The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

A Dynamic Priority Strategy in Decentralized Motion Planning for Formation Forming of Multiple Mobile Robots

Shuang Liu, Dong Sun, Changan Zhu, and Wen Shang

Abstract—**This paper presents a new approach to formation forming of multiple mobile robots with decentralized motion planning. When the robots enter the required formation, there exists the formation-structure constraint, which causes disorder or even deadlock of the formation. A dynamic priority strategy is developed to solve the problem of the formation-structure constraint, and coordinate the robots to form the formation in a proper order. Simulations are performed on a group of mobile robots to demonstrate the validity of the proposed strategy to the formation system.**

I. Introduction

In many applications, robots are required to form formations in accomplishing complex tasks, such as transportation of large awkward objects, mapping, search, and rescue. Motion planning is one of the most important issues for such formation tasks. In this paper, a decentralized planning with a novel dynamic strategy is proposed for formation forming problem.

The existing approaches to multi-robot motion planning are often categorized as either centralized or decentralized planning. In centralized planning, all decisions are made by a single decision maker that computes optimal trajectories based on full knowledge of the environment for all robots [1-5]. To reduce the complexity of problem, a centralized planning often use decoupled methods [6-8], such as the methods of prioritized planning [9-11] and the approach based on a coordination-diagram [12, 13]. Although the decoupled methods reduce the complexity, computational cost arisen from the centralized planning increases as the number of robots increases. To further increase the efficiency, a decentralized planning is often demanded in multi-agent system, in which each robot plans its path and acts independently. An ideal decentralized strategy does not require direct communication amongst the robots, while ensuring collision avoidance and minimal interference of each robot with the purposeful motion of the other robots [14]. In the decentralized planning, the computational complexity does not depend on the number of robots, and some independent planning-based methods can be utilized, such as maze search.

In this paper, the problem of formation forming of multiple robots is discussed, where the goal position of each robot is pre-assigned and all the robots move toward their goals by passing through an unknown environment containing obstacles. Unlike most of the existing formation plans [15-18], there is no knowledge of the whole information of the team and robots cooperate with each other through reactive motion only. A decentralized planning methodology, named Local-Constraint-Path Planner (LCPP) [19], was proposed to plan a local path for an individual robot online. To coordinate the motions of these robots to form the target formation, the constraints amongst the independent planning of robots are firstly presented in a novel tree structure in this paper, and then a dynamic priority strategy is developed to reduce the influence of motion constraints in formation forming.

The paper is organized as follows. Section 2 formulates the decentralized motion planning and multi-robot formation problem. Section 3 presents the development of the proposed decentralized motion planning methodology using the proposed LCPP planner. In Section 4, a dynamic priority strategy is proposed to apply the LCPP to multi-robot formation forming system. Simulations are given in Section 5. Finally, conclusions of this work are given in Section 6.

II. Robot Modeling

Decentralized motion planning is formulated as the combination of motion planning of each robot and coordination of all the robots for collision avoidance. Denote $A_1, A_2, \cdots, A_n$ as n mobile robots in the workspace. The state of each robot A_i is represented by $q_i = [x_i, y_i, \theta_i]^T$, where $i = 1, \cdots, n$ is the index of the robot, x_i and y_i denote the position coordinates, and θ_i the orientation of robot A_i. All the robots are assumed to have online environment sensing capability.

The robot kinematics is represented by a unicycle model as follows

$$\dot{x}_i = v_i \cos\theta_i, \quad \dot{y}_i = v_i \sin\theta_i, \qquad \dot{\theta}_i = w_i \tag{1}$$

This work was supported in part by a grant from Research Grants Council of the Hong Kong Special Administrative Region, China (Reference no. CityU 119907), and a grant from City University of Hong Kong (Reference no. 7002461).

S. Liu is with Control and Mechatronics Group, Suzhou Research Center of City University of Hong Kong and the University of Science and Technology of China, Suzhou, P. R. China (phone: 86-0512-87161379; fax: 86-0512-87161381; e-mail: shuang@mail.ustc.edu.cn).

D. Sun is with Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong, P. R. China (e-mail: medsun@cityu.edu.hk).

C. Zhu is with Department of Precision Machinery and Instrumentations, University of Science and Technology of China, Hefei, P. R. China (e-mail: changan@ustc.edu.cn).

