2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Controlling Swarms of Mobile Robots for Switching between Formations Using Synchronization Concept

Dong Sun and Can Wang

Department of Manufacturing Engineering and Engineering Management
City University of Hong Kong, Hong Kong
Tel: (852) 2788-8405 Fax: (852) 2788-8423
Email: {medsun, cwang2}@cityu.edu.hk

Abstract — **This paper presents a synchronous control approach to swarms of mobile robots in switching between formations. According to the desired formation, a synchronization control goal is derived, based on which the position synchronization error is defined as differential position errors between very pair of two neighboring robots. A decentralized trajectory tracking controller is then developed with feedback of both position and synchronization errors, formed with a combination of feedforward and feedback controls. It is proven that this tracking controller can asymptotically converge both position and synchronization errors to zero. Simulations are performed on a group of twenty fully-actuated mobile robots in a switching task between different ellipse curves. The simulation results demonstrate the effectiveness of the proposed synchronous control design for the formation control.**

Keywords — **formation, mobile robots, synchronization.**

I. Introduction

Study on coordination of multiple mobile robots has received increasing attentions in recent years. In this paper, we discuss the fundamental issues underlying trajectory controls of swam of robots while maintaining desired formations that may be time-varying.

There are three conventional approaches to coordination of multiple mobile robots reported in the literature: behavior-based method [1-5]; virtual structure techniques [6-9]; and leader-following strategy [10-14]. In behavior-based control, several desired behaviors are prescribed for each agent, and the final control is derived from a weighting of the relative importance of each behavior. The advantage of this strategy is that the group dynamics contain formation feedback by coupling the weightings of the actions. The disadvantage is that it is difficult to describe the dynamics of the group and to guarantee the stability of the whole system [11]. In the virtual structure approach, the entire formation is treated as a single entity. Desired motion is assigned to the virtual structure which traces out trajectories for each member of the formation to follow. The advantage is that the method is easy to prescribe formation strategy and has stability guarantees for the robots. The disadvantage is that the controller is not in decentralized architecture and may encounter difficulty in some applications. In the leader-following strategy, some robots are designed as leaders, while others are designed as followers. The advantage of this strategy is that it is easy to control multiple robots in a desired formation using only two controllers and it is suitable for describing the formation of robots. The disadvantage lies in the difficulty to consider the ability gap of a robot [11].

In this paper, we propose to use a synchronization control strategy to address multirobot coordination, utilizing the concept of cross-coupled approach [15]. The basic idea is to let a team of mobile robots track desired trajectories while using cross-coupled controls to synchronize motions amongst the robots so that a certain kinematics relationship can be maintained for a desired formation. The cross-coupling technology [16] provides advantages and opportunities to design such a synchronized controller. Over the past decade, the cross-coupling concept has been widely used in multi-axis motions and other applications such as reducing contour error of CNC machines [17-20]. Recently, the cross-coupling concept was incorporated into adaptive control architecture to solve position synchronization of multiple axes [21]. The cross-coupling technology has been used in robotics, such as controls of mobile robots [22] and robot manipulators [23]. The solutions for contour tracking problem can be found in [24-25]. To avoid usage of modeling, a model free variable-gain cross-coupling controller was introduced for a general class of contours [26]. The effort to examine the stability and robustness of the cross-coupled control system was reported in [27]. Applications of the synchronization approach to addressing formation control problem have not been reported yet in the literature.

The advantages of using the synchronization control idea for the formation control are threefold. First, the synchronization control goal is determined according to the desired formation, and then divided into a number of sub-goals for each individual robot without discrimination. By this way, the formation strategy can be well prescribed and the ability of each robot is not limited. Second, a synchronization controller that guarantees asymptotic stability of both position tracking and synchronization errors can be designed in a decentralized architecture [21]. Third,

This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project CityU 119706, and by a grant from the City University of Hong Kong under Project 7001923.

the controller can be greatly simplified by synchronizing the motion of each robot with its two neighbors only [23,2]. In other words, each robot control only requires the information of two nearby robots but not all other ones.

This paper has the following two contributions.

