Coordinated Task Manipulation by Nonholonomic Mobile Robots

Nusrettin Gulec Mustafa Unel Asif Sabanovic

Mechatronics Program, Faculty of Engineering and Natural Sciences,
Sabanci University, Orhanli-Tuzla, Istanbul, 34956, TURKEY
nusrettin@su.sabanciuniv.edu *{munel,asif}@sabanciuniv.edu*

Abstract— **Coordinated task manipulation by a group of autonomous mobile robots has received significant research effort in the last decade. Previous studies in the area revealed that one of the main problems in the area is to avoid the collisions of the robots with obstacles as well as with other members of the group. Another problem is to come up with a model for successful task manipulation. Significant research effort has accumulated on the definition of forces to generate reference trajectories for each autonomous mobile robots engaged in coordinated behavior. If the mobile robots are nonholonomic, this approach fails to guarantee successful manipulation of the task since the so-generated reference trajectories might not satisfy the nonholonomic constraint. In this work, we introduce a novel coordinated task manipulation model inclusive of an online collision avoidance algorithm. The reference trajectory for each autonomous nonholonomic mobile robot is generated online in terms of linear and angular velocity references for the robot; hence these references automatically satisfy the nonholonomic constraint. The generated reference velocities inevitably depend on the nature of the specified coordinated task. Several coordinated task examples, on the basis of a generic task, have been presented and the proposed model is verified through simulations.**

I. Introduction

Modeling groups of autonomous mobile robots engaged in coordinated behavior has been of increasing interest in the last years [1] - [10]. In 1995, Vicsek proposed the “neighbors” method, in which each member interacts only with its closest neighbors [3]. This approach has been used widely to define coordination among a group of mobile robots [4]. Yamaguchi introduced “formation vectors” to model coordinated motion for the achievement of a coordinated task by mobile robot troops [5].

Most of the research effort on the problem has been devoted to define virtual forces on each robot, which will yield coordinated motion of the group. Appropriately defined forces might be sufficient to achieve coordination among a group of autonomous holonomic (omnidirectional) mobile robots. In this case, the problem of achieving coordination requires appropriate definition of forces, under the effect of which the dynamics of the system will dictate the generated reference trajectories. If the group consists of nonholonomic mobile robots, these reference trajectories should satisfy the nonholonomic constraint so that the robots can follow. Generation of reference trajectories for the robots using a “*virtual reference robot*” that satisfies the nonholonomic constraint guarantees the generation of reference trajectories that the nonholonomic mobile robots can follow [11].

Achievement of a goal by multiple autonomous mobile robots in coordination is an example of decentralized systems. Studies on decentralized systems have been increasing in the last decade [12] - [14]. Failure of a single member in centralized systems results in system failure, whereas this is not the case in decentralized systems. A huge single robot, no matter how powerful it is, will be spatially limited while smaller robots could achieve a given goal more efficiently. Flocking birds, schooling fish and bees building a honeycomb in the beehive, as well as a group of people working as a search and rescue team, are examples of decentralized natural groupings.

Manipulation of a given coordinated task by a group of autonomous mobile robots requires the group to achieve certain formations. The necessary formation will inevitably depend on the definition of the coordinated task [15]. It could be better to carry a heavy rectangular object by multiple robots in a rectangular formation, while a circular formation might be better for capturing and enclosing an invader to provide security in surveillance areas.

The organization of this paper is as follows. In section II, we describe a generic coordinated task scenario and present some complicated coordinated task examples. In section III, we present a novel method that defines reference linear and angular velocities, which in turn imply online collision-free reference trajectories for the autonomous nonholonomic mobile robots. In section IV, the simulation results for some coordinated task examples are presented to verify the framework we introduce. In section V, we conclude with some remarks and suggest some future work.

II. Coordinated Task Scenarios

We consider a group of n autonomous nonholonomic mobile robots, namely $R_1, R_2, \ldots, R_{n-1}, R_n$, and an object, T, that will serve as a target for the group.

The generic coordinated task scenario and the required formations we consider in this work can be summarized as follows:

- Starting from any initial setting of the robots and the target, $R_1, R_2, \ldots, R_{n-1}, R_n$ should form a circle of radius d_T, with T being at the center.

0-7803-9511-5/06/$20.00 ©2006 IEEE

- The robots should be uniformly distributed on the circle, with each robot maintaining a certain distance d_{P2}, that certainly depends on d_T and the number of robots, from its closest neighbors.
- Each R_i should orient itself towards T once it satisfies the conditions given in the previous items.