W. Shang is with Suzhou Research Center of City University of Hong Kong, Suzhou, P. R. China (wenshang@cityu.edu.hk).

where v_i and w_i denote the linear and angular velocities of robot A_i, respectively, which can be formulated as

$$v_i = \frac{v_{il} + v_{ir}}{2} \qquad w_i = \frac{v_{il} - v_{ir}}{D_i} \tag{2}$$

where v_{il} and v_{ir} are the velocities of the left and the right driving wheels of robot A_i, respectively; $0 < v_{il}, v_{ir} < V_{\max}$, and $V_{\max}$ represents the maximum linear velocity of the robot; D_i is the distance between the two wheels of the robot. From (2), we can obtain the constraints of v_i and w_i as follows

$$v_i \le \frac{2V_{\max} - w_i D_i}{2} \qquad w_i \le \frac{V_{\max}}{D_i} \tag{3}$$

For simplicity, the robot moving backward problem is not considered in this study.

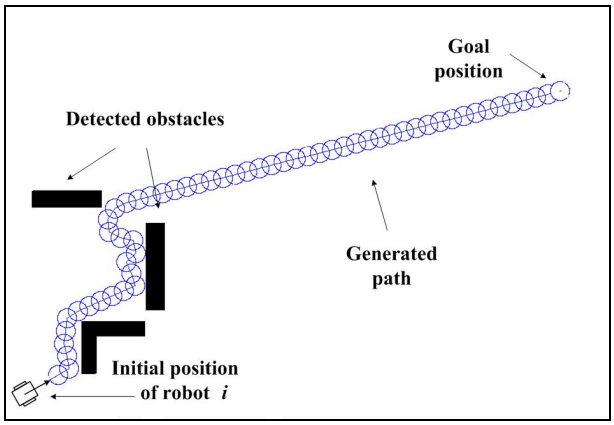
III. Local-Constraint-Path Planner (LCPP)

In this section, we will introduce a Local-Constrained-Path Planner (LCPP) [19] to employ the path planning online. The planner includes three steps. The first step is to determine a set of key waypoints (KWPs) on the way to the goal. The second step is to generate a local constraint path from the current state to the next KWP along the path. The third step is to optimize the planning by shortening the path length through adjusting KWPs.

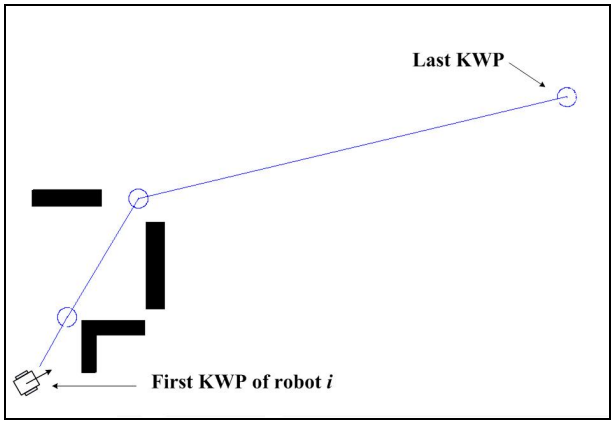
A. Determination of KWPs

The geometric path to the goal can be planned using a bi-directional RRT algorithm [20]. To implement the decentralized motion planning where only local information can be available to each robot, the LCPP planner assumes that there is a clean environment outside the sensing range. As a result, a straight line path is assigned from the currently known KWP to the goal position, as seen in Fig. 2(a).

Define a geometric path l_i of robot A_i as a composition of a set of positions, denoted by $l_i = \{c_{i,1}, c_{i,2}, ..., c_{i,k}, ...\}$, where $c_{i,k} = (x_{i,k}, y_{i,k})$, and k denotes the index of the positions along the path. Amongst all positions in l_i, there exists a few ones that play more important roles than the others in directing the path. We denote these positions as the key waypoints (KWPs). A KWP-pruning function is thereby designed to prune the given geometric path l_i and select the most important positions as KWPs. The initial and the final positions are the first and final KWPs, respectively. The rest positions $c_{i,k}$ are determined as KWPs if they meet the condition that $s_i(c_{i,KWP}^{pre}, c_{i,k}) \notin C_{free}$, where $c_{i,KWP}^{pre}$ refers to the previous KWP with respect to the current position $c_{i,k}$, $s_i(c_{i,KWP}^{pre}, c_{i,k})$ denotes the straight-line path between $c_{i,KWP}^{pre}$ and $c_{i,k}$, and C_{free} denotes the set of positions without any collision to static obstacles. The resulting KWPs based on the example in Fig. 2(a) are shown in Fig. 2(b), where the small circles denote the KWPs determined along the geometric path. This straight line path will be further modified until more environmental information can be known as the robot moves.