First, we successfully transform the two simultaneous actions of trajectory tracking and formation to be a motion synchronization task. A synchronization equation is determined according to the desired formation, based on which a synchronization control goal can be derived. In a similar manner to [21], we drive each robot to approach its target position while synchronizing its motion with its two neighbors. To evaluate the synchronization control effect for the formation, the concept of the synchronization error is introduced, which is defined for each robot as the combination of the position errors of two nearby robots with coupling parameters.

Second, we develop a decentralized cross-coupled controller to stabilize multi-robot motions while synchronizing positions of the robots for the desired formation. After definition of a coupled position error by combining the position error and the integration of the synchronization errors, a simple tracking controller is constructed with feedforward and feedback controls, with information request of two neighboring robots only. It is proven that the proposed controller can guarantee asymptotic convergence to zero of both position and synchronization errors. Simulations are performed on swam of robots in switching between different ellipse curves, to demonstrate the effectiveness of the proposed approach.

The proposed synchronization control approach is more suitable to the tasks of maintenance of formation shape when group robots move as a whole, and switching between formations.

II. Formation via Synchronization

Consider the control problem of guiding and positioning a group of n planar and fully actuated robots along the boundary (curve) of a two-dimensional compact set. The dynamics of the robot i is given by:

$$\ddot{q}_i = u_i \quad (i = 1 \sim n) \tag{1}$$

where $q_i = [x_i, y_i]^T$ is a 2×1 vector containing coordinates in x-y plane, and u_i denotes the control input of the robot i. Introduce a time-varying desired shape, $\mathrm{S(q)}$, where $\mathrm{q} = [q_1^T \cdots q_n^T]^T$ denotes the configuration of the swam of robots. The boundary of $\mathrm{S(q)}$ is parameterized by a two dimensional planar curve, $\partial\mathrm{S(q)} = 0$. Consider a task of switching between different formations $\mathrm{S(q)}$. Assume that each robot is assigned by a target positions q_i^d, and all target positions are located in the curve $\partial\mathrm{S(q^d)} = 0$. The goal is to determine the appropriate control inputs for dynamic (1) such that the n robots converge to these target positions while maintaining the formation shape $\mathrm{S(q)}$.

Define the position error of each robot as $e_i = q_i^d - q_i$. The goal of the robot position control is to drive $e_i \to 0$. Meanwhile, the robots need to achieve the goal of the formation control, namely $\exists q_i, \partial\mathrm{S}(q_i) = 0$. This formation control problem can be solved by using the synchronization control concept. The basic idea is to regulate motions of the robots when they approach the target positions q_i^d, so that all robots maintain in the required boundary curve $\partial\mathrm{S(q)} = 0$.

Define $f(q_1 \cdots q_n)$ as a function of synchronization that all robots are required to satisfy simultaneously. This function is determined based on the formation control goal subject to the constraint $\partial\mathrm{S(q)} = 0$. It actually represents a new task-dependent requirement in kinematics. An example is given below to show how such a synchronization function is derived from $\partial\mathrm{S(q)} = 0$.

Example: Consider that n robots are required to maintain in an ellipse curve during the motion. The coordinate q_i of robot i is subject to the following equation:

$$\partial\mathrm{S}(q_i): q_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \cos\varphi_i & \\ & \sin\varphi_i \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = A_i \begin{bmatrix} a \\ b \end{bmatrix} \tag{2}$$

where a and b denote the longest and the shortest radices of the ellipse, respectively, $\varphi_i = \arctan\left(\frac{b\sin\alpha_i}{a\cos\alpha_i}\right)$, and $\alpha_i = \arctan\left[\frac{y_i}{x_i}\right]$ denotes the angle of the robot location on the ellipse. Assume that the robots are not placed in the longest or the shortest axis of the ellipse so that the inverse of A_i exists. For the desired formation, all robots need to satisfy the following synchronization function simultaneously:

$$f(q_1 \cdots q_n): A_1^{-1} q_1 = A_2^{-1} q_2 = \cdots = A_n^{-1} q_n = \begin{bmatrix} a \\ b \end{bmatrix}. \qquad \blacksquare$$

