_{\text{Coord}} \\ v^i_{\text{Coord}} \end{pmatrix}, \tag{17}$$

where $\begin{pmatrix} u^i_{\text{Coord}} \\ v^i_{\text{Coord}} \end{pmatrix}$ is the coordination control component and $\begin{pmatrix} u^i_{\text{Form}} \\ v^i_{\text{Form}} \end{pmatrix}$ is the formation control component. In the proposed method, these two components are designed separately.

The coordination control component is introduced to prevent the collision of the robots and maintain a desired distance between each robot and its nearest two neighbors.

For the formation control, two phases are considered. At the beginning of the first phase, the robots are positioned randomly in a predefined area. The aim of this phase is to achieve the desired formation shape. The second phase begins immediately after the desired formation shape is formed. In this phase, the robots are expected to maintain the formation shape by tracking the formation shape trajectory. The design of the formation control components in these two phases is quite different. Implicit and parametric representations of the formation shape are used in the first and second phases, respectively. Coordination component exists in both phases.

4.1. Coordination control

In addition to the shape formation task robots need to move in a coordinated manner. The aim of this coordination is to avoid collisions and keep a desired distance between each robot and its neighbors. The interaction between each robot and its two nearest neighbors is modeled by a spring as shown in Figure 3. To determine two nearest neighbors, the distance between one robot and all others can be measured, assuming positions of all others are available from sensory data, and sorted. Robot vehicles p and q are then considered as the nearest neighbors if their distances from the underlying robot are the smallest. This can easily be done in simulations, since we know positions of all robots. However, in practice the nearest neighbors will be determined by onboard sensors. Consider a mobile robot which carries a vision system or a laser range finder or some other sensor. In such a case, two nearest neighbors will be two closest robots that are sensed by the robot.

In Figure 3, p and q are the two closest/nearest neighbors of the robot i. The spring has a normal length which is equal to the desired distance between the robots. The spring is assumed to produce a force linearly proportional to the difference between the actual and the desired distances between the robots. The spring force is along the direction of the vector from the i^{th} robot to its neighbor; i.e.,

$$\begin{pmatrix} u^i_{\text{Coord}} \\ v^i_{\text{Coord}} \end{pmatrix} = k(d_{des} - d^{ip}_{\text{actual}}) \begin{pmatrix} x_i - x_p \\ y_i - y_p \end{pmatrix} / \left\| \begin{matrix} x_i - x_p \\ y_i - y_p \end{matrix} \right\| + k(d_{des} - d^{iq}_{\text{actual}}) \begin{pmatrix} x_i - x_q \\ y_i - y_q \end{pmatrix} / \left\| \begin{matrix} x_i - x_q \\ y_i - y_q \end{matrix} \right\| \tag{18}$$

where k is an adaptable spring constant; the spring constant is different in various phases of the coordination. This constant can be tuned according to the desired stiffness of the coordination. For example, when robots are far away from the desired formation shape, k may assume relatively large values. On the other hand, if they are going to form the desired formation shape, k may assume smaller values. p and q are the indices of the robots that are the nearest two neighbors of i^{th} robot. d^{ip}_{actual} and d^{iq}_{actual} are the actual distances of robot i from robots p and q, respectively. (x_p, y_p) and (x_q, y_q) are the x and y position coordinates of the robots p and q. d_{des} is the desired distance between the robots.

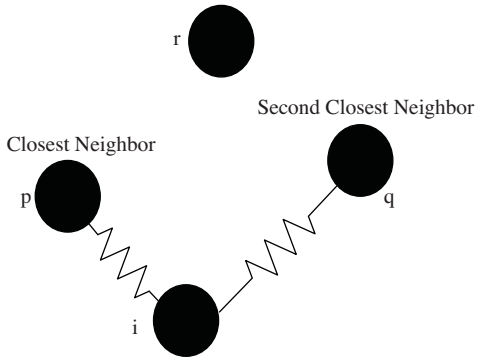

Figure 3. Modeling describing the coordination control involving the two nearest neighbors.

This coordination model based on spring action will imply large initial forces that could be difficult to stabilize if the initial distances between robots are large. However, we don't have to take such spring forces into account if there are large initial distances between the robots. We can simply deactivate spring forces if

the distance between the robots exceeds a threshold value. We should remark that linear spring model makes sense if the deflection is small. In practice nearest neighbors will be sensed either by onboard sensors or some external vision sensor such as a top camera. Therefore, coordination in terms of spring forces is activated only when the two nearest robots are inside a circular region with radius less than the threshold value mentioned above.