(a)



(b)

Fig. 2. Determination of the KWPs with local knowledge of environment. (a) A geometric path generated. (b) Resulting KWPs determined along the geometric path.

B. Local constraint path generation

Each KWP can be treated as a subgoal for the robot when the robot moves toward the final goal. With kinematics constraints in eqs. (1)~(3), the local constraint path can be generated from the current state to the next KWP as follows.

Based on the current state $q_i = [x_i, y_i, \theta_i]^T$, a new state q_i' in the path toward the next KWP, $c_{i,KWP}$, after sampling time period Δt, can be derived as

$$q_i' = \begin{pmatrix} x_i + v_i \int_t^{t+\Delta t} \cos(\theta_i + w_i t) dt \\ y_i + v_i \int_t^{t+\Delta t} \sin(\theta_i + w_i t) dt \\ \theta_i + w_i \Delta t \end{pmatrix} \tag{4}$$

By iterating the derivation of the new state q_i', the path toward the KWP $c_{i,KWP}$ can be generated with the biased

inputs v_i and w_i. To generate the shortest length to the desired goal, the inputs v_i and w_i should be well designed such that the robot can have a quick turn to the target direction and then go along a straight line.

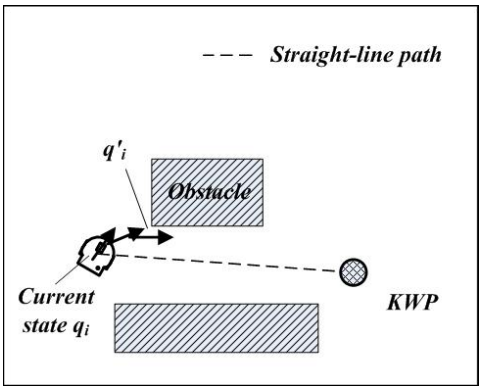
Define the angle of the robot with state q_i relative to the goal position $c_{i,KWP}$ as α_i. From (3), the maximum angle that the robot can turn within the sampling time period Δt can be derived as:

$$\alpha_{i,\max} = \frac{V_{\max}\Delta t}{D_i} \tag{5}$$

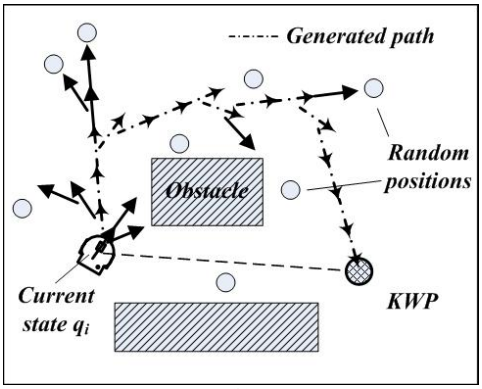
When $\alpha_i = 0$ and $\alpha_i > \alpha_{i,\max}$, the linear velocity v_i and the angular velocity w_i can be easily determined using the maximum value in (3). When $0 < \alpha_i < \alpha_{i,\max}$, the two inputs v_i and w_i should meet the condition that the robot can turn to the target direction within the sampling time period Δt, represented as $\alpha_i = w_i \Delta t$. Substituting $w_i = \frac{\alpha_i}{\Delta t}$ into (3), we can derive the maximum value of v_i. In summary, the solutions of w_i and v_i are derived as follows

$$\begin{cases} w_i = \dfrac{V_{\max}}{D_i} & v_i = \dfrac{V_{\max}}{2} & \alpha_i > \alpha_{i,\max} \\ w_i = 0 & v_i = V_{\max} & \alpha = 0 \\ w_i = \dfrac{\alpha_i}{\Delta t} & v_i = V_{\max} - \dfrac{D_i \alpha_i}{2\Delta t} & 0 < \alpha_i < \alpha_{i,\max} \end{cases} \tag{6}$$