From the above example, it is seen that the synchronization function can be generally represented in the form:

$$f(q_1 \cdots q_n): c_1 q_1 = c_2 q_2 = \cdots = c_n q_n \tag{3}$$

where c_i denotes the coupling coefficient of the robot i and is assumed to be nonzero. (That c_i equals zero means the robot has no formation requirement and thus needs not the formation control.) Note that c_i is either constant or time-varying. Since eq. (3) holds at all desired coordinates q_i^d, we have

$$f(q_1^d \cdots q_n^d): c_1 q_1^d = c_2 q_2^d = \cdots = c_n q_n^d. \tag{4}$$

Then, the following synchronization goal can be derived by combining (3) and (4):

$$c_1 e_1 = c_2 e_2 = \cdots = c_n e_n \tag{5}$$

Eq. (5) actually represents the formation control goal implicitly. Further, the synchronization goal (4) can be divided into n sub-goals such as $c_i e_i = c_{i+1} e_{i+1}$, with a boundary condition that when $i=n$, $i+1=1$.

We then introduce the concept of position synchronization errors, which are defined as a subset of all possible pairs of two neighboring robots in the following way:

$$\begin{aligned} \varepsilon_1 &= c_1 e_1 - c_2 e_2 \\ \varepsilon_2 &= c_2 e_2 - c_3 e_3 \\ &\vdots \\ \varepsilon_n &= c_n e_n - c_1 e_1 \end{aligned} \tag{6}$$

where ε_i denotes the synchronization error of the robot i. Obviously, if the synchronization error $\varepsilon_i = 0$ for all $i = 1 \sim n$, the synchronization goal (5) can be achieved automatically. The synchronization error represents the degree of coordination amongst the actuated robots in the formation, and is not equivalent to the conventional tracking error. Employment of the synchronization error provides each robot with motion information both from itself and from the other robots. Hence, the motions of all robots are well coordinated.

Now the control problem is to drive n robots converge to each target position so that $e_i = 0$, while achieving the formation control goal (5) by regulating the synchronization errors in (6) to zero. The next objective is to design a controller that guarantees asymptotic convergence to zero of both the position error e_i and the synchronization error ε_i. In a similar manner to [21], the robot i can be designed to approach its target position q_i^d while synchronizing its motion with its two neighboring robots. By this way, the control of each robot does not require the information of all robots except for two neighbors, and hence the implementation is greatly simplified.

The above development shows how to make a formulation to pose the formation control problem as a synchronization control problem. The synchronization errors are defined based on the formation control goal $\partial \mathrm{S}(\mathrm{q}) = 0$, and thereby represent the formation errors implicitly. The approach is extensible to an arbitrary formation shape if it can be constrained by (3).

III. Synchronization Control Design

Define a coupled position error as

$$E_i = c_i e_i + \beta \int_0^t (\varepsilon_i - \varepsilon_{i-1}) dw \tag{7}$$

where β is a positive constant, and w is a variable from time zero to t. The synchronization error ε_i in (7) is subject to the boundary condition that when $i=1$, $i-1=n$. Note that the coupled position error E_i of the robot i contains the information of two neighboring robot $i-1$ and $i+1$, which can be seen from eq. (7) and the definition of ε_i in (6).

Differentiating E_i with respect to time yields

$$\dot{E}_i = \dot{c}_i e_i + c_i \dot{e}_i + \beta (\varepsilon_i - \varepsilon_{i-1}) \tag{8}$$

To construct an asymptotically stable tracking controller to drive $E_i \to 0$ and $\dot{E}_i \to 0$, we utilize [28] to introduce a command vector u_i that leads to combined position and velocity errors. Unlike [28], here there exists a time-varying coupling parameter c_i in addition to e_i in (7). Hence, we may define the command vector u_i as follows:

$$u_i = c_i \dot{q}_i^d + \dot{c}_i e_i + \beta (\varepsilon_i - \varepsilon_{i-1}) + \Lambda E_i \tag{9}$$

where Λ is a diagonal positive gain matrix. Definition of u_i in (9) leads to the following position/velocity vectors:

$$\begin{aligned} r_i &= u_i - c_i \dot{q}_i = c_i \dot{e}_i + \dot{c}_i e_i + \beta (\varepsilon_i - \varepsilon_{i-1}) + \Lambda E_i \\ &= \dot{E}_i + \Lambda E_i \end{aligned} \tag{10}$$

Now the control objective is to design the torque input τ_i to restrict r_i to lie on the sliding surface [28], so that the coupled errors E_i and $\dot{E}_i$ tend to zero. For easy implementation, the torque control input is ideally designed in a decentralized architecture, and each robot control only considers the synchronization between this robot and its two neighbors but not all others. Further, adding the synchronization control would not affect the stability of the whole system.

