Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Motion Planning for Multi-robot Coordination on Representation Space

Jianbo Su * **Yanjun Zhang**

* *Department of Automation, Shanghai Jiao Tong University, Shanghai, 200240, China ; e-mail: jbsu@sjtu.edu.cn).*

Abstract: The multi-robot coordination task is investigated in its Representation Space (RS). First, the RS model of the robot system as well as its prescribed task is formulated by clarifying the internal and external constraints affecting task realization. All constraints are depicted as unreachable areas in RS model. Then, whether a task is feasible or not is transformed to check 1) if the final representation denoting task realization of the robot system is reachable; and 2) if there is a connecting trajectory illustrating the process that the start representation varies to the final representation, amidst those unreachable areas denoting system constraints. Performance of different planning algorithms are also evaluated upon the representation space, from which the optimal planning algorithm could thus be recognized. The task of multi-robot formation navigation in a warehouse environment is exemplified to illustrate the performance of the proposed scheme: the formation of the robot system is maintained in motion with collisions avoided. Moreover, by incorporating the task oriented motion planning framework, it is capable of transforming an infeasible task into a feasible one with least adjustments in the system's formation to adapt to inner and/or external constraints during task realization.

1. INTRODUCTION

In many multi-robot coordination tasks, a group of robots are required to march in a predefined spatial pattern or formation. For example, in automatic warehouse applications, several robots are sometimes required to perform storage and retrieval task in a specific formation. In search and rescue tasks, a group of robots navigate inside a building in formation to ensure a complete and efficient search for survivals. And for transportation tasks, robots sometimes need to maintain certain formation in order to transport objects.

Many research work has already been dedicated to the problem of multi-robot navigation in cluttered environment. Multi-robot system is difficult to be analyzed by constructing configuration space model due to its extremely high dimension. Fully decentralized methods like flocking or schooling strategies enable control and coordinations of large groups of robots with relatively little computation [1]. Each robot agent adjusts its velocity according to its neighbors, subject to whole formation requirements. However, in the presence of obstacles, formation constraints or collision avoidance can not easily be guaranteed. The relatively less centralized approaches are behavior-based methods [4]. These approaches plan motions by specifying the relations between the robots. The robots move while maintain these relationships. Much of these work do not guarantee that formation constraints are maintained in the presence of obstacles. In [6], an abstraction method was proposed to establish a boundary for the robots, with which collision avoidance and formation constraints can be guaranteed. However, in this method, state information of each robot must be accessible to all other robots to enable centralized control. Therefore, the computational burden is extremely heavy when this scheme is applied to robot groups of large numbers.

It is noted that the complexity of formation motion planning problem can greatly be reduced by combining motion planning with the prescribed task. The idea of integrating task and motion planning has been investigated for planar robot navigation among movable obstacles . Obstacles can be moved if they block the robot's way from initial position to goal position. The proposed approach decouples computations of the robot motions and obstacle movements and maintains an explicit state of the task's state space. Path searching is conducted in this state space and probabilistic completeness is proven. However, only planar applications are considered and the proposed approach is not extendable for multi-robot system.

If the planner fails, it normally means that the task is infeasible. Then how can an infeasible task be converted to a feasible one? This problem is referred to as "Constraint Relaxation" in [2]. The idea is to generate an initial path considering only some of the constraints with high priority and iteratively refine the path into a feasible one by incorporating additional constraints [5]. A similar strategy for multi-limbed robots moving on uneven terrain is proposed in [3]. In [8], this problem is formulated as Minimum Constraint Removal (MCR) motion planning problem, in which the objective is to remove the fewest constraints necessary to allow the start point and the goal point connectable. A sampling-based motion planner is presented that incrementally grows a roadmap according to how many constraints are violated. The algorithm is proved to be asymptotically optimal in that as more time spent fewer numbers of obstacles need to be removed to produce a feasible path. However, all these methods mentioned above can only find out which constraint makes the task infeasible but can not provide hints on how to modify the constraints to convert the task to be feasible.