Upon generation of each new state q_i' by inputs v_i and w_i, the system will check whether the path segment from q_i to q_i' has collision to obstacles. If the path segment is free of collision, the new state q_i' is accepted, and the original state q_i is treated as its parent. This process is iterated from the new state q_i' to the KWP $c_{i,KWP}$ again until the path is close to this KWP. If the state q_i' cannot be accepted due to existence of collision, a random search for a new state will have to be done. Fig. 4(a) illustrates that the newly generated state q_i' toward the KWP encounters a collision and is not accepted, and then the random search is performed. Due to kinematic constraints, the robot cannot go to the KWP directly along a straight path practically. In Fig. 4(b), a number of new positions are randomly selected in the environment for random search, and the tree structure is constructed and expanded to these positions, respectively. The upper space around the obstacles is explored by the tree structure, and finally an available path to the KWP is found, as shown in dashed arrows in Fig. 4(b).



(a)



(b)

Fig. 4. Random search in local path generation. (a). Path segment encounter a collision with obstacles. (b) Tree construction for the random search to the KWP.

C. Path optimization

The generated path needs to be further optimized by inspecting whether some KWPs can be discarded such that the path length can be shortened in a possible way. Define the KWPs assigned to robot A_i as $\{c_{i,KWP1}, c_{i,KWP2},..., c_{i,KWPk},...\}$. Consider that $c_{i,KWP1}$ is the next KWP from the current position $c_{i,current}$. If the condition $s(c_{i,current}, c_{i,KWP2}) \in C_{free}$ is satisfied, which implies that the path between the current position $c_{i,current}$ to the KWP $c_{i,KWP2}$ is collision-free, then the KWP $c_{i,KWP1}$ can be discarded, and a new path from the position $c_{i,current}$ directly to the KWP $c_{i,KWP2}$ is planned.

IV. A Dynamic Priority Strategy for Formation Forming

A. Formation-structure constraint

With the LCPP developed above, each robot can plan its own path gradually. However, when entering the formation, the goal positions of the robots may constrain to each other, which causes the robots to disorder in entering the formation. This constraint is defined as *formation-structure constraint* in this paper, and is detailed as follows.

Denote q_i^g and q_j^g as the goal states of robot A_i and A_j,

respectively. If the robot state q_j^g blocks the path of robot A_i, then A_j is defined as a formation-structure constraint robot of A_i. Define F_i as the set containing all constraint robots to A_i, and obviously, $A_j \in F_i$.

An example is shown in Fig. 5. Two robots A_i and A_j are represented by rectangles, labeled by *i* and *j*, respectively, and the dashed lines denote their paths. q_i^g and q_j^g are their goal states in a line formation, respectively. The collision area of state q_j^g is represented by a dashed circle. It can be seen that the goal state q_j^g is on the path of A_i. In other words, if A_j enters the formation before A_i passes the collision area, A_j will block the path of robot A_i. Hence, A_j is a *formation-structure constraint* robot of A_i.

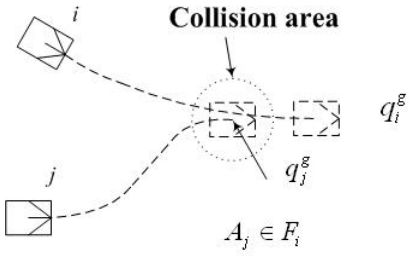


Fig. 5. An example of formation-structure constraint.

According to the formation-structure constraints amongst the robots, a constraint tree can be constructed. The tree reflects the constraint relationship amongst the robots. An example is shown in Fig. 6, where the robots pointed by arrows have the formation-structure constraints to the robots at the head of the arrows. In particular, $A_i \to A_j$ means A_j holds the formation-structure constraint to A_i. Further, if $A_i \to A_j$ and $A_j \to A_k$, we have $A_i \to A_k$. This can also be represented as:

$$\forall A_j \in F_i \text{ and } A_k \in F_j, \text{ then } A_k \in F_i \tag{7}$$

In Fig. 6, the constraint relationships are $F_1 = \{A_2, A_3, A_4\}$, $F_2 = \{A_3, A_4\}$, $F_3 = \phi$, and $F_4 = \phi$.