We finally design the controller as follows:

$$\tau_i = c_i^{-1} (\dot{u}_i - \dot{c}_i \dot{q}_i) + K_{ri} c_i^{-1} r_i + c_i^T K_\varepsilon (\varepsilon_i - \varepsilon_{i-1}) \tag{11}$$

where K_{ri} and K_ε are positive feedback control gains. The last term in (11) is used to compensate for the effect of introducing the cross-coupling control on the overall system dynamics, which is required by the stability analysis.

It appears that parameters β in (7) and K_ε in (11) will dominate the control of the synchronization error. β plays a major role to reduce the synchronization error, while K_ε ensures the stability of the system when adding the cross-coupling control with β on the overall system dynamics.

Substituting (11) into the robot dynamics (1) yields the closed-loop dynamics:

$$c_i^{-1} \dot{r}_i + K_{ri} c_i^{-1} r_i + c_i^T K_\varepsilon (\varepsilon_i - \varepsilon_{i-1}) = 0 \tag{12}$$

Theorem 1 The proposed tracking controller (10) leads to asymptotic stability of the system, namely, $e_i \to 0$ and $\varepsilon_i \to 0$ as time $t \to \infty$, under the condition that the control gain K_{ri} is large enough to satisfy $\lambda_{\min}(K_{ri}) \geq \lambda_{\max}(\dot{c}_i^{-1} c_i)$, where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the minimum and maximum eigenvalues of the matrices.

Proof: Define a Lyapunov function candidate as

$$V=\sum_{i=1}^{n}\left[\frac{1}{2}\left(c_i^{-1}r_i\right)^T c_i^{-1}r_i+\frac{1}{2}\varepsilon_i^T K_\varepsilon \varepsilon_i\right]+\frac{1}{2}\left(\int_0^t\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i-1})dw\right)^T \Lambda\beta K_\varepsilon\left(\int_0^t\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i-1})dw\right) \quad (13)$$

Differentiating V with respect to time yields

$$\dot{V}=\sum_{i=1}^{n}\left[\left(c_i^{-1}r_i\right)^T c_i^{-1}\dot{r}_i+\left(c_i^{-1}r_i\right)^T \dot{c}_i^{-1}r_i+\varepsilon_i^T K_\varepsilon \dot{\varepsilon}_i\right]+\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i-1})^T\Lambda\beta K_\varepsilon\int_0^t\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i-1})dw \quad (14)$$

Multiplying both sides of (12) by $\left(c_i^{-1}r_i\right)^T$ yields

$$\left(c_i^{-1}r_i\right)^T c_i^{-1}\dot{r}_i+\left(c_i^{-1}r_i\right)^T K_{ri}c_i^{-1}r_i+r_i^T K_\varepsilon(\varepsilon_i-\varepsilon_{i-1})=0 \quad (15)$$

Substituting (15) into (14) yields

$$\dot{V}=-\sum_{i=1}^{n}\left[\left(c_i^{-1}r_i\right)^T\left(K_{ri}-\dot{c}_i^{-1}c_i\right)c_i^{-1}r_i\right]-\sum_{i=1}^{n}\left[r_i^T K_\varepsilon(\varepsilon_i-\varepsilon_{i-1})\right]+\sum_{i=1}^{n}\varepsilon_i^T K_\varepsilon\dot{\varepsilon}_i+\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i-1})^T\Lambda\beta K_\varepsilon\int_0^t\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i-1})dw \quad (16)$$