We should also remark that coordination could also be modeled in terms of a spring and a damper. Injecting damping into the system can stabilize undesired oscillations that may take place in the motion of the robots.

4.2. Formation control using implicit polynomial (IP) potential functions

In the first phase of the shape formation, robots are randomly positioned in a predefined area and the aim is to achieve the desired formation shape. Desired formation shape can be defined in many different ways depending on the task to be performed. For example, it could be the boundary of a certain geographical area extracted from a camera image provided by an unmanned aerial vehicle (UAV).

For the design of the formation component of control input, the implicit polynomial representation of the desired shape is used. The implicit polynomials are suitable for finding whether a point is on a curve/surface or not. Besides, the implicit polynomials can be used as a representation for the distance between a point and a curve[13]. In this study, the position error between the robot i and the formation curve is given by the *algebraic distance* of the robot to the curve which is defined as

$$e^i_{\text{Form}} = F(x_i, y_i), \tag{19}$$

where e^i_{Form} is the position error with respect to the desired curve. According to this representation, the error is zero when the robot is on the curve. If the robot is inside the desired curve, the error is negative/positive, if it is outside the curve the error has the opposite sign of this error, i.e. positive/negative. The control is designed to obtain an exponentially decaying error, i.e.

$$\dot{e}^i_{\text{Form}} = -\lambda e^i_{\text{Form}} \tag{20}$$

where λ is a positive number. Substituting $e^i_{\text{Form}} = F(x_i, y_i)$ into the equation above yields

$$\dot{F}(x_i, y_i) = -\lambda F(x_i, y_i). \tag{21}$$

Using chain rule of differentiation, we have

$$F_x(x_i, y_i)\dot{x}_i + F_y(x_i, y_i)\dot{y}_i = -\lambda F(x_i, y_i). \tag{22}$$

This equation can be written in vector form as

$$\begin{pmatrix} F_x(x_i, y_i) & F_y(x_i, y_i) \end{pmatrix} \begin{pmatrix} \dot{x}_i \\ \dot{y}_i \end{pmatrix} = -\lambda F(x_i, y_i) \tag{23}$$

$$\Rightarrow \begin{pmatrix} F_x(x_i, y_i) & F_y(x_i, y_i) \end{pmatrix} \begin{pmatrix} u_{\text{Form}} \\ v_{\text{Form}} \end{pmatrix} = -\lambda F(x_i, y_i). \tag{24}$$

Note that we have two unknowns, u_{Form} and v_{Form}, but one equation. The minimum norm solution for the formation control can be found using the *pseudo inverse*; i.e.

$$\begin{pmatrix} u^i_{\text{Form}} \\ v^i_{\text{Form}} \end{pmatrix} = -\lambda \frac{1}{||\nabla F(x_i, y_i)||^2} F(x_i, y_i) \begin{pmatrix} F_x(x_i, y_i) \\ F_y(x_i, y_i) \end{pmatrix}. \tag{25}$$

4.3. Maintaining formation using elliptic Fourier descriptors (EFD)

After the robots reach the desired formation, a new control is used to keep the robots in this formation by tracking the formation trajectory. Since EFD representations are parametric, they are more useful for trajectory tracking than the implicit representations. In the design of the control for maintaining the formation, the positions on the desired formation are described by using the parametric representation of the formation shape, namely

$$\begin{aligned} x^*(t) &= a_0 + \sum_{k=1}^{n} a_k \cos(k\omega t) + b_k \sin(k\omega t) \\ y^*(t) &= c_0 + \sum_{k=1}^{n} c_k \cos(k\omega t) + d_k \sin(k\omega t). \end{aligned} \tag{26}$$