Complicated coordinated tasks can be designed in a sequential manner in which each phase will be easier than the overall task to be accomplished. We call the above scenario *generic* because it might serve as the first phase in a variety of more complicated coordinated tasks. Depending on the definition of the overall coordinated task, the next phase in the sequence might need to be initiated after all robots are done with the current phase; i.e. R_i might need to make sure all other robots have completed the current task before it goes on with the next phase.

A more complicated coordinated task example could be the manipulation of an object by the mobile robot group. Heavy-load lifting, box-pushing and the alignment of disordered objects are some of the examples encountered in the literature [16]. In this case, the object to be manipulated will be embedded in the above scenario as the target, T. Once the above described generic formation is achieved, the robots can grasp and move the object in a coordinated manner.

Another example for a complicated coordinated task could be the capturing of an invader by a group of mobile robots by enclosing him. In this case, the target in the generic scenario, T, will be the invader. Once that circular formation is achieved, the next phase would be decreasing d_T, and accordingly d_{P2}.

The coordinated task examples can be further increased. No matter what the coordinated task is, autonomous robots must avoid collisions with other members of the group and any other static or dynamic obstacles during each phase. Collision turns out to be one of the most essential problems in the context of coordinated motion [17].

In this work, we assume a stationary target, T, the position of which is a priori known by all autonomous mobile robots. We are also assuming that each robot, R_i, can sense at least two of the other robots at each instant.

III. Definition of Kinematics

In this work, we are assuming that the group consists of autonomous mobile robots that are nonholonomic. Hence, the generated reference trajectories should satisfy the nonholonomic constraint so that the robots can track. For this purpose, we are defining coordination among the group by the appropriate design of linear and angular velocity references of each robot.

In the lower level, the regulation of the errors between the reference pose, obtained by the integration of the reference velocities, and the actual robot's pose to zero are guaranteed with the application of the smooth time-varying feedback control law given in [18].

A. Reference Velocities

The linear and angular velocity references of each robot are defined to achieve the coordinated motion of the group. The velocities will depend on the position of target so that the group moves towards T. On the other hand, the velocities will be dependent on the positions of the other members in the group so that the robots move in a coordinated manner; i.e. they will maintain certain formations.

1) Target Velocity: Each robot in the group has to move towards T from any initial position so that the generic task described above can be accomplished.

For each robot, a reference velocity vector due to T, $\overrightarrow{v}_T$, is defined as follows to move the robot towards the target:

$$\overrightarrow{v}_T = k_v(d_{i2T} - d_T)\overrightarrow{n}_{i2T} \ , \tag{1}$$

where $k_v > 0$ is a positive proportionality constant, d_{i2T} is the distance between R_i and T, and $\overrightarrow{n}_{i2T}$ is the unit vector in the direction from R_i to T.

2) Coordination Velocity: Coordinated motion of a group of autonomous mobile robots is defined in terms of maintaining certain mutual distances between the robots. We consider a biologically inspired coordination method where R_i is in interaction only with its closest two neighbors. Indeed, the interaction with the second closest neighbor is killed when the robot is close to the target, so that the accomplishment of the task is favored compared to coordination among the robots.

The reference velocity vector of each robot due to its closest neighbors, $\overrightarrow{v}_{coord}$, is given as follows to enable coordinated motion:

$$\overrightarrow{v}_{coord} = \overrightarrow{v}_{n1} + \overrightarrow{v}_{n2} \ ,$$

$$\overrightarrow{v}_{n1} = k_v(d_{i2n1} - d_{coord})\overrightarrow{n}_{i2n1} \ ,$$

$$\overrightarrow{v}_{n2} = \begin{cases} k_v(d_{i2n2} - d_{coord})\overrightarrow{n}_{i2n2} & \text{if} \quad d_{i2T} \geq d_{task} \\ 0 & \text{if} \quad d_{i2T} < d_{task} \end{cases} \ , \tag{2}$$

where $\overrightarrow{v}_{n1}$ and $\overrightarrow{v}_{n2}$ are the velocities due to the closest and second closest neighbors respectively, d_{i2n1} and d_{i2n2} are the distances between R_i and its closest and second closest neighbors, $\overrightarrow{n}_{i2n1}$ and $\overrightarrow{n}_{i2n2}$ are the unit vectors from R_i to its closest neighbors, and d_{task} is a critical distance of R_i to T below which the accomplishment of the task is favored compared to coordination.