From (7)~(10), we have

$$\begin{aligned}\sum_{i=1}^{n}r_i^T K_\varepsilon(\varepsilon_i-\varepsilon_{i-1})&=\sum_{i=1}^{n}(r_i-r_{i+1})^T K_\varepsilon\varepsilon_i\\&=\sum_{i=1}^{n}\begin{bmatrix}\dot{\varepsilon}_i+\beta(2\varepsilon_i-\varepsilon_{i-1}-\varepsilon_{i+1})+\Lambda\varepsilon_i+\\\Lambda\beta\int_0^t(2\varepsilon_i-\varepsilon_{i-1}-\varepsilon_{i+1})dw\end{bmatrix}^T K_\varepsilon\varepsilon_i\\&=\sum_{i=1}^{n}\dot{\varepsilon}_i^T K_\varepsilon\varepsilon_i+\sum_{i=1}^{n}\varepsilon_i^T\Lambda K_\varepsilon\varepsilon_i+\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i+1})^T\beta K_\varepsilon\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i+1})\\&+\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i+1})^T\Lambda\beta K_\varepsilon\int_0^t\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i+1})dw\end{aligned} \quad (17)$$

Substituting (17) into (16) and utilizing the condition of *Theorem 1*, we have

$$\begin{aligned}\dot{V}\le&-\sum_{i=1}^{n}\left[\left(c_i^{-1}r_i\right)^T\left(\lambda_{\min}(K_r)-\lambda_{\max}\left(\dot{c}_i^{-1}c_i\right)\right)c_i^{-1}r_i\right]-\\&\sum_{i=1}^{n}\varepsilon_i^T\Lambda\beta K_\varepsilon\varepsilon_i-\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i+1})^T\beta K_\varepsilon\sum_{i=1}^{n}(\varepsilon_i-\varepsilon_{i+1})\\&\le 0\end{aligned} \quad (18)$$

Therefore, $r_i\to 0$ and $\varepsilon_i\to 0$ as time $t\to\infty$. The synchronization goal (5) is achieved. From (10), we further have $E_i\to 0$.

We now prove $e_i=0$ when $E_i=0$ and $\varepsilon_i=0$. Combining all equations in (7) from i to n, we have

$$c_1e_1+c_2e_2+\cdots+c_ne_n=0 \quad (19)$$

Substituting (5) into (19) yields

$$c_1e_1=c_2e_2=\cdots=c_ne_n=0.$$

Since c_i is not zero, $e_i=0$. Therefore, the system is asymptotically stable. ■

IV. SIMULATIONS

Simulations were performed in Matlab to verify effectiveness of the proposed synchronization control approach for formation. Fig. 1 illustrates twenty fully-actuated mobile robots standing on an ellipse curve. The robots, denoted by little squares, are required to move as a whole to approach the final desired ellipse curve denoted by dashed line in Fig. 1. During the switch of formations, the robots must maintain a set of ellipse curves with the given time-varying longest and shortest radices. In particular, each robot tracks its individual desired trajectory that is determined according to the desired formation and the index of each robot represented by the angle $\alpha_i=\arctan\left[\frac{y_i}{x_i}\right]$. Meanwhile, all robots must maintain the desired ellipse formation (that is time-varying) during the tracking.

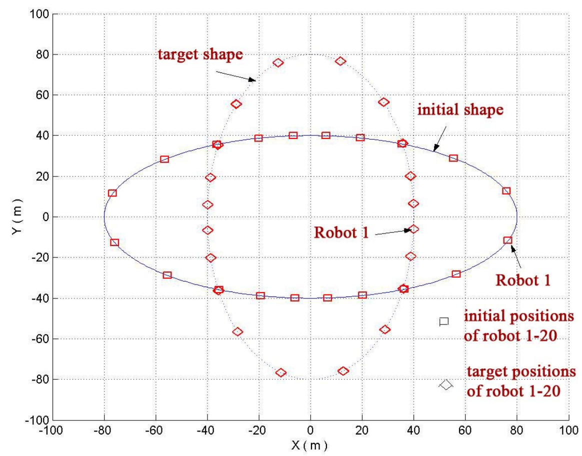

Fig. 1. The desired ellipse formation

Define that the longest and the shortest radices of the ellipse change in the following way:

$$\begin{aligned}a&=a_0-(a_f-a_0)(1-e^{-t})\\b&=b_0+(b_f-b_0)(1-e^{-t})\end{aligned} \quad (20)$$

where a_0 and a_f denote the initial and the final desired longest radices of the ellipse, b_0 and b_f denote the initial and the final desired shortest radices of the ellipse. The equation of ellipse curve has been given in (2). The coupling parameters are thereby defined as $c_i=A_i^{-1}=\begin{bmatrix}\cos\varphi_i & \\ & \sin\varphi_i\end{bmatrix}^{-1}$.