Using this description, the error for keeping the formation shape is defined as the difference between the desired position and the actual position of the robot as

$$e^i_{\text{Form}} = \begin{pmatrix} x^*_i(t) \\ y^*_i(t) \end{pmatrix} - \begin{pmatrix} x_i(t) \\ y_i(t) \end{pmatrix}, \tag{27}$$

where $x^*_i(t)$ and $y^*_i(t)$ are the desired positions and $x_i(t)$ and $y_i(t)$ are the actual positions. The control is designed to make the error decrease exponentially, namely

$$\dot{e}^i_{\text{Form}} = -\lambda e^i_{\text{Form}}. \tag{28}$$

In light of Equation (27), we have

$$\dot{e}^i_{\text{Form}} = \begin{pmatrix} \dot{x}^*_i(t) \\ \dot{y}^*_i(t) \end{pmatrix} - \begin{pmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \end{pmatrix}, \tag{29}$$

which implies that

$$\Rightarrow \dot{e}^i_{\text{Form}} = \begin{pmatrix} \dot{x}^*_i(t) \\ \dot{y}^*_i(t) \end{pmatrix} - \begin{pmatrix} u^i_{\text{Form}} \\ v^i_{\text{Form}} \end{pmatrix}. \tag{30}$$

Using Equations (28) and (30), the formation control component is determined as

$$\begin{pmatrix} u^i_{\text{Form}} \\ v^i_{\text{Form}} \end{pmatrix} = \lambda e^i_{\text{Form}} + \begin{pmatrix} \dot{x}^*_i(t_i) \\ \dot{y}^*_i(t_i) \end{pmatrix}. \tag{31}$$

4.4. Formation control of nonholonomic mobile robots

The shape formation method presented in the previous sections was designed for point-particle model. This method can be extended to the formation control of nonholonomic mobile robots by using the positions calculated from point particle model as references for the nonholonomic robots. Figure 4.4 shows a block diagram which depicts the shape formation control method for a nonholonomic mobile robot. In this figure, EFD and IP representations of the desired formation shape are generated inside the block denoted "Desired Curve Representation," where equations (4) and (5) are implemented. The block labeled "Positions of Two Nearest Neighbors" provides positions of two closest neighbors of the robot. It determines these positions based on the measured distances between each robot and all other robots, which is just a search process. As we mentioned earlier, in a practical scenario two nearest neighbors are determined using sensory data from onboard sensors. Output of these two blocks and the (x, y) position of the nonholonomic robot are input to the block labeled "Point Particle Control," which implements shape formation control by computing u and v according to equations (17), (18) and (25). Note that the desired point particle positions, x^* and y^*, are calculated as the integral of these control inputs. The initial positions of the point particle robots are the same as the nonholonomic robots.

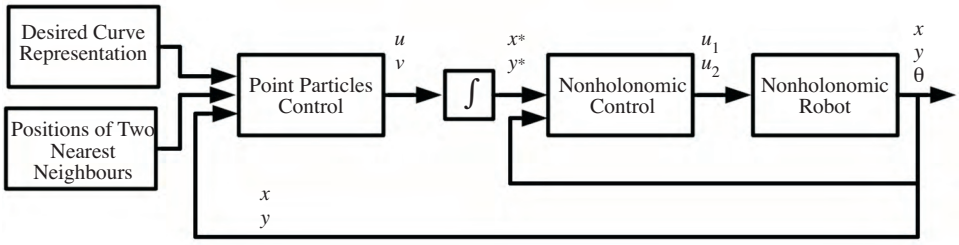

Figure 4. Shape Formation Control for Nonholonomic Robot

The block denoted "Nonholonomic Control" calculates the control inputs u_1 and u_2 for the nonholonomic mobile robot using the positions of the point particle robots and the feedback measurement of the pose (x, y and θ) of the nonholonomic robot. This block implements the trajectory tracking control law given by equation (15). If these control inputs are above the physical limits of the actuators, they will be saturated. Computation of the control law in equation (15) requires knowledge of u_{1r}, u_{2r} and θ_r, which can be computed using equation (9), which in turn necessitates computation of the first and second derivatives of x_r and y_r. As explained earlier, these derivatives are computed using Euler's backward difference on the "filtered" position obtained from the position signal by a low-pass filter.

5. Simulation results and discussions

Simulations are performed in Matlab/Simulink. The program is written to be modular so that the simulations can be carried out with any desired number of robots. Two different simulations will be presented. First one is a simulation of a single robot to see the efficiency of the proposed formation control. The second one is done with 5 and 6 robots on two different desired formation shapes to see the success of the control in both formation and coordination.

