# Motion Planning for Multiple Mobile Robot Systems using Dynamic Networks

*Christopher M. Clark & Stephen M. Rock*
*Aerospace Robotics Lab*
*Department of Aeronautics & Astronautics*
*Stanford University*
*{chrisc, rock}@sun-valley.stanford.edu*

*Jean-Claude Latombe*
*Department of Computer Science*
*Stanford Universty*
*latombe@cs.stanford.edu*

## ABSTRACT

Presented is a new approach to multi-robot motion planning that is based on the concept of planning within dynamic robot networks. The system enables multiple mobile robots that have limited ranges of sensing and communication to maneuver safely in dynamic, unstructured environments. As the robots move about their workspace, localized robot groups form networks within which world models can be shared and centralized motion plans can be calculated in parallel. The motion planning algorithm used within networks is based on kinodynamic randomized motion planning techniques that construct trajectories in real-time. Both simulations and real robot experiments are used to validate the system.

## 1. INTRODUCTION

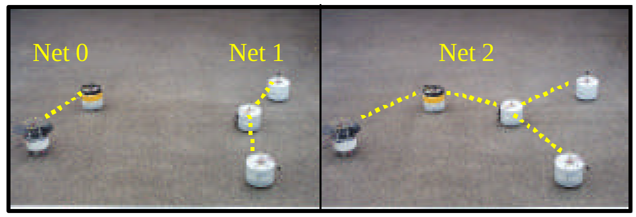
When large groups of robots are working together within unknown environments with moving obstacles, high-level motion planning is required to avoid collisions. In real systems, continuous inter-robot communication may not be feasible, and it is unlikely that a system of sensors can provide global knowledge. Also, in dynamic environments, the system must be able to react quickly. Hence, a motion planning system that uses local sensing and communication, distributes global information among robots, and can plan in real-time is required.

To meet these requirements, this paper introduces the concept of motion planning within dynamic robot networks that coordinate centralized planning. The system is based on the use of networks of robots that are capable of 1) forming dynamically whenever communication and sensing capabilities permit, 2) sharing world models within the networks, and 3) constructing trajectories for all robots in the network using a real-time motion planner. The remainder of this section is used to define what robot networks are and how they are formed. Outlined is the method for controlling planning within networks.

### *1.1 Network Formation Overview*

When any two robots are within communication range of each other, they establish a communication link. A network of robots is formed when two or more robots establish links between one another. Two robots in a network can then exchange information not just through direct communication links, but through any series of communication links within the network.

Figure 1a) illustrates an example environment involving 5 robots in which 2 networks have formed. In Net 1, the top and bottom robots can exchange information via their communication links with the middle robot.




a) b)

Figure 1: An example environment involving 5 robots. Dashed lines between robots indicate there exists a communication link. In b), two robots have moved and Nets 0 and 1 have combined to form Net 2.

Because robots are always moving towards their goal locations, the networks are dynamic. Robots may leave networks, enter different networks, and form new networks, (see Figure 1b). To handle this dynamic behavior, an application level protocol is used which ensures that robots within networks always have access to the local sensing information of all other robots in the network, and hence can share common world models.

### *1.2 Network Plan Control*

To control the planning process within each network, a protocol has been developed that enables parallel motion planning for all robots within a network. This process is initiated by any of the triggers listed below.

Triggers to Initiate a New Motion Plan

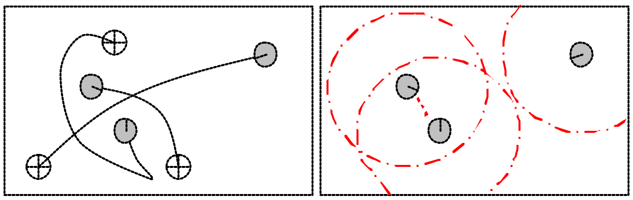
1) Two robots from different networks enter one another's range of communication resulting in the joining of two networks.
2) A significant change in the world model occurs.
3) A new goal location is requested.

When a new motion plan is required for robots in a network, (i.e. when any of the above triggers occur), data is exchanged between robots so that each robot has an updated world model that includes each robot's local world model and goal location. Once robots have shared information, each robot uses it's own centralized motion planner to construct trajectories for all robots in the network. Each robot then broadcasts it's plan to all other robots in the network. If any single robot cannot find a plan, then it broadcasts a "plan failure" message to inform other robots not to wait. After each robot has received a plan from all other robots, it will implement the best plan. To accomplish these steps, the following protocol is used.

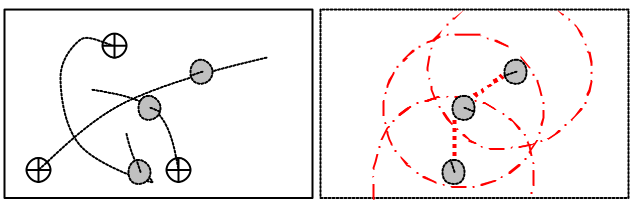
Application level Protocol: Plan Control

1) Broadcast latest world state information.
2) Construct a plan consisting of collision-free trajectories for all robots in the network.
3) Broadcast the plan to other robots in the network.
4) Implement the plan received that minimizes a predetermined cost function.