The planning and coordination approaches of multi-robot system proposed above have widely been applied in practice. However they are all constrained by specific tasks and robot system attributes. If either the task requirement or the robot system configuration is changed, collision avoidance or formation constraints can not be guaranteed. The Representation Space (RS) strategy towards robot task planning was first proposed by Su *et al.* in [7]. An RS is constructed to denote all factors that affect the task accomplishments, including inner factors from robot system, such as its motion ability and/or manipulation ability, and outer factors from surrounding environments, such as obstacle avoidance and/or mutual collision avoidance. At the start and the end stage of the task realization, the corresponding representations of the robot system could be denoted in its affiliated RS. The process of the robot system to accomplish a specific task is the process that the robot's representation varies subject to inner and outer constraints in its RS. If the task is feasible, there must be at least one trajectory in the RS connecting its start representation to the end representation. And the optimal strategy to accomplish the task could be figured out among those trajectories subject to a prescribed optimal index.

On the other hand, if there is no trajectory connecting start and end representation of the robot system in its RS, the task should then be infeasible. In this case, the factors that prohibit the task to be accomplished could be recognized in its RS. And it is natural to see what factors make the task infeasible and how to change those factors to convert an infeasible task to a feasible one. In this paper, we investigate the robotic formation motion task in a task oriented motion planning framework based on its affiliated RS. By mapping task related constraints into representation space, it is possible to adjust the task requirement and/or robot system and turn the original infeasible task into a feasible one.

This paper is organized as follows. Section II introduces the multi-robot formation task from the viewpoint of representation space model of the robot system. Section III investigates task realizability as well as its optimal realizing strategy. Section IV shows how to transform an infeasible task to a feasible one with the help of RS, followed by Conclusions in Section V.

2. MULTI-ROBOT FORMATION TASK

For a mobile vehicle capable of translating and rotating in a 2D environment, its configuration space is constructed as a 3 dimensional manifold $SE(2) = R^2 \times S^1$. For a flying robot able to translate and rotate in 3D space, its configuration space can be $SE(3) = R^3 \times SO(3)$. The configuration space model characterizes the location and orientation of the robot. Similarly we can model a multi-robot system using configuration space by combining the configuration space model of each robot using Cartesian product. This model, therefore, is of extremely high dimension. Planning in a space of such high dimension causes computational problems. To solve the problem, we propose the representation space of multi-robot formation task and design the hierarchical structure.

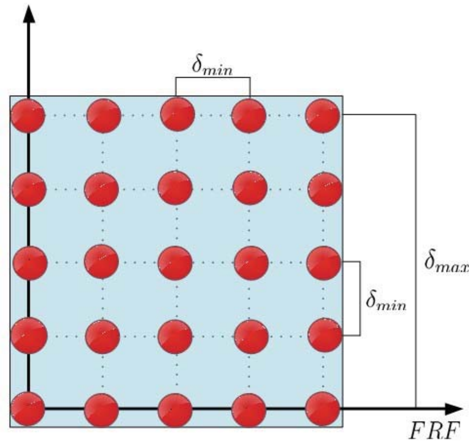


Fig. 1. Maximum number of robots in a rectangular formation

2.1 Representation Space Model

Based on the representation space framework, we can model the multi-robot formation by modifying the idea of generalized coordinates [4]. The task related attributes may include location, orientation, pattern and size of a formation, which can be described in terms of representation variable.

Formation Reference Frame(FRF) The FRF is a body frame whose origin and orientation are uniquely determined for any configuration of the formation elements.

The position vector from the origin of the inertial frame I to the origin of FRF is denoted by $p_o = (p_o)_I$, where $(\cdot)_I$ indicates a column vector of components relative to frame I.

The location of a formation is defined as location of the origin of formation reference frame(FRF). The orientation of formation is defined as the orientation of the FRF.

During any rigid displacement of the formation characterized by a rotation R, FRF is also rotated by R and origin of FRF remains rigidly fixed to the formation. We define R as rotation matrix relating FRF frames and I, so that for any vector r, $(r)_I = R \cdot (r)_{FRF}$.

To define the formation pattern, we first define the notion of formation feature points.

Formation Feature Point Formation feature points are the elements in a formation from which the location and orientation of all other elements can be derived.

The formation pattern σ is defined as $\sigma = f(x_1, x_2, \ldots, x_p)$, in which x_1, x_2, ..., x_p are feature points of certain formation. The size s represents the boundary and size of the abstraction which encloses the group of robots. The choice of size s is very flexible, as long as the change of formation shape can be modeled explicitly. Formation feature points maintain the boundary of formation to keep robots from escaping the formation.