The parameter λ which appears in the control laws determines the rate of the exponential decay of the formation error. It can be selected as any positive value. Larger values imply fast convergence. We set $\lambda = 3$

in all simulations. The length of the spring is chosen to be $d_{\text{desired}} = 0.1$ units. This parameter can assume any reasonable value that should be no less than the distance between the centers of two actual robots. Recall that the spring parameter was an adaptable one and could be selected differently during various phases of formation control. When the robots are close to the desired shape, they may need to relax spring forces by choosing a smaller value for k. On the other hand, when they are far away from the desired formation k can be set to a larger value. With these considerations in mind, two different spring coefficients are considered: $k_1 = 2$ when the robots are far away from the formation and $k_2 = 1$ when they are close to the formation. Of course one can select relatively large or small values for these spring coefficient. One thing to note is that larger values of k may imply unnecessarily large forces that can be hard to stabilize. Damping may be needed to suppress undesired oscillations.

5.1. Simulations with point particle robots

In the first simulation, the proposed method is tested with a single robot. Desired formation shape is represented by a Fourier descriptor using 7 harmonics which implies a corresponding implicit polynomial of degree 14. The initial position of the robot is: $x = 2.5$, $y = -2$. The route of the robot is depicted in Figure 5.

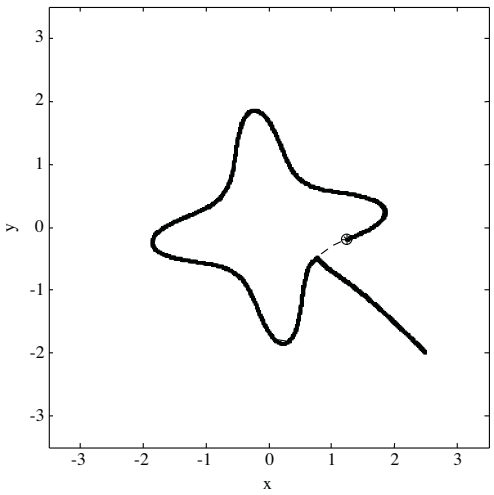

Figure 5. Route of point particle mobile robot.

Note that the proposed formation control approach enables the robot to reach the desired formation curve and then moves on it without any problem.

Next, the method is tested with a group of five robots. The desired formation shape is an ellipse generated by a Fourier descriptor with 1 harmonic and a corresponding implicit polynomial of degree 2. Ellipse formation is shown in Figure 6. Note that robots approach the desired formation curve by keeping the desired distance between their two nearest neighbors. After reaching the curve, they start traveling around that curve. It can also be observed that a desired distance is kept between the robots while moving on the formation curve.

The last simulation in this section is performed with a group of six point particle robots. Formation shape is a more complicated star-shaped curve represented by 7 harmonics and a corresponding implicit polynomial of degree 14. Formation results are depicted in Figure 7. It is clear that the proposed method enables robots to achieve and maintain the desired formation while keeping good coordination with each other.