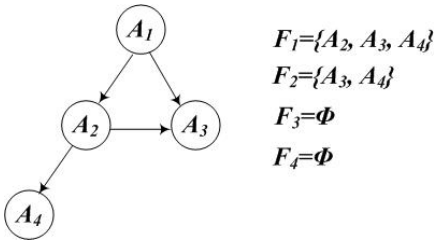


Fig. 6. A formation-structure constraint tree.

When the final positions of the robots in the formation are close, the structure constraint always exists and produces large influence on the robot's path planning. Due to the structure constraint, some robots have to re-plan their paths, which usually enlarge the path lengths. As is seen in Fig. 5, if robot A_j becomes a constraint of robot A_i, robot A_i treats robot A_j as a suddenly appeared obstacle, and re-plans a new path. If there are other robots moving into the formation, the change of planning will cause disorder or even deadlock. To solve this problem, we propose to utilize the priority concept based on the constraint nature. Define p_i and p_j as the priorities of robots A_i and A_j, respectively. If $A_j \in F_i$, as seen in the constraint tree in Fig. 6, then $p_i > p_j$, which implies that robot A_i has the higher priority than robot A_j in forming formation.

B. A dynamic priority strategy

A dynamic priority strategy is introduced to give orders to the robots when entering the formation. All the robots can utilize priorities to coordinate the motions. Based on the formation-structure constraint, we design the dynamic priority as follows

$$p_i = K_i + \sum_{j=1}^{n(j\neq i)} \lambda_j (1 + p_j - K_j) \tag{8}$$

$$\lambda_j = \begin{cases} 0 & A_j \notin F_i \\ 1 & A_j \in F_i \end{cases} \tag{9}$$

where K_i and K_j are small constants of robots A_i and A_j, respectively, used to prioritize the robots when both robots have the same priorities based on constraints. The second term in the right-hand side of (8) is determined based on the formation-structure constraints tree. If the robot has many sub-trees, its priority is calculated by accumulating priorities of all robots that belong to F_i. Therefore, the robot with more sub-trees employs higher priority.

Assume that robot A_i can know the state q_j and the velocity v_j of robot A_j when A_j enters the sensing area of A_i, and further, can obtain the information of KWPs of robot A_j through information exchange. If there exists a formation-structure constraint between A_i and A_j, they need to coordinate their trajectory plans to avoid collision. Define $T_{i,g}(t)$ and $T_{j,g}(t)$ as times that robots A_i and A_j need to spend to reach their goals at time t without any constraint. If $p_i > p_j$ and $T_{i,g}(t) > T_{j,g}(t)$, then $A_j \in F_i$ and A_j will be likely to reach its goal before A_i. Then robot A_j needs to give the way to robot A_i and let A_i move first, until to meet the condition $T_{i,g}(t) < T_{j,g}(t)$. In this way, the robot with higher priority reaches its goal earlier than the robot with lower priority. If $p_i > p_j$ and $T_{i,g}(t) < T_{j,g}(t)$, A_i and A_j continue to move unless they have a mutual collision. When the mutual collision is detected, robot A_j with lower priority gives the way to the robot A_i with higher priority.

To check the mutual collision between A_i and A_j, a secure distance D_{sec} is determined in such a way that the two robots will not have collision within an expected time period when moving with the maximum velocities.

C. Evaluation of the constraint

Two criteria are used to measure efficiency of the dynamic priority strategy. One is the increment of path length ΔL_{path}, and the other is the increment of accumulative turning angle $\Delta\theta_{ac}$, represented as follows,