Through the modeling of multi-robot formation, it is possible to characterize the p_o, R, s, σ by the representation variables ζ_{p_o}, ζ_R, ζ_s, ζ_σ respectively. If task is navigation, then only position and orientation information are necessary in most cases. Therefore, the representation model can be constructed by $\zeta = (\zeta_p, \zeta_R)$. If task is to change the formation shape, then the representation model can be $\zeta = (\zeta_\sigma, \zeta_s)$.

For a infeasible task, if either the initial representation or the goal representation are within the unreachable scope of representation space, task or robot system must be modified to make sure that both the initial representation and goal representation are within the reachable region of representation space. When the task is infeasible due to there is no feasible solution of the motion planning problem, the representation model must be rebuilt in order to remove the constraints that make the task infeasible. Therefore, it is possible to evaluate the task realization under the representation space framework.

2.2 Hierarchical structure

This section presents the specification of hierarchical structure of the approach. At the bottom level, individual robots execute the continuous controllers that are designed to satisfy internal constraints and maintain desired formation shape. At the middle level, the robots that act as the formation feature points communicate with each other to maintain certain formation constraints. At the top level, the formation navigates the space while avoid collision with the obstacles. In general, the team of robots except the formation feature points can be heterogeneous; thus they might not share the same configuration space.

To eliminate the possibility of local minima which may occur when incorporate two controllers, the hierarchical control system enables multiple time scale approach[6]. Motion of individual robots are assumed to evolve on a much faster time-scale than the motion of formation. Sufficient time scale separation between formation and individual robots ensures the two controllers can be designed independently.

Input to each agent in the local reference frame of the formation is

$$u_i^F = u_i^F(x_1^f, x_2^f, \ldots, x_p^f, s), \tag{1}$$

in which, x_i^f is the position of formation feature points i and s is the shape.

Moreover, input to each agent in the global reference frame can be derived as

$$u_i = R(\theta)u_i^F + u_F^{p_o}, \tag{2}$$

where $u_F^{p_o}$ is the translational component of the abstraction input, $R(\theta)$ is the rotation matrix at θ, and u_i^F is the individual input of robot i in the local coordinate frame defined by FRF.

Since the boundary restricts the allowable space for the robot motion, the robots' configuration space is decoupled from the cluttered physical workspace. To solve the local navigation problem, we can use navigation function approach that guaranteeing that the robots stay within the formation boundary. If the robot formation was being used for surveillance of different spaces, a Voronoi coverage type controller [9] can be used. If it is required to maintain a specific shape, [2] can be used. For stricter formations, [10] provide a more structured organization of robots.

2.3 Task Related Constraints

In multi-robot formation task, the external constraints include obstacle constraints of the environment and formation constraints. In order to avoid collisions, robots must maintain a minimum distance from each other. A maximum distance must be maintained if two robots need to keep communication link.

Setting shape constraints [10] is critical to ensure there is enough space in the formation to maintain the desired shape. Knowing the shape of the formation, we can determine the minimum size of formation so that it is large enough to contain the number of robots in the formation. The minimum size will depend on the number of robots in the formation, the desired formation shape and a minimum distance for collision constraint δ_{min}, if desired. It is possible that two formations with the same number of robots require different shape constraints.

For a rectangular formation, we use infinity norm:

$$n \leq (\lfloor \frac{s_w(n)}{\delta_{min}} \rfloor + 1)(\lfloor \frac{s_h(n)}{\delta_{min}} \rfloor + 1). \tag{3}$$

If we would like to additionally ensure graph completeness in the formation, we can enforce $max\{s_w, s_h\} \leq \delta_{max}$, where δ_{max} is the maximum distance at which communication can occur. Fig.1 illustrates an example of the maximum number of robots in a group. Here, $\frac{\delta_{max}}{\delta_{min}} = 4$, so that $n_{max} = (4+1)^2 = 25$ is the maximum number of robots in the group. Although these are general guidelines, choosing the shape constraints relies heavily on the robot formation.

3. TASK FEASIBILITY AND OPTIMALITY

3.1 Completeness of Planning Algorithm

The concept of task feasibility is closely related to completeness, which has been studied extensively.