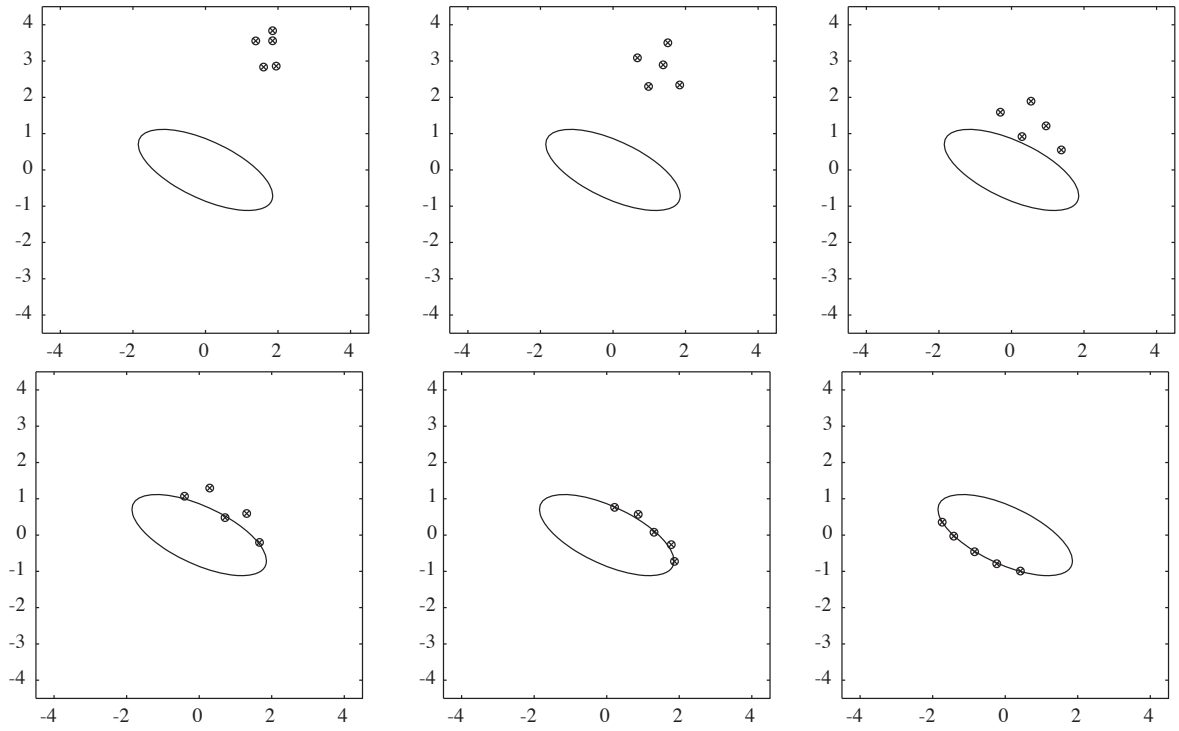

Figure 6. Desired (ellipse) formation with five point particle robots

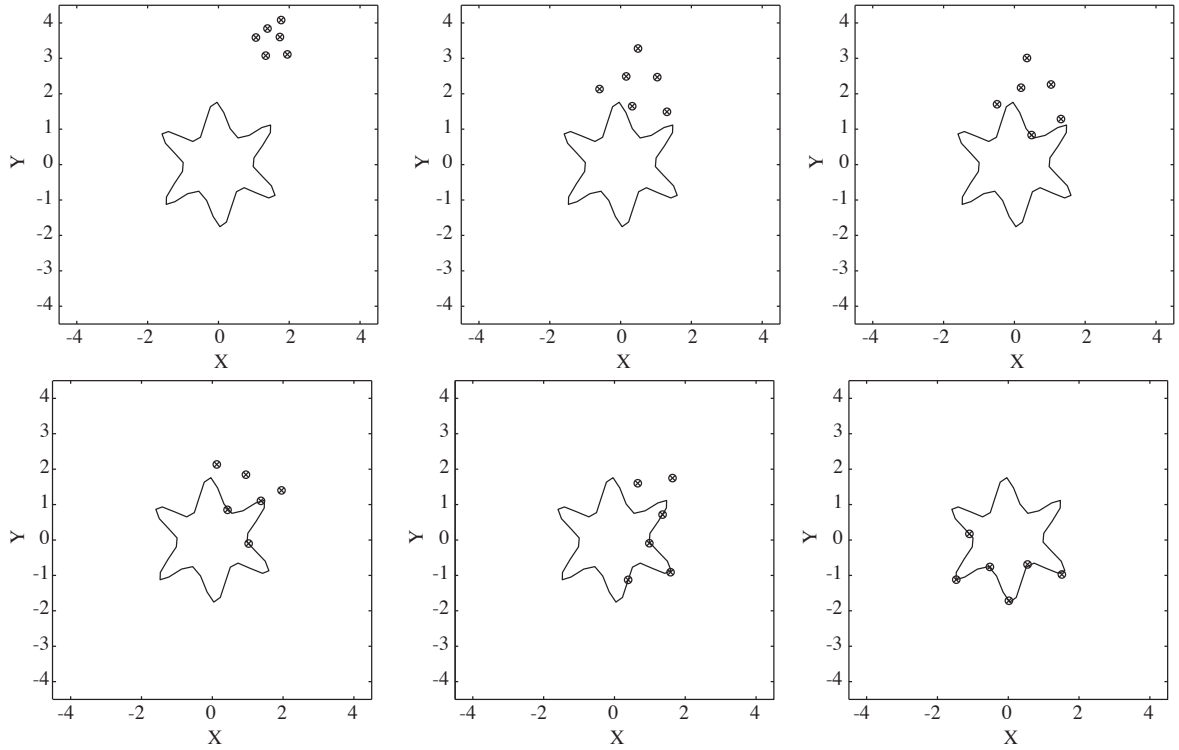

Figure 7. Desired (star shape) formation with six point particle robots.