3) Reference Velocity Combination: Each robot has two reference velocities as described above; one for moving towards T, and one for coordination with the other members of the group.

The reference velocity vector for each robot, namely $\overrightarrow{v}_{ref}$ is defined as a linear combination of the above velocities by appropriate coefficients:

$$\overrightarrow{v}_{ref} = k_T\overrightarrow{v}_T + k_{coord}\overrightarrow{v}_{coord} \ , \tag{3}$$

where $k_T > 0$ and $k_{coord} > 0$.

Successful achievement of certain formations might require changes in the parameters of the above equations. In our scenario, we split the coordinated task into two main phases:

$P1$ Approaching T starting from an initial setting.

$P2$ Achieving a circular formation of radius d_T in which all robots head towards the center that is defined by the position of T.

B. Parameter Switching

In phase $P1$, i.e. when R_i is approaching T from its initial position, the priority is given to coordination. In other words, $\vec{v}_{coord}$ is dominant in this case so the robots move as a group. However, $\vec{v}_T$ still contributes to the reference velocity so the group approaches T. To achieve the dominance of $\vec{v}_{coord}$, we choose $k_{coord} > k_T$ in (3). In this phase, d_{coord} in (2) is set to the initially defined value d_{P1}, while k_{coord} is set to the predefined value k_{P1}. R_i remains in this phase as long as $d_{i2T} \geq d_{task}$; i.e. R_i is far from T.

If d_{i2T} is lower than d_{task}, R_i enters phase $P2$. In this phase, the priority is given to maintaining the distance d_T from T. Hence, $\vec{v}_T$ is dominant over $\vec{v}_{coord}$. To achieve this, we set $k_T > k_{coord}$ in (3). In this phase, k_{coord} is set to the predefined value $k_{P2} \leq k_{P1}$. To achieve a uniform distribution on the formation circle, d_{coord} in (2) should also be changed to a new value, d_{P2}, possibly different from the initial coordination distance, d_{P1}. It follows from *Law of Cosines* that for a uniform distribution of n robots on a circle with radius d_T, d_{P2} should be given as follows:

$$d_{P2} = d_T\sqrt{2(1 - cos(2\pi/n))} \; . \tag{4}$$

When R_i maintains distance d_T from T and d_{P2} from its closest neighbor, the last maneuver it should take is to orient itself towards T, hence complete phase $P2$.

The last parameter that affects the generation of reference velocities is k_T in (3) which defines the dependency of $\vec{v}_{ref}$ on $\vec{v}_T$. Since k_{coord} is switched from k_{P1} to k_{P2} when R_i enters phase $P2$, a constant k_T might be used as long as is satisfies the condition $k_{P1} > k_T > k_{P2}$. However, for the dominance in both phases to be more significant, we also switch k_T from a lower value in $P1$ to a higher value in $P2$.

k_{coord} is switched as a continuous function of d_{i2T}. It is decreased when R_i completes phase $P1$ so that the priority is given to the achievement of the formation around T. This switching is modeled by the following sigmoid function:

$$k_{coord} = k_{P2} + \frac{k_{P1} - k_{P2}}{1 + exp(\mu(d_{task} - d_{i2T} + \phi))} \; , \tag{5}$$

where $\mu > 0$ and $\phi > 0$ are constants.

Similarly, d_{coord} is changed as a continuous function of d_{i2T}, which is modeled by another sigmoid function given by:

$$d_{coord} = d_{P2} + \frac{d_{P1} - d_{P2}}{1 + exp(\mu(d_{task} - d_{i2T} + \phi))} \; . \tag{6}$$

To facilitate the computations, we define k_T as a function of k_{coord} as follows:

$$k_T = 1 - k_{coord} \; . \tag{7}$$

In doing so, we make only one sigmoid function computation for two different parameters, k_{coord} and k_T. The choice of k_{P1} and k_{P2} in (5) should satisfy: $0 \leq k_{coord} \leq 1$ to be able to use (7). Fig. 1 depicts so obtained switching of k_T.

C. Collision Avoidance

Collision avoidance is the last factor contributing to the definition of the reference velocities for the robots.