For comparison purpose, two control algorithms have been applied in the simulations. One is the proposed synchronous control. The other one is the non-synchronous control by setting $c_i=1$, $\beta=0$, and $K_\varepsilon=0$. Since c_i has significant effect to the position error e_i as seen in (8), a very large value of c_i may affect the comparison of the two algorithms.

To overcome this problem, we multiplied a scale value of 0.1 to c_i in the simulations.

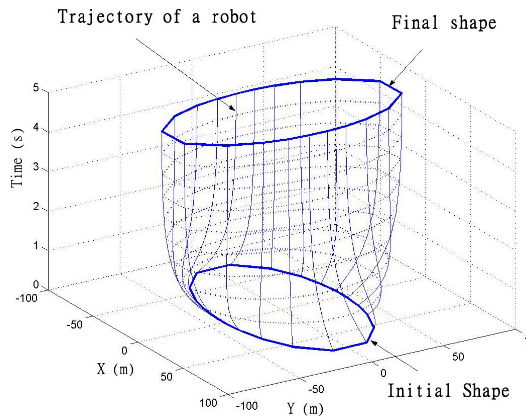

Fig. 2. Robot trajectories and switching between formations.

The control gains were chosen as follows: Λ =20, K_ε =1, $K_r = 100$, and $\beta = 0.5$. When using the non-synchronous control, $\beta = 0$ and $K_\varepsilon = 0$. The sampling period was set to 0.005sec. Fig. 2 illustrates trajectories of all robots, where switching between the initial to final formations is clearly shown. Figs. 3 and 4 illustrate the position errors of all twenty robots under the proposed synchronous control and the non-synchronous control, respectively. It is seen that all position errors converge to zero successfully, and the errors of the synchronous control appear to be smaller than those of the non-synchronous control. These results indicate that the proposed synchronization control can completes the robot trajectory tracking task successfully, with even better motion performance than the non-synchronous control. Figs. 5 and 6 illustrate the synchronization errors with the two control algorithms. Note that these synchronization errors actually represent the formation errors. Apparently, the synchronization errors under the synchronous control are much smaller than those under the non-synchronous control. Table 1 gives a detailed comparison of the synchronization errors of each individual robot under the two control algorithms. Smaller synchronization error implies better formation.

Table 1: The maximum synchronization errors (m)

Robots	**non-synchronous control**	**synchronous control**
Robot 1	1.164	0.011
Robot 2	1.223	0.315
Robot 3	0.530	0.244
Robot 4	0.446	0.119
Robot 5	1.060	0.039
Robot 6	1.223	0.001
Robot 7	1.012	0.043
Robot 8	0.428	0.122
Robot 9	0.587	0.251
Robot 10	1.210	0.307
Robot 11	1.182	0.011
Robot 12	1.219	0.310
Robot 13	0.522	0.245
Robot 14	0.440	0.120
Robot 15	1.055	0.039
Robot 16	1.242	0.001
Robot 17	1.013	0.042
Robot 18	0.440	0.123
Robot 19	0.573	0.249
Robot 20	1.210	0.305