Completeness In robot motion planning, an algorithm $\mathcal{A}$, for problem $\mathcal{P}$, accepts an instance of $\mathcal{P}$. $\mathcal{A}$ is complete for $\mathcal{P}$ if it is guaranteed to find a solution when one exists and to return failure otherwise.

From the definition, we can derive the two conditions for completeness:

Corollary: Two Conditions for completeness

1.If a solution exists, $\mathcal{A}$ should find it in finite time.

2. If no solution exist, $\mathcal{A}$ should terminate in finite time and report failure.

Roadmap methods and exact cell decomposition are complete. Given the robot's initial and goal representations, these approaches can compute a collision-free path if one exists; otherwise they report path non-existence. However, these methods are known to have a high theoretical complexity and are very difficult to implement. Thus their practical application have been limited to simple planar robots, convex polytopes or some other special shapes.

Incompleteness Algorithm $\mathcal{A}$ is incomplete, if it is not guaranteed to find a solution even when one exists.

One widely used approach for navigation problem is the naive potential field. It is known that this approach has the problem of falling into local minima which may cause it to terminate without finding a path when one exists. Thus this algorithm is incomplete. Algorithms that use a uniform grid or lattice to discretely sample a continuous solution space are also incomplete. There are two ways in which the algorithm may fail. First, it may overlook a solution that falls between lattice points. Second, it may terminate before considering a longer solution that lies on the lattice.

Some algorithms satisfy only the first condition for completeness: it is guaranteed to find a solution when one exists. Such algorithms will not overlook potential solutions; yet they are not necessarily complete. As suggested by John Reif, we use the term "exact" to describe such algorithms.

Exactness Algorithm $\mathcal{A}$ is exact, if it is guaranteed to find a solution when there exists one.

Approximate cell decomposition is exact. At each iteration the space is further divided in search of a collision-free path from start to goal. If a path exists, this algorithm will eventually find it. However, if a path does not exist, the algorithm may continue searching with finer and finer decompositions and will not terminate. For practical use, we terminate the decomposition at a prespecified resolution or when the number of subdivisions is sufficiently high. Thus, according to Latombe, we define resolution completeness as follows.

Resolution Completeness Algorithm $\mathcal{A}$ is resolution complete, if it is guaranteed to find a solution when a solution exists at that resolution.

However, resolution complete algorithms still suffer from the problem of heavy computational burden. For the general case of motion planning problem, a breakthrough was achieved with the development of sampling-based motion planners. These methods are easy to implement and can easily be applied to general robots with high D.O.Fs. The increased performance of these algorithms comes at the cost of relinquishing completeness. Those algorithms can only guarantee probabilistic completeness.

Probabilistically Complete Algorithm $\mathcal{A}$ is probabilistically complete for any robustly feasible path planning problem $(\mathfrak{R}_{free}, \zeta^o, \zeta^G)$, if

$$\lim_{n\to\infty} inf\mathbb{P}(\{\exists\ \zeta^g \in V_n^{ALG} \bigcap \zeta^G \quad s.t.\ \zeta^o\ is\ connected\ to\ \zeta^g\ in\ G_n^{ALG}\}) = 1. \quad (4)$$

Algorithm $\mathcal{A}$ is probabilistically complete for a robustly feasible path planning problem, if the limit

$$\lim_{n\to\infty} \mathbb{P}(\{\exists\ \zeta^g \in V_n^{ALG} \bigcap \zeta^G \quad s.t.\ \zeta^o\ is\ connected\ to\ \zeta^g\ in\ G_n^{ALG}\}) = 1. \quad (5)$$

On the other hand, the same limit is equal to zero if the problem is not robustly feasible.

In the definition, G_n^{ALG} is the graph constructed by the random roadmaps, and V_n^{ALG} is the set of vertexes on G_n^{ALG}.

3.2 Optimal Planning Algorithm

Based on the approximation of continuous space, different algorithms return solutions of different levels of optimality guarantee. The solution returned by complete algorithms like visibility graph are optimal. The exact algorithm as approximate cell decomposition can only guarantee optimality under the resolution. And algorithms like grid-A* and grid-D* can only guarantee optimality on the graph. Solution returned by PRM is optimal on the graph constructed by the random roadmap. While algorithm like RRT is non optimal.