For each R_i, we consider a *virtual collision prediction arc (VCPA)*, Ω_i, as depicted in Fig. 2(a). R_i detects a collision risk when any other member of the group touches Ω_i. The proposed algorithm uses sensory information of the robots to predict collisions and updates the reference velocities to avoid them *online*. In case of such a collision risk, the velocity reference of R_i is updated based on the algorithm described below. After such an update, the reference is kept constant for a predefined, short period of time to ensure successful avoidance of a possible collision. After this delay, the reference velocity is calculated by (3) and the collision avoidance algorithm explained below is run to predict further possible collisions.

The updated velocity reference of R_i to avoid the collision is designed based on the relative velocity of the sensed robot, R_j, with respect to R_i. The coordinate frame attached to R_i and two parts, R_Ω and L_Ω, as shown in Fig. 2(b) are defined for the collision avoidance algorithm.

If a collision risk is detected, the following algorithm is run:

- Calculate the component of the velocity of R_j projected on the axis y_i.
- Rotate counter-clockwise if that component is positive.
- Rotate clockwise if that component is negative.
- If that component is zero:
  - Rotate clockwise if R_j touches L_Ω.
  - Rotate counter-clockwise if R_j touches R_Ω or x_i.

The explained algorithm is illustrated by the illustrated examples of Fig. 3. In both cases, only R_A detects a collision risk with R_B so the reference velocity for R_A is updated while that of R_B is constant. In Fig. 3(a), the reference velocity vector of R_A is rotated in counter-clockwise direction since this is the shortest path to avoid the collision. In Fig. 3(b), on the other hand, the path in the counter-clockwise direction seems to be shorter. However, this would result in consecutive collision predictions. To avoid this occurrence, the reference velocity vector of R_A is rotated in clockwise direction.

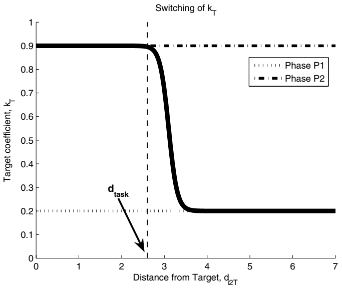

Fig. 1. Continuous switching of target coefficient vs distance from T

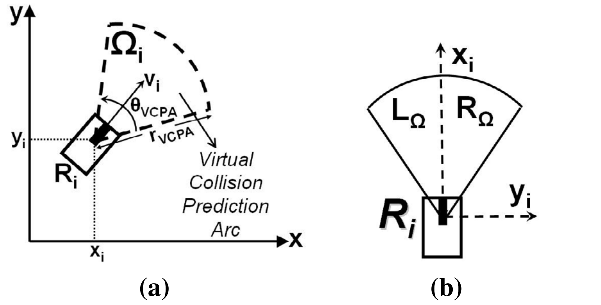

Fig. 2. (a)$VCPA$, Ω_i, for R_i (b)R_i's coordinate frame and parts of Ω_i

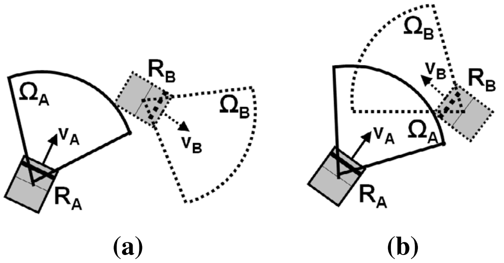

Fig. 3. Examples for illustration of collision avoidance algorithm

D. Trajectory Generation

The reference linear and angular velocities, u_L and u_R, of each robot, R_i, are derived from the reference velocity vector, $\overrightarrow{v}_{ref} = [\ v_{xref} \quad v_{yref}\ ]^t$, given by (3).

The reference angular velocity, u_R, is calculated by a negative feedback of proportionality constant k_r as follows:

$$u_R = k_r e_\theta \qquad \text{with} \qquad e_\theta = (\theta_{ref} - \theta_i) \quad , \tag{8}$$

where e_θ is the orientation error, $\theta_{ref} = \tan^{-1}(v_{yref}/v_{xref})$ is the orientation reference and θ_i is the actual orientation.