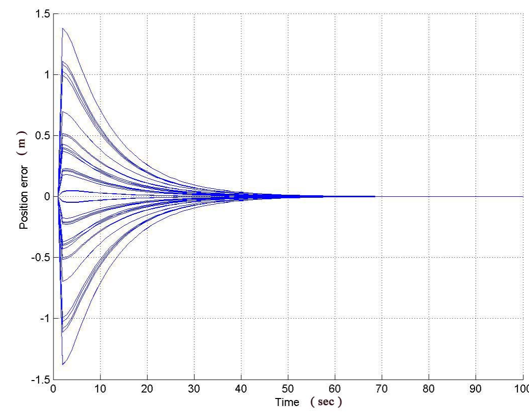

Fig. 3. Position errors with the synchronous control

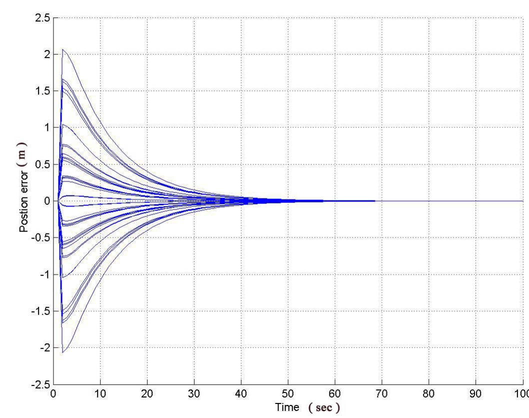

Fig. 4. Position errors with the non-synchronous control

IV. CONCLUSIONS

In this paper, a synchronous control approach is developed for switching between formations of swarms of robots. A position synchronization error is defined as differential position error between every pair of two neighboring robots. A decentralized trajectory tracking controller is then developed with feedback of both position and synchronization errors, formed with a combination of feedforward and feedback control. It is proven that the developed controller guarantees asymptotic convergence to zero of both position and synchronization errors. The simulation results demonstrate the effectiveness of the proposed synchronous control strategy for the formation control. Future work will be an extension of this synchronization control approach to real group of mobile robots with considerations of robot/environment uncertainty and vision/sensing systems.

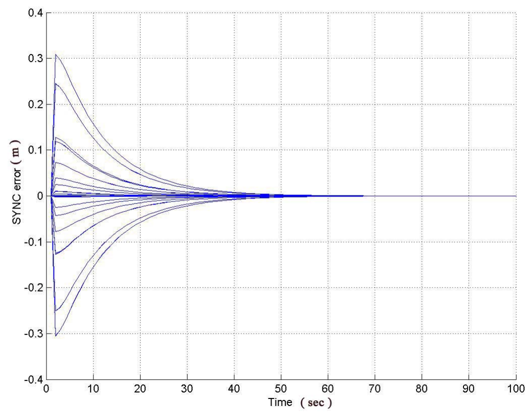

Fig. 5. Synchronization errors with the synchronous control

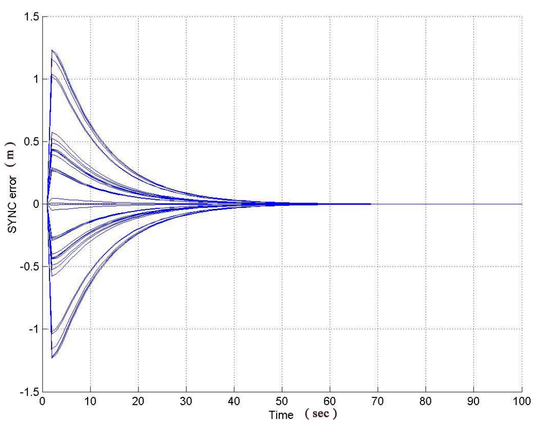

Fig. 6. Synchronization errors without the non-synchronous control

REFERENCES

[1] T. Balch and R. Arkin, "Behavior-based formation control for multi-robot systems," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 926-939, Dec. 1998.

[2] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Trans. Robot. Automat.*, vol. 19, pp. 933-941, Dec. 2003.

[3] L. E. Parker, "ALLIANCE: An architecture for fault tolerant multirobot cooperation," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 220-240, Apr. 1998.

[4] S. Berman, Y. Edan, and M. Hamshidi, "Navigation of decentralized autonomous automatic guided vehicles in material handling," *IEEE Trans. Robot. Automat.*, vol. 19, pp. 743-749, Aug. 2003.

[5] M. Long, A. Gage., R. Murphy, and K. Valavanis, "Application of the distributed field robot architecture to a simulated deming task," in *Proc. IEEE Int. Conf. Robotics and Automation,* pp. 3204-3211, Barcelona, Spain, Apr. 2005.

[6] M. A. Lewis, and K. H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robot.*, vol. 4, pp.387-403, 1997.

[7] M. Egerstedt, and X. Hu, "Formation constrained multi-agent control," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 947-951, Dec. 2001.