$$\Delta L_{path} = L_{path} - L'_{path} \tag{10}$$

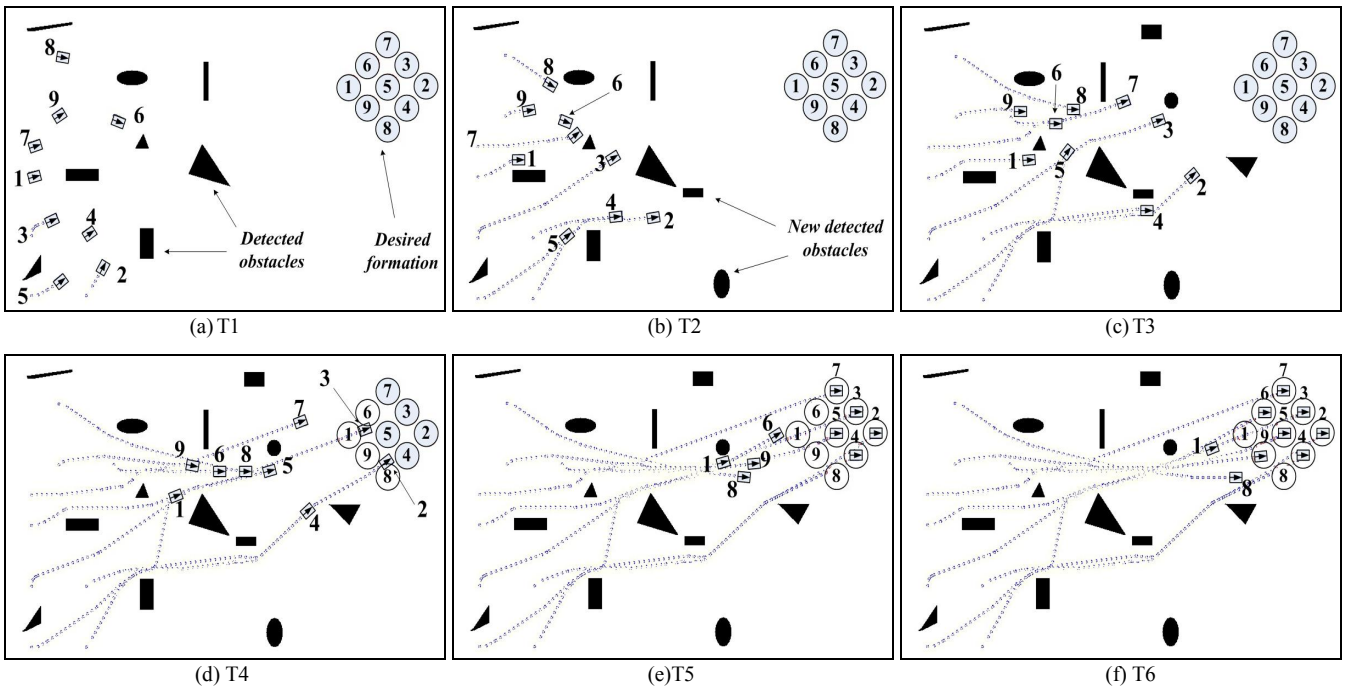
$$\Delta\theta_{ac} = \theta_{ac} - \theta'_{ac} \tag{11}$$

where L_{path} is the sum of length of every robot's path, θ_{ac} is the sum of turning angle of every robot, and L'_{path} and θ'_{ac} are ideal values representing the sum of length and turning angle without influence amongst the robots, which can only be calculated in the single robot planning. We should try to minimize $\Delta\theta_{ac}$ and ΔL_{path} in the motion planning.

V. Simulation

To demonstrate the validity of the proposed decentralized motion planning in formation forming, simulations were performed in Microsoft Visual C++ 6.0 platform under Windows XP. The size of the virtual environment is 1000×700 (pels), and the size of each robot is 21×17(pels). We assume that the sensor with the maximum range of 300 (pels) can construct a map of the environment [21], and the controller can track the path perfectly. LCPP was used to generate the trajectory path for each robot.