The reference linear velocity, u_L, is obtained from the reference velocity vector and error in orientation as:

$$u_L = \begin{cases} \dfrac{1}{|e_\theta|}\sqrt{v_{xref}^2 + v_{yref}^2} & \text{if} \quad |e_\theta| > \theta_{sat} \\ \\ \dfrac{1}{\theta_{sat}}\sqrt{v_{xref}^2 + v_{yref}^2} & \text{if} \quad |e_\theta| \leq \theta_{sat} \end{cases} \quad , \tag{9}$$

where $\theta_{sat} > 0$ is a limiting value.

The reference pose of R_i, $[\ x_{ref} \quad y_{ref} \quad \theta_{ref}\ ]^t$, is obtained by the integration of its linear and angular velocity references, as follows:

$$\begin{bmatrix} x_{ref} \\ y_{ref} \\ \theta_{ref} \end{bmatrix} = \begin{bmatrix} \int u_L \cos(\theta_{ref})dt \\ \int u_L \sin(\theta_{ref})dt \\ \int u_R dt \end{bmatrix} . \tag{10}$$

Finally, the generated reference pose is input to the low-level controller that applies the smooth time-varying feedback control law given in [18].

IV. Simulation Results

Simulations and animations were carried out for the verification of the proposed algorithm. In the simulations, maximum linear speed of the robots was set to $0.5m/sec$ and maximum rotational speed was set to $(\pi/6)rad/sec$. The parameters for Ω_i were set as $r_{VCPA} = 0.9m$ and $\theta_{VCPA} = (\pi/2)rad$.

A. Collision Avoidance Simulations

The simulations were run with $\overrightarrow{v}_{coord} = \overrightarrow{v}_T = 0$, and maximum initial speed for the robots, to test the performance of the collision avoidance algorithm.

In the first scenario, two robots initially move towards each other as seen in Fig. 4(a). As each robot touches the *VCPA* of the other, they both predict a collision. They both change their orientations and the final situation is depicted in Fig. 4(c). Hence, the algorithm proves to be successful in avoiding head-to-head collisions.

In the second scenario, depicted in Fig. 5, only one of the robots detects a collision risk. Robot-A doesn't change its orientation and the algorithm is verified in the case of a single robot's collision prediction. In the third scenario, three robots are headed towards each other as given in Fig. 6(a). In Fig. 6(b), each robot detects a collision risk with another robot. Although the situation is more complicated, they manage to avoid collisions and the result is given in Fig. 6(c).

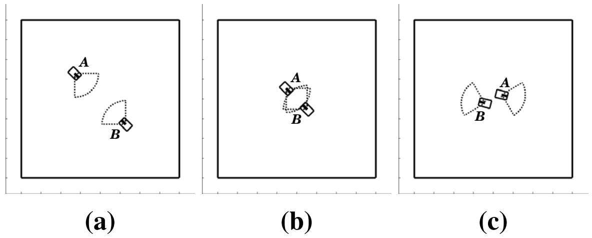

Fig. 4. Head-to-Head (a)Before (b)Prediction (c)After

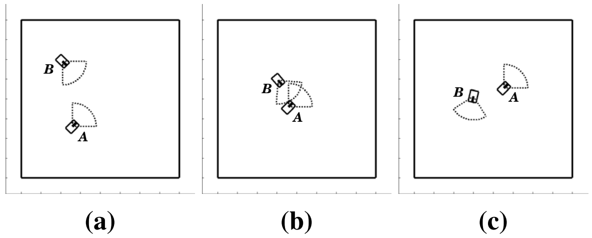

Fig. 5. Single robot (a)Before (b)Prediction (c)After

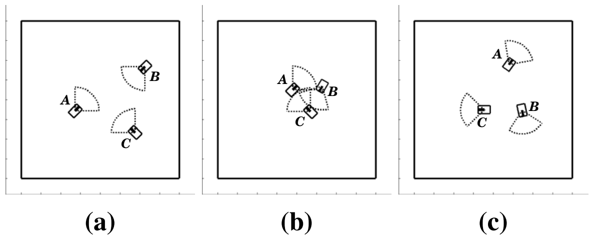

Fig. 6. Three robots (a)Before (b)Prediction (c)After

B. Coordinated Task Manipulation Simulations

The developed coordinated task manipulation scheme was simulated for groups of three, four and five robots. Simulations and animations were carried out both for the explained generic coordinated task and the coordinated invader-enclosing task that is mentioned in Section II. The following values were used in the simulations: $k_v = 0.5$, $d_T = 2.0m$ and $d_{task} = 2.6m$ in (1) and (2); $k_{P1} = 0.8$, $k_{P2} = 0.1$, in (5); $d_{P1} = 2.0m$ in (6), $\mu = 10$ and $\phi = 0.5$ in (5) and (6).