5.2. Simulations with nonholonomic mobile robots

The aim of the simulations in this section is to see the success of the proposed shape formation control with nonholonomic mobile robots. The position references for the nonholonomic mobile robots are obtained from the positions of the point particle robots, which were simulated in the previous section.

First, a single nonholonomic mobile robot is simulated. This robot starts from the position $x = 2.5$, $y = -2$. It can be seen from Figure 8 that the robot has reached the desired curve successfully. Some oscillations can be seen at the point where the robot reaches the formation shape. These oscillations are caused by the nonholonomic constraints on the velocity of the robot.

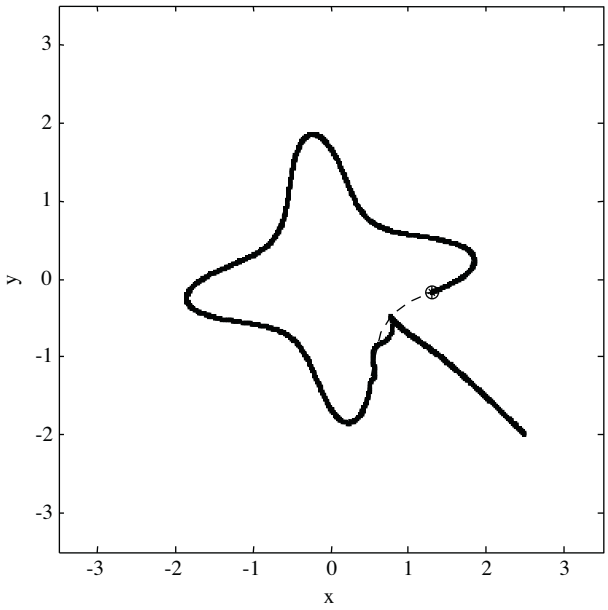

Figure 8. Route of the nonholonomic mobile robot.

Next, a group of five nonholonomic robots is simulated. In the simulation, the desired formation curve is an ellipse which is represented by EFD with 1 harmonic and a corresponding IP of degree 2. Sample snapshots from this simulation are shown in Figure 9. Note that robots have reached the formation curve and traveled on it in a coordinated manner.

Finally, a group of six nonholonomic mobile robots are considered. The desired formation shape is a more complex one which is represented by EFD with 7 harmonics and a corresponding IP of degree 14. Sample snapshots from these simulations are presented in Figure 10. Note again that the nonholonomic robots have reached the curve and traveled on it by maintaining the coordination.