Fig. 7 illustrates the trajectories of the robots at six different times, with the priorities calculated by the proposed dynamic priority strategy as given in Table 1. The priorities of the robots were initialized by $p_i = K_i = 0.01\times i$, $(i = 1,...,9)$, at the beginning of time T1, and were then updated at time T2. Robot A_6 detected the collision with robot A_7 that had higher priority, and then stopped, as shown in Fig. 7(b). Robot A_4 coordinated its motion to let robot A_2 move first, as shown in Figs. 7(a) and (b). At time T3, robots A_5 updated its priority, because robots A_1, A_6, and A_9 entered its sensing range, and were found to hold formation-structure constraints to A_5, i.e., $F_5 = \{A_1, A_6, A_9\}$. Similarly, robot A_3 updated its priority, because it found that robot A_5 held constraint to A_3. The other robots kept the same priorities at time T3. At time T4, robot A_4 obtained its constraint set as $F_4 = \{A_8, A_9\}$, and updated its priority. Since robot A_4 held constraint to A_2, i.e., $F_2 = \{A_4\}$, F_2 was also changed to $F_2 = \{A_4, A_8, A_9\}$ based on (7), and the priority was updated. Similarly, robot A_5 updated its priority due to its new constraint set of $F_5 = \{A_1, A_9\}$, which further led to change of the constraint set of robot A_3 as $F_3 = \{A_1, A_5, A_9\}$. Based on the priorities, the four robots A_5, A_6, A_8, and A_9 cooperated with each other to move in a proper order in formation forming, as shown in Fig. 7 (d). At time T5, robots A_2, A_3, A_4, A_7 entered the formation for their higher priorities, and the other robots entered the formation at the later time of T6, as shown in Figs. 7 (e) and (f).



(a) T1 (b) T2 (c) T3

(d) T4 (e) T5 (f) T6

Fig. 7. Formation forming with dynamic priority strategy.

TABLE 1
DYNAMIC PRIORITIES OF NINE ROBOTS AT EACH TIMES

ROBOT/TIME	T1	T2	T3	T4	T5	T6
1	0.01	0.01	0.01	0.01	0.01	0.01
2	0.02	1.02	1.02	3.02	--	--
3	0.03	3.03	4.03	3.03	--	--
4	0.04	0.04	0.04	2.04	--	--
5	0.05	0.05	3.05	2.05	--	--
6	0.06	1.06	1.06	1.06	1.06	--
7	0.07	2.07	2.07	2.07	--	--
8	0.08	2.08	2.08	1.08	0.08	0.08
9	0.09	1.09	1.09	0.09	0.09	0.09

To better show the performance of the dynamic priority strategy in reducing the influence of formation-structure constraint, we randomly exchanged the goal positions of the robots in the desired formation and obtained 20 different target formations. We repeated the simulations for each target formation using both the dynamic and the static priority strategy to compare the values of ΔL_i and $\Delta\beta_i$. Fig. 8 illustrates the average values of ΔL_i and $\Delta\beta_i$ for each robot. When using the static priority strategy, the robots with higher priority usually arrive in their goals earlier than the robots with lower priority, and are less influenced by the formation-structure constraint. The robots with lower priority are often blocked by the robots with higher priority, and have to move longer path, which leads to higher values of ΔL_i and $\Delta\beta_i$. When using the proposed dynamic priority strategy, the robots enter the formation in a proper order, and the influence of the formation-structure constraint is greatly minimized, which leads to lower values of ΔL_i and $\Delta\beta_i$ for all the robots.

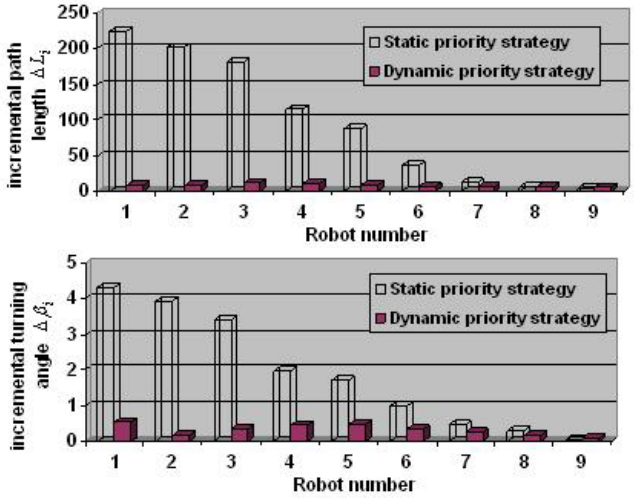


Fig. 8. Comparison between static priority and dynamic priority strategies.