1) Circular Formation: Simulations of the developed model for the generic task described in Section II are carried out.

The initial setting in Scenario-1 is depicted in Fig. 7(a). Coordination effects are dominant in phase $P1$ as seen in Fig. 7(b). Once the robots reach the circle, they spread around to achieve neighboring distances of d_{P2}, but stay on the circle due to the dominance of $\overrightarrow{v}_T$; hence perform circular motion. The resulting formation is shown in Fig. 7(d).

Four robots are considered in Scenario-2. The only difference is that the group forms a parallelogram instead of a triangle. Fig. 8 shows some phases of this animation.

The initial setting in Scenario-3 is depicted in Fig. 9(a). As there are more robots, the risk of collision increases for the same value of d_T. As depicted in Fig. 9(b), some collisions were predicted around the formation circle, but they were successfully avoided and the robots started circular motion on the formation circle as shown in Fig. 9(c). The final formation is satisfactory with a uniform distribution of robots as shown in Fig. 9(d).

2) Enclosing an Invader: The results from the previous section are extended to verify the conformity of the generic task as the first phase for a complicated task.

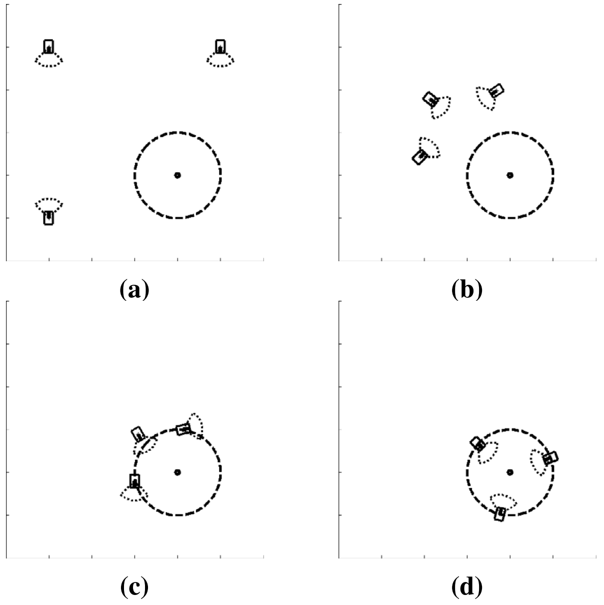

Fig. 7. Scenario-1: (a)Initial configuration (b)Coordination dominant (c)Circular motion (d)Desired formation achieved

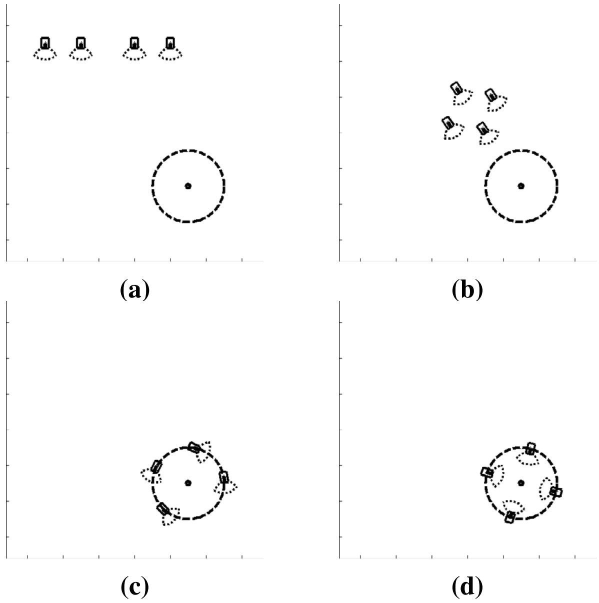

Fig. 8. Scenario-2: (a)Initial configuration (b)Coordinated motion (c)Circular motion (d)Desired formation achieved