[8] R. W. Beard, H. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Trans. Control Syst. Technol.,* vol. 9, pp. 777-790, Nov. 2001.

[9] W. Kang, N. Xi, and A. Sparks, "Formation control of autonomous agents in 3D workspace," in *Proc. IEEE Int. Conf. Robotics and Automation,* pp. 1755-1760. Apr. 2000.

[10] H. G. Tanner, S. G. Loizou, and K. J. Kyriakopoulos, "Nonholonomic navigation and control of multiple mobile manipulators," *IEEE Trans. Robot. Automat.*, vol. 19, pp. 53-64, Feb. 2003.

[11] H. Takahashi, H. Nishi, and K. Ohnishi, "Autonomous decentralized control for formation of multiple mobile robots considering ability of robot," *IEEE Trans. Indus. Eelectronics*, vol. 51, pp. 1272-1279, Dec. 2004.

[12] J. P. Deasi, V. Kumar, and P. Ostrowski, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 905-908, Dec. 2001.

[13] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 813-825, Oct. 2002.

[14] J. Huang, S. M. Farritor, A. Qadi, and S. Goddard, "Localization and follow-the-leader control of a heterogeneous group of mobile robots," *IEEE/ASME trans. Mechatron.*, vol. 11, no. 2, pp. 205-215, Apr. 2006.

[15] Y. Koren, "Cross-coupled biaxial computer controls for manufacturing systems," *ASME J. Dynam. Syst., Meat., and Contr.*, vol. 102, pp. 265-272, 1980.

[16] M. Tomizuha, J. S. Hu, and T. C. Chiu, "Synchronization of two motion control axes under adaptive feedforward control," *ASME J. Dynam. Syst., Meat., and Contr.*, vol. 114, no. 2, pp. 196-203, 1992.

[17] A. Rodriguez-Angeles and H. Nijmeijer, "Mutual synchronization of robots via estimated state feedback: a cooperative approach," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 4, Jul. 2004.

[18] Q. Zhong, Y. Shi, J. Mo, and S. Huang, "A linear cross coupled control system for high speed machining," *The Int. J. Advanced Manufacturing Technology*, vol. 19, pp: 558-563, 2002.

[19] S. S. Yeh and P. L. Hsu, "Estimation of the contouring error vector for the cross-coupled control design," *IEEE/ASME trans. Mechatron.*, vol. 7, No. 1, March 2002.

[20] M. T. Yan, M. H. Lee, and P. L. Yen, "Theory and application of a combined self-tuning adaptive control and cross coupling control in a retrofit milling machine," *Mechatronics*, vol 15, no. 2, pp. 193-211, Mar. 2005.

[21] D. Sun, "Position synchronization of multiple motion axes with adaptive coupling control." *Automatica*, vol. 39, no. 6, pp. 997-1005, Jun. 2003.

[22] L. Feng, Y. Koren, and J. Borenstein, "Cross-coupling motion controller for mobile robots," *IEEE Control Syst.*, pp. 35-43, Dec. 1993.

[23] Sun D., and Mills, "Adaptive Synchronized Control for Coordination of Multi-Robot Assembly Tasks," *IEEE Trans. Robot. Automat.*, vol. 18, no. 4, pp. 498-510, Aug. 2002.

[24] T. C. Chiu, and M. Tomizuka, "Contouring control of machine tool feed drive systems: A task coordinate frame approach," *IEEE Trans. Control Syst. Technol*, vol. 9, no. 1, Jan. 2001.

[25] R. McNab and T. Tsao, "Multiaxis contour tracking: A receding time horizon linear quadratic optimal control approach," *Dynam. Syst., and Contr.*, vol. 2, Nov. 1994.

[26] Y. Koren, and C. C. Lo, "Variable-gain cross-coupling controller for contouring," *Ann. CIRP*, vol. 40, no. 1, pp. 371-374, 1991.

[27] S. Yeh, and P. Hsu, "Theory and applications of the robust cross-coupled control design," in *Proc. American Control Conf.*, Jun. 1997.

[28] J. J. E. Slotine, and W. Li, "On the adaptive control of robot manipulators", International Journal of Robotics Research, vol. 6, no.3, pp. 49-59, 1987.