In [11], Karaman proposed the definition of asymptotic optimality, that describes algorithm's ability to return solutions whose cost converge to the global optimum.

Asymptotic Optimality Algorithm $\mathcal{A}$ is asymptotically optimal if, for any path planning problem $(\mathfrak{R}_{free}, \zeta^o, \zeta^G)$ and cost function $c : \Sigma \to \mathbb{R}^*$ that admit a robustly optimal solution with finite cost c^*,

$$\mathbb{P}(\{\lim_{n\to\infty} sup Y_n^{\mathcal{A}} = c^*\}) = 1.$$

Note that, $Y_n^{\mathcal{A}}$ is the extend random variable corresponding to the cost of the minimum-cost solution included in the graph returned by $\mathcal{A}$ at the end of iteration n.

The proposed algorithms are analyzed for asymptotic optimality. It is proven that the PRM*, RRG and RRT* algorithms, as well as their k-neaerest versions, are all asymptotically optimal.

3.3 Algorithm Summary

Table 1 is a brief summary of the commonly used algorithms in motion planning.

The problem of determining whether task is feasible is closely related to the completeness of planning algorithm. For the purpose of planning in representation space, the intuitive choice would be complete algorithms. However, complete algorithms all suffered from expensive computational burden and poor scalability of dimension. The sampling based approach provide a solution for planning in representation space. These algorithms like RRT and PRM can be efficient even in high dimensional space and are faster compared to the complete algorithms.

The increased performance is at the cost of sacrificing completeness. Only probabilistic completeness is guaranteed for sampling based approach. While the probability of failure to find an existing solution converges to zero exponentially as the number of samples increasing. In practical application, we can set the number of iteration to be considerably high. Therefore, if the algorithm return no path, it is reasonable to declare that the task is infeasible with a high probability.

One problem with traditional sampling based approach is the poor quality of returned solution. While in many cases, one may be interested in solution paths of minimum cost. A breakthrough was achieved with the development of PRM* and RRT in [11]. The asymptotic optimality of these algorithms makes them good choice for motion planning in representation space.

Table 1. Algorithm Summary

Algorithm	Complete	Optimal	Good D.O.F. Scalability	Efficient Updates	Non-Holonomic
Navigation Function	no	yes	no	no	no
Grid-A*	no	grid	no	no	no
Grid-D*	no	grid	no	yes	no
Visibility Graph	yes	yes	no	no	no
RRT	probabilistic	no	yes	semi	yes
PRM	probabilistic	graph	yes	no	semi
RRT*	probabilistic	asymptotic	yes	semi	yes
PRM*	probabilistic	asymptotic	yes	no	semi

4. CONVERSION OF INFEASIBLE TASK

In this section, we investigate the problem of how to change an infeasible task so that the modified task can be performed with task objectives met. Two specific tasks are presented and the general framework of task oriented motion planning is used to find out what to do if the task is infeasible.

4.1 First Scenario

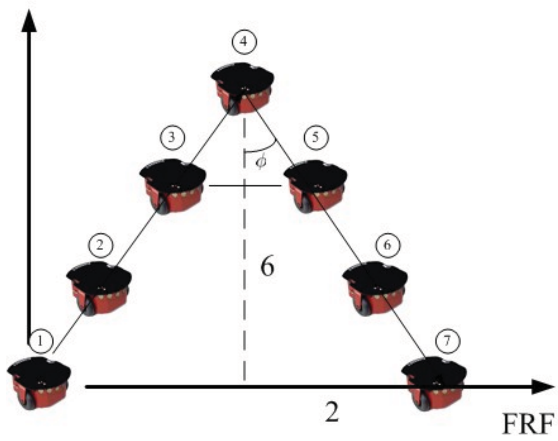
In the first simulation, seven robots are required to reach a target position while maintaining a formation as in Fig.4.2. For navigation task, position and orientation of the robot system are essential information. We choose the representation vector as $\zeta = (x, y, \theta)$, in which x and y are coordinates of the origin of FRF and θ is the orientation of FRF. From which position of all 6 other robots can be calculated using the 2D transformation matrix.