VI. CONCLUSION

This paper presents a novel dynamic priority strategy used in a decentralized motion planning for formation forming of multiple mobile robots. A new LCPP path planner is proposed to plan path for an individual robot. A dynamic priority strategy is developed to give orders of the robots to enter the formation with collision avoidance. Simulations demonstrate that the proposed motion planning can enable multiple mobile robots to form formations in a proper order.

REFERENCES

[1] J. E. Hopcroft, J. T. Schwartz, and M. Sharir, "On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the 'warehouseman's problem'," Int. J. Robot. Res., vol. 3, no. 4, pp. 76-88, 1984.

[2] J. Barraquand, B. Langlois, J.-C. Latombe, "Numerical potential field techniques for robot path planning," IEEE Trans. Syst., Man, and Cyb. vol. 22, no. 2, pp. 224-241, 1992.

[3] P. Svestka and M. Overmars, "Coordinated path planning for multiple robots," Robot. Auton. Syst., vol. 23, pp. 125-152, 1998.

[4] S. M. LaValle and S. A. Hutchinson, "Optimal motion planning for multiple robots having independent goals," IEEE Trans. Robot. Autom. , vol. 14, no. 6, pp. 912-925, Dec. 1998.

[5] H. Wu, D. Sun and Z. Y. Zhou, "Model identification of a micro air vehicle in loitering flight based on attitude performance evaluation," IEEE Trans. on Robotics, vol. 20, no. 4, pp. 702-712, Aug. 2004.

[6] K. Kant and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," Int. J. Robot. Res., vol. 5, no. 3, pp.72–89, 1986.

[7] Y. Guo and L. Parker, "A distributed and optimal motion planning approach for multiple mobile robots," in Proc. IEEE Int. Conf. Robot. Autom., pp. 2612-2619, 2002.

[8] C. Ferrari, E. Pagello, J. Ota, and T. Arai, "Multirobot motion coordination in space and time," Robot. Auton. Syst., vol. 25, pp. 219-229, 1998.

[9] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," Algorithmica, vol. 2, pp. 477-521, 1987.

[10] S. Buckley, "Fast motion planning for multiple moving robots," in Proc. IEEE Int. Conf. Robot. Autom., pp. 322-326.1989

[11] M. Bennewitz, W. Burgard, and S. Thrun, "Finding and optimizing solvable priority schemes for decoupled path planning techniques for teams of mobile robots," Robot. Auton. Syst., vol. 41, no. 2, pp. 89-99, 2002.

[12] P. A. O' Donnell and T. Lozano-P´erez, "Deadlock-free and collision-free coordination of two robot manipulators," in Proc. IEEE Int. Conf. Robot. Autom., pp. 484-489, 1989.

[13] T. Sim´eon, S. Leroy, and J.-P. Laumond, "Path coordination for multiple mobile robots: A resolution complete algorithm," IEEE Trans. Robot. Autom., vol. 18, no. 1, pp. 42-49, Feb. 2002.

[14] J. Lumelsky and K. R. Harinarayan, "Decentralized motion planning for multiple mobile robots: The cocktail party model," Auton. Robots, vol. 4, no. 1, pp. 121-136, 1997.

[15] D. Sun, C. Wang, W. Shang, and G. Feng, "A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations," IEEE Trans. on Robotics, accepted, 2009.

[16] J. Chen, D. Sun, J. Yang, and H. Chen, "A leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme," Int. J. of Robotics Research, vol. 28, 2009.

[17] K.-H. Tan, M.A. Lewis, "Virtual structures for high-precision cooperative mobile robotics control," Autonomous Robots, vol. 4, pp. 387-403, 1997.

[18] T.D. Barfoot, C.M. Clark, "Motion planning for formations of mobile robots," Robotics and Autonomous System, vol.46, pp. 65-78, 2004.

[19] S. Liu, D. Sun, C. Zhu, and W. Shang, "A decentralized local constraint path planner for multiple mobile robots," IEEE Int. Conf. on Information and Automation, China, pp. 1222-1227, June 2009.

[20] J. Kuffner and S. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in Proc. IEEE Int. Conf. on Robot. Autom., pp. 995-1001, 2000.

[21] H. Chen, D. Sun, and J. Yang, "Global localization of multirobot formations using ceiling vision SLAM strategy," Mechatronics, vol. 19, vol. 5, pp. 618-628, August 2009.