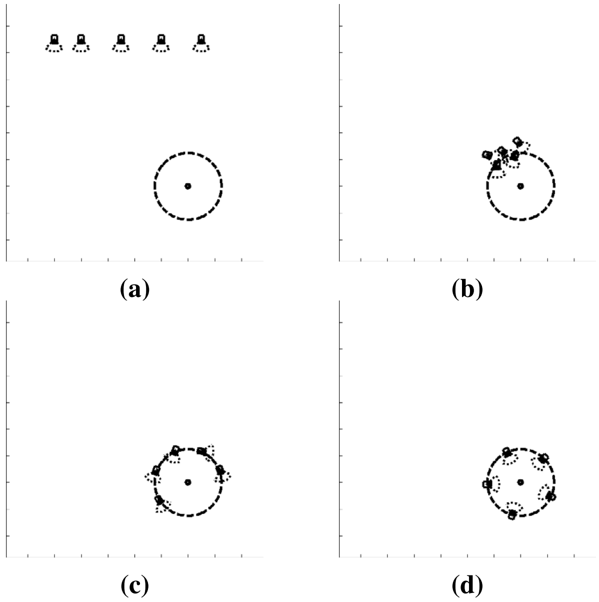

Fig. 9. Scenario-3: (a)Initial configuration (b)Higher collision possibilities (c)Circular motion (d)Desired formation achieved

The initial setting in Scenario-1 is depicted in Fig. 10(a). Once the robots achieve the circular formation shown in Fig. 10(c), they move towards the invader and enclose it as seen in Fig. 10(d).

The initial setting of Scenario-2 is shown in Fig. 11(a). As depicted in Fig. 11(b), the robots perform circular motion and achieve the circular formation of the generic task. Fig. 11(c) is a snapshot of the robots approaching the invader while keeping the circular structure. The invader is successfully enclosed in the final formation as shown in Fig. 11(d).

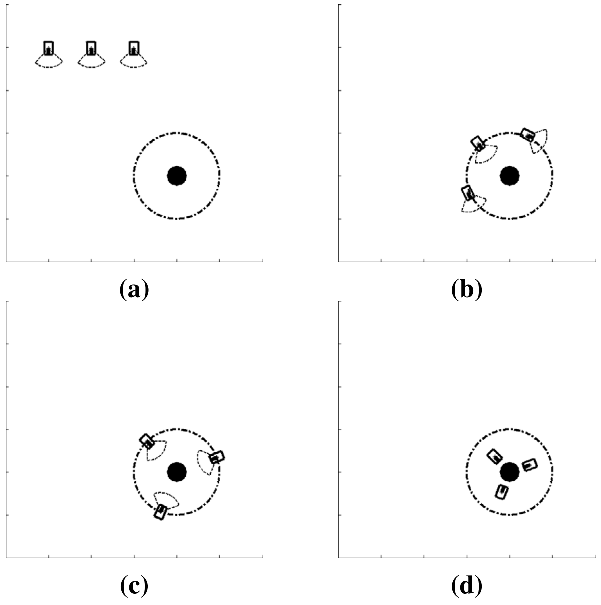

Fig. 10. Scenario-1: (a)Initial configuration (b)Circular motion (c)Circular formation (d)Invader enclosed

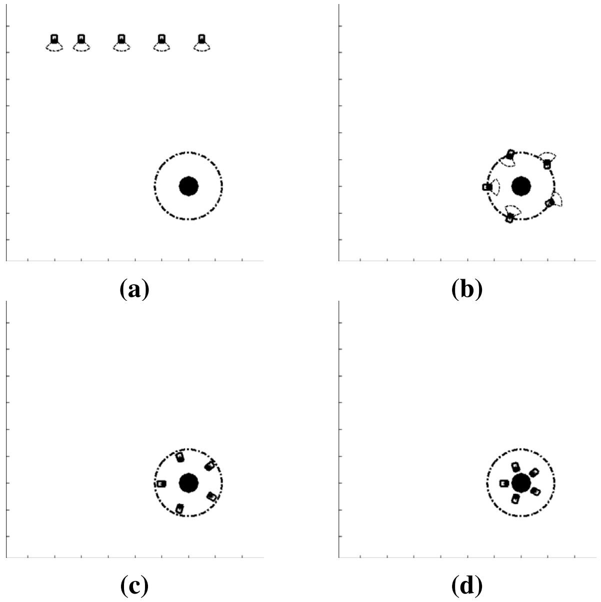

Fig. 11. Scenario-2: (a)Initial configuration (b)Circular motion (c)Enclosing the invader (d)Invader enclosed

V. Conclusions

In this paper, we have developed a decentralized coordination algorithm for a group of wheeled mobile robots by appropriately defining reference linear and angular velocities for the robots. Possible collisions are predicted by $VCPA$s and the reference velocities are updated when a collision risk is detected.