To address the motion planning problem in representation space, we use the RRT* algorithm. After 100,000 iterations, RRT* reports no path exists, therefore we say that task is infeasible. We choose to modify the formation shape constraint, as the formation shape requirement is more flexible. We reconstruct the representation model by adding the formation shape variable ϕ as a representation variable. ϕ is defined as half of the angle of the link that connect robot 4 and robot 7 and that connect robot 4 and robot 1. The representation vector then becomes $\zeta = (x, y, \theta, \phi)$. RRT* figures out the path. Therefore, task is feasible now in this new representation space. The process of task realization is shown in Fig. 2(a).

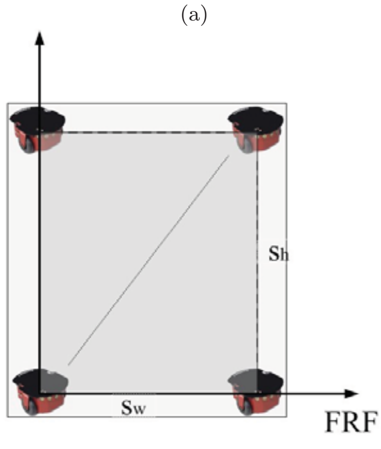
4.2 Second Scenario

This second scenario investigate the navigation of a rectangular formation which encloses 27 robots inside the formation boundary. The width $s_w = 6m$, and the length $s_h = 7.5m$. Robots that act as formation feature points maintain the formation boundary so that there is enough room for the number of robots in the formation. Formation reference frame is as in Fig.2(b).

In this simulation, the grey box at the left part of warehouse is movable. Let representation be $\zeta = (x, y, \theta)$. After 100,000 iterations, RRT* report no feasible path found, so the task is infeasible. To navigate to the target position, we first attempt to remove the box that blocks the narrow passage. So we use $disp$ to represent the displacement of box and make it one of the representation variables. We set $disp \in [-1,\ 9]$ and reconstruct the representation model



(a)



(b)

Fig. 2. (a)Formation of the first scenario. (b)Formation of the second scenario

as $\zeta = (x, y, \theta, disp)$. However, the task is still infeasible in the new representation space. What we do next is to take the formation shape constraint into consideration. Therefore we reconstruct the representation model as $\zeta = (x, y, \theta, disp, s_w, s_h)$. In which, s_w and s_h is the width and height of the rectangle maintained by the formation feature points. What is to be noticed is that formation shape must satisfy certain constraints. To ensure there is enough room for each individual robots, s_w and s_h must satisfy

$$(\lfloor \frac{s_w}{\delta_{min}} \rfloor + 1)(\lfloor \frac{s_h}{\delta_{min}} \rfloor + 1) \geq 27. \quad (6)$$