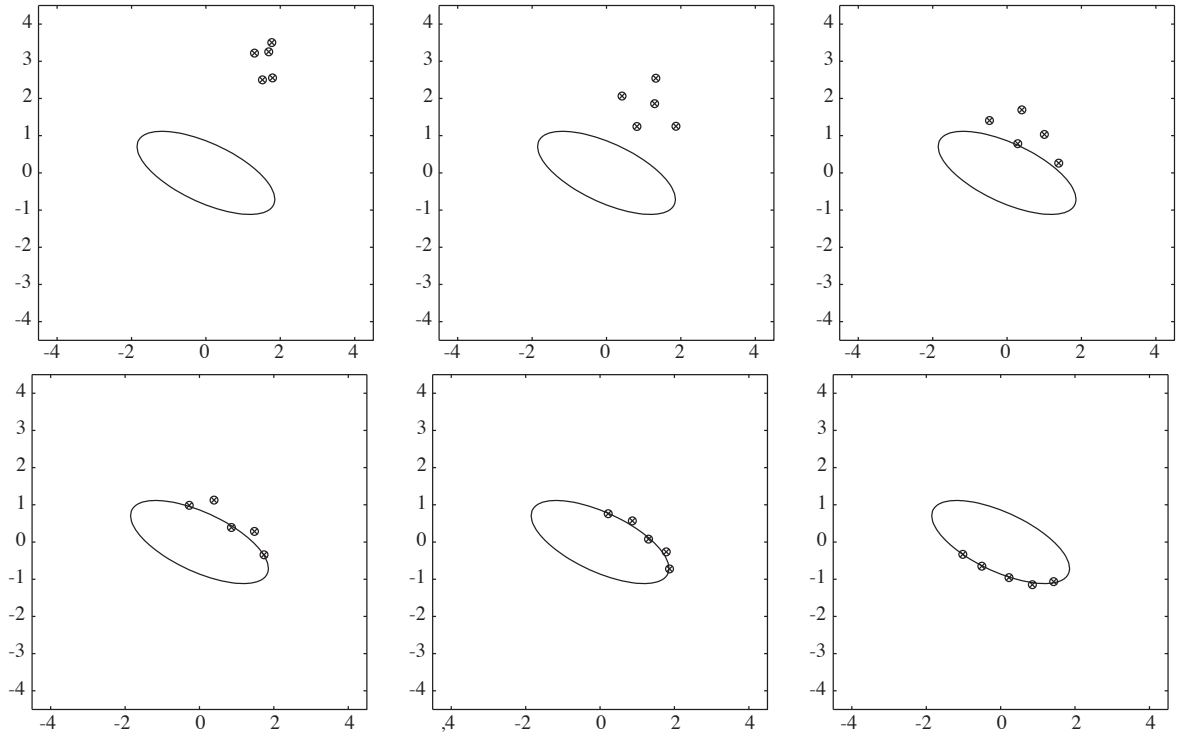

Figure 9. Desired formation (ellipse) with five nonholonomic robots

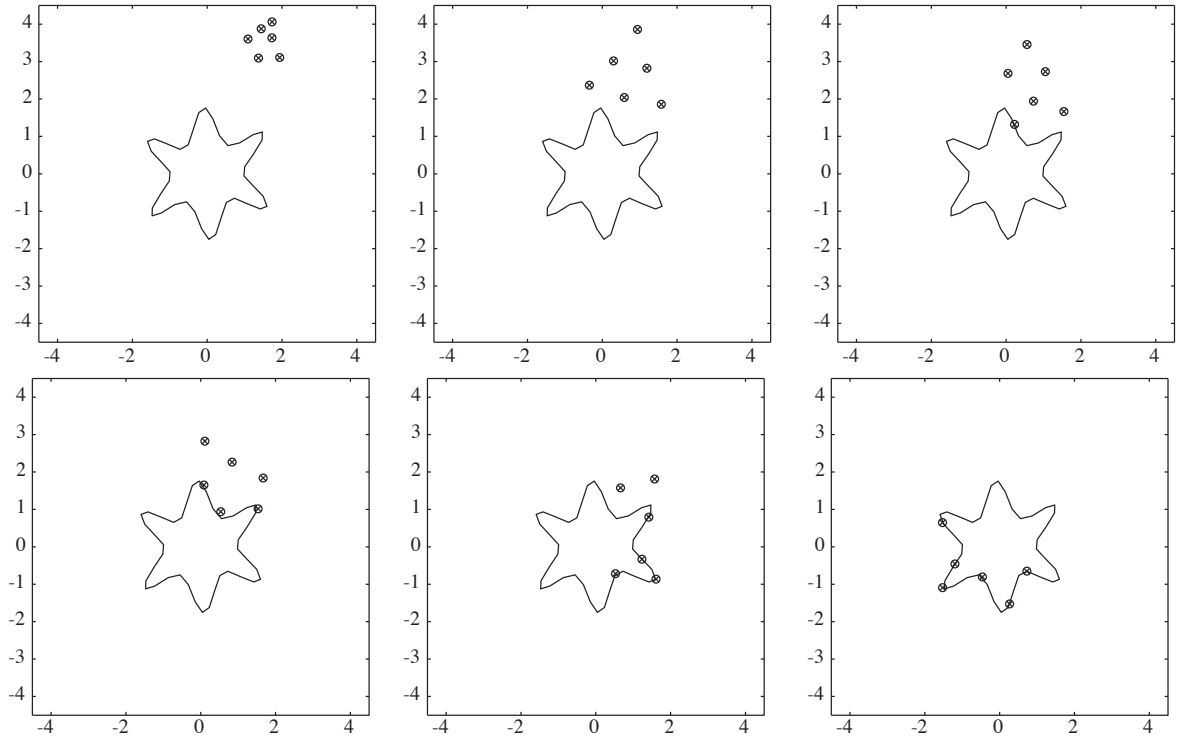

Figure 10. Desired formation (star shape) with six nonholonomic robots

6. Conclusions

We have now presented a novel method for a group of nonholonomic mobile robots to form arbitrary desired shapes using parametric (elliptic Fourier descriptors) and implicit (implicit polynomials) descriptions. The main advantages of our approach are increased flexibility in the formation shape, scalability to large number of robots and efficient implementation. Formation control algorithms are first developed for point particle robots and then extended to the nonholonomic mobile robots. Simulations performed in Matlab/Simulink demonstrated effectiveness of the formation control.