Simulation results are promising. They reveal that the algorithm suffices to model and control coordinated motion as well as coordinated task manipulation for a group of autonomous nonholonomic mobile robots. The success with different number of robots verifies the developed model and emphasizes the modular structure.

We are working on the addition of more complicated coordinated tasks in the sequence and the realization of the system using visual sensing.

References

[1] S.G. Loizou, H.G. Tanner, V. Kumar, K.J. Kyriakopoulos, “Closed Loop Motion Planning and Control for Mobile Robots in Uncertain Environments”, *Proceedings of the 42nd IEEE Conference on Decision and Control*, Vol. 3, December 2003, 2926-2931.

[2] T. Suzuki, T. Sekine, T. Fujii, H. Asama, I. Endo, “Cooperative Formation among Multiple Mobile Robot Teleoperation in Inspection Task”, *Proceedings of the 39th IEEE International Conference on Decision and Control*, Vol. 1, December 2000, 358-363.

[3] T. Vicsek, A. Czirk, E. Ben-Jacob, I. Cohen, O. Shochet, “Novel Type of Phase Transition in a System of Self-Driven Particles” *Phys. Rev. Lett.*, Vol. 75, No. 6, August 1995, 1226-1229.

[4] A. Jadbabaie, J. Lin, A.S. Morse, “Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules”, *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 3, December 2002, 2953-2958.

[5] H. Yamaguchi, “A Cooperative Hunting Behavior by Mobile-Robot Troops”, *The International Journal of Robotics Research*, Vol. 18, No. 8, September 1999, 931-940.

[6] J.B. de Sousa, F.L. Pereira, “Specification and design of coordinated motions for autonomous vehicles”, *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 1, December 2002, 101-106.

[7] P. Seiler, A. Pant, K. Hedrick, “Analysis of Bird Formations”, *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 1, December 2002, 118-123.

[8] A. Borkowski, M. Gnatowski, J. Malec, “Mobile Robot Cooperation in Simple Environments”, *Proceedings of the Second International Workshop on Robot Motion and Control*, October 2001, 109-114.

[9] T.B. Gold, J.K. Archibald, R.L. Frost, “A Utility Approach to Multi-Agent Coordination”, *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA ’00*, Vol. 3, April 2000, 2052-2057.

[10] T. Weigel, J.S. Gutmann, M. Dietl, A. Kleiner, B. Nebel, “CS Freiburg: Coordinating Robots for Successful Soccer Playing”, *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, October 2002, 685-699.

[11] 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*, Vol. 2, April 1991, 1136-1141.

[12] S. Souissi, X. Defago, T. Katayama, “Decomposition of Fundamental Problems for Cooperative Autonomous Mobile Systems”, *Proceedings of the 24th International Conference on Distributed Computing Systems*, March 2004, 554-560.

[13] J.D. Sweeney, H. Li, R.A. Grupen, K. Ramamritham, “Scalability and Schedulability in Large, Coordinated, Distributed Robot Systems”, *Proceedings of the 2003 International Conference on Robotics and Automation, ICRA ’03*, Vol. 3, September 2003, 4074-4079.

[14] J.S. Baras, X. Tan, P. Hovareshti, “Decentralized Control of Autonomous Vehicles”, *Proceedings of the 42nd IEEE Conference on Decision and Control*, Vol. 2, December 2003, 1532-1537.

[15] Y. Hur, R. Fierro, I. Lee, “Modeling Distributed Autonomous Robots using CHARON: Formation Control Case Study”, *Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, ISORC ’03*, May 2003, 93-96.

[16] J. Spletzer, A.K. Das, R. Fierro, C.J. Taylor, V. Kumar, J.P. Ostrowski, “Cooperative Localization and Control for Multi-Robot Manipulation”, *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2, October 2001, 631-636.

[17] T. Rabie, A. Shalaby, B. Abdulhai, A. El-Rabbany, “Mobile Vision-based Vehicle Tracking and Traffic Control”, *Proceedings of the Fifth IEEE International Conference on Intelligent Transportation Systems*, September 2002, Singapore, 13-18.

[18] C. Samson, “Trajectory tracking for non-holonomic vehicles: overview and case study”, *Proceedings of the Fourth International Workshop on Robot Motion and Control, RoMoCo’04*, June 2004, 139-153.