To maintain the complete information graph, s_w and s_h must satisfy

$$max\{s_w, s_h\} \leq 10. \quad (7)$$

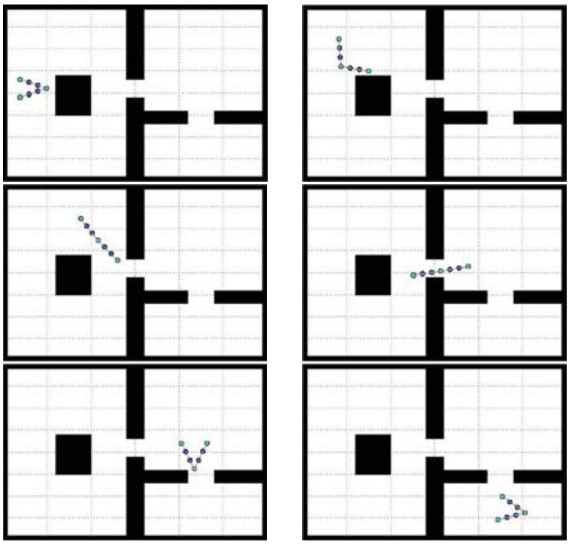


Fig. 3. Simulation result of first scenario

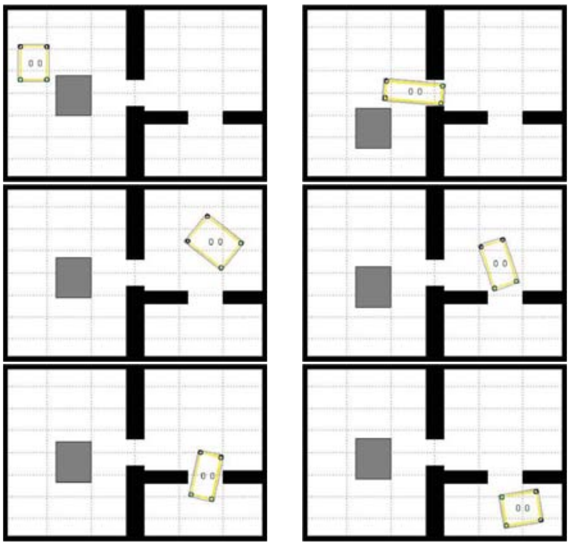


Fig. 4. Simulation result of second scenario

However, RRT* returns no feasible path in this new representation space. What is limiting the formation shape is the maximum distance for communication. If the communication module of robots is capable of communicating at larger distance, then the task is maybe feasible. To get an idea of the range of communication to make the task feasible, we add communication range com as one of the representation variable and reconstruct the representation model as $\zeta = (x, y, \theta, disp, s_w, s_h, com)$. Therefore, the communication constraint become

$$max\{s_w, s_h\} \leq com. \tag{8}$$

Task is finally feasible in this representation space. Fig.4 is the task realization in the warehouse. Two numbers in the picture indicate whether the constraints of formation size and communication are violated, 0 for not violated and 1 for violated. To navigate the formation to the target position in the warehouse, robot communication module should be able to communicate at a maximum distance of $13.46m$.

5. CONCLUSIONS

The concept of representation space is introduced to analyze the multi-robot formation task. Representation space of the task is built to characterize all the task related attributes with a hierarchical approach.Collision avoidance and formation constraints can be guaranteed. Moreover, feasibility of task can be evaluated in representation space. If the task is infeasible, robot system and/or task requirements must be modified to make it feasible. We present two scenarios to illustrate the process of turning an infeasible task into a feasible one. Ongoing work is investigating a variety of more complicated extensions to the basic formation navigation problem, including multiple non-unit obstacle costs and optimizing both path costs and constraint removal costs.

REFERENCES

[1] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol.52 , no.5, pp.863-868, 2007.

[2] M. A. Hsieh, V. Kumar, and L. Chaimowicz, "Decentralized controllers for shape generation with robotic swarms," *Robotica*, vol.26, no.5, pp.691-701, 2008.

[3] M. Chueh, Y. Yeung, K-P-C. Lei, and S. S. Joshi, "Following controller for autonomous mobile robots using behavioral cues," *IEEE Transaction on industrial electronics*, vol.55, no.8, pp.3124-3132 , 2008.

[4] J. Chen, D. Sun, J. Yang, and H. Chen, "Leader-follower formation control of multiple nonholonomic mobile robots incorporating receding-horizon scheme," *International Journal of Robotics Research*, vol.29, no.6, pp.727-747, 2010.

[5] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transaction on Automatic Control*, vol.51, no.3, pp.401-420, 2006.

[6] N. Ayanian and V. Kumar, "Abstractions and controllers for groups of robots in environments with obstacles," *Proceedings of IEEE International Conference on Robotics and Automation*, pp.3537-3542, 2010.

[7] J. Su and W. Xie, "Motion planning and coordination for robot systems based on representation space," *IEEE Transactions on Systems, Man, & Cybernetics*, Part B: vol.41, no.1, pp.248-259, 2011.

[8] K. Hauser, "The minimum constraint removal problem with three robotics applications," *Proceedings of International Workshop on the Algorithmic Foundations of Robotics*, to be published.

[9] M. Schwager, D. Rus, and J.-J. E. Slotine, "Decentralized, adaptive control for converage with networked robots," *International Journal of Robotics Research*, vol.28, no.3, pp.357-375, 2009.

[10] N. Ayanian, V. Kumar, and D. Koditschek, "Synthesis of controllers to create, maintain, and reconfigure robot formations with communication constraints," *Proceedings of International Symposium of Robotics Research*, vol.70, pp.625-642, 2011.

[11] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol.30, no.7, pp.846-894, 2011.