As a future work, the method will be extended to 3D shape formation.

References

[1] D. Fox, W. Burgard, H. Kruppa, S. Thrun, "A Probabilistic Approach to Collaborative Multi-Robot Localization," *Auton. Robots*, vol. 8, no. 3, pp.325-344, 2000

[2] J. Feddema, D. Schoenwald, "Decentralized Control of Cooperative Robotic Vehicles, " Proceedings of SPIE, Orlando, Florida, pp.852-864, 2001

[3] K. Hayashi, Y. Yokokohji, T. Yoshikawa ve W. Moreno, "Tele-existence Vision System with Image Stabilization for Rescue Robots," Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005

[4] A. Birk, S. Carpin, "Merging Occupancy Grid Maps From Multiple Robots," *Proceedings of the IEEE*, vol. 94, no.7, pp.1384-1297, 2006

[5] T. Maneewam, P. Rittipravat, "Sorting Objects by Multiple Robots Using Voronoi Separation and Fuzzy Control," Proceedings of the 2003 IEEWRSJ Intl. Conference on Intelligent Robots and Systems, Las Vegas, Nevada, 2003

[6] T. Sugar, V. Kumar, "Control and Coordination of Multiple Mobile Robots in Manipulation and Material Handling Tasks," *ExperimentalRobotics VI: Lecture Notes in Control and Information Sciences*, vol. 250, pp.15-24, New York: Springer-Verlag, 2000

[7] P. Johnson, J. Bay, "Distributed control of simulated autonomous mobile robot collectives in pay load transportation,", *Autonomous Robot*, pp.43-64, 1995.

[8] T. Balch, M. Hybinette, "Social Potentials for Scalable Multi-Robot Formations," Proceedings of IEEE International Conference on Robotics and Automation, pp.85-94, 2000

[9] H. Yamaguchi, "A cooperative hunting behavior by mobile robot troops," Intl. J. Robotics Research, pp.931-940, 1999.

[10] T. Balch, R.Arkin, "Behavior-based formation control for multirobotic teams," IEEE Transactions on Robotics and Autonomation, pp.926-934, 1998.

[11] J. P. Ostrowski, J. Desai ve V. Kumar, "Controlling formations of multiple mobile robots," Proc. IEEE Int. Conf. Robot Automat., pp.2864-2869, 1998.

[12] P. Song, V. Kumar, "A potential field based approach to multi-robot manipulation," Roc. of the 2002 IEEE Int. Conference on Robotics and Automation, pp.870-876, 2002.

[13] H. Yalcin, M. Unel, W. Wolovich, "Implicitization of Parametric Curves by Matrix Annihilation," *International Journal of Computer Vision* vol. 54, no.1/2/3, pp.105-115, 2003

[14] W. Kang, N. Xi, Y. Zhao, J. Tan, Y. Wang, "Formation Control of Multiple Autonomous Vehicles: Theory and Experimentation," Proceeding,. IFAC 15th Triennial World Congress, pp.1155-1160, 2002

[15] L. Barnes, W. Alvis, M. Fields, K. Valavanis, W. Moreno, "Heterogeneous Swarm Formation Control Using Bivariate Normal Functions to Generate Potential Fields," Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006

[16] Y. H. Esin, M. Unel, M. Yildiz, "Formation Control of Multiple Robots Using Parametric and Implicit Representations," Proceedings of the ICIC'08, Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 558-565, 2008.

[17] F.P. Kuhl, C.R. Giardina, "Elliptic Fourier Features of a Closed Contour," *Computer Graphics and Image Processing*, vol. 18, pp.236-258, 1982.

[18] C. Samson, K. Ait-Abderrahim, "Feedback Stabilization of a Nonholonomic Wheeled Mobile Robot," Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems, pp.1242-12470, 1991.

[19] C. Samson, K. Ait-Abderrahim, "Feedback Control of a Nonholonomic Wheeled Cart in Cartesian Space," Proceedings of the IEEE International Conference on Robotics and Automation, pp.1136-1141, 1991.

[20] C. Samson, "Trajectory tracking for nonholonomic vehicles: overview and case study," Proceedings of the Fourth International Workshop on Robot Motion and Control, pp.139-153, 2004.