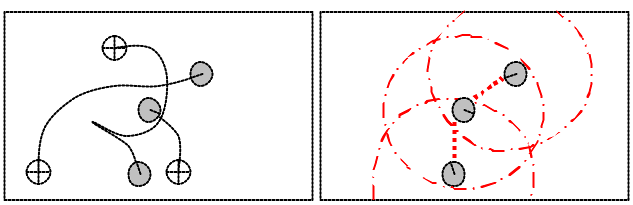
This process is illustrated through an example involving 5 robots, (see Figure 2 below).



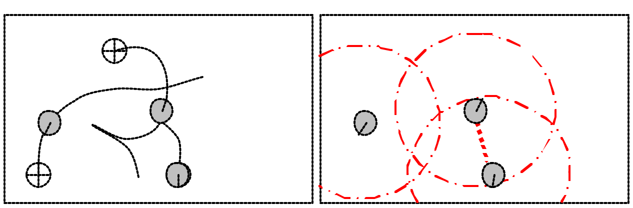
a) All three robots are following their initial trajectories. The two left robots are in communication range and have formed a network to create collision-free trajectories.



b) As the robots move along their trajectories, the middle robot enters communication range with the robot on the right and forms a larger network.



c) A new plan is made for all three robots in the network. The plan consists of collision-free trajectories for all three robots.



d) As robots continue along their new trajectories, they leave communication range of each other and some network connections are broken.

Figure 2: Top-down view of an example of planning within a multiple mobile robot system. Illustrations on the left depict top-down views of mobile robots as they follow their trajectories (dashed lines) to their respective goal locations (cross-hairs). Illustrations on the right depict the existing communication links (dotted lines) for robots that are in communication range (large circles) of one another.

*1.3 World Model Description*

Describing the world model in a concise but useful form is necessary to allow for information sharing between robots. For the implementation described in this research, world models consist of a list of robots, their descriptions, and a list of obstacles and their descriptions. The following table outlines the information stored in each list.

World Model Description

1) List of Robot Descriptions
   - State
   - Size (Radius)
   - Most Recent Update Time
   - Information Source
   - Goal location
   - Current Trajectory

2) List of Obstacle Descriptions
   - State
   - Size (Radius)
   - Most Recent Update Time
   - Information Source

Each object on the list, whether robot or obstacle, has an associated state, size, most recent update time and information source. Robots can report their own size and state, while obstacle sizes and states must be estimated using sensor data.

The most recent update time is used to keep track of the age of the object model. This is useful when updating world models with information received from other robots.

The information source is a robot identification number that keeps track of which robot sensed (or communicated with) the object. This is used to keep track of which robots are currently in the network.

Several assumptions had to be made to allow such a concise world model. The first is that all objects can be considered circular, or approximated by a set of circular objects. This allows the geometry to be described completely by one parameter (i.e. a radius).

The second assumption is that obstacles have constant velocity. When trajectories are constructed, they are built on the premise that the future position of the obstacle can be estimated to some precision. If at any later time this estimate diverges considerably from what was predicted during trajectory construction, then the robot calls for construction of a new plan.

A key assumption is that objects in the environment are identifiable (e.g. by sensors) and that any discrepancy between two world models can easily be resolved. Depending on the environment, this might not be a practical assumption and hence a different model will be required.

In the future, it is hoped that these assumptions can be relaxed to provide implementation in more complicated systems. Specifically, future work will involve developing more complicated models and handling their fusion.

The rest of the paper is organized as follows. A literature review is given in Section 2. A brief description of the KRMP algorithm used in trajectory construction is given in Section 3. Section 4 describes the Micro-Autonomous RoverS (MARS) test platform used for simulations and real robot experiments. Results from the simulations and experiments are presented in Section 5. In Section 6, conclusions are drawn and future work on the MARS test platform is discussed.

## 2. BACKGROUND REVIEW

*2.1 Centralized versus Decentralized*

The majority of previous work on multi-robot motion planning can be grouped into the categories of centralized and decentralized [1]. Centralized planners [9], [21] can be advantageous because they allow the possibility of completeness and global optimization. Examples of centralized planning that search for optimal solutions include [4], [5]. In [4] trajectories were constructed for each robot in a specific order such that each trajectory is collision-free of previously constructed trajectories. A search routine was used to find the order that provides shorter paths and in some cases was essential to finding a solution. In [5], two manipulator trajectories are coordinated to produce time-optimal trajectories.

A drawback of most centralized planners is that they are computationally intensive due to high dimensional configuration spaces. This drawback has lead to the development of several randomized methods that can search high dimensional configuration spaces quickly at the cost of losing optimality. For example, one randomized planning technique, Probabilistic Road Maps, was applied to multi-robot systems in [24]. Another example is [6] where randomization is used to coordinate the time stamps on independently constructed trajectories.

To relax the problem of searching through high-dimensional configuration spaces, several decentralized motion planning strategies have been developed. Their distributed nature allows for tractable solutions in high dimensional configuration spaces and even real-time planning [14].

Reactive style planning is one type of decentralized planning that has proven suitable for many applications because it is fast, enabling real-time planning. A common reactive approach is Potential fields [15]. This approach has been applied to both single robots and extended to multi-robot applications [25] including robot soccer [19]. A major drawback of most potential field methods is their susceptibility to deadlock.

Other types of decentralized planning coordinate trajectories that are independently constructed by each robot. One example of such planners is [13], where the trajectory for one robot is constructed irrespective of the other robot's trajectories. To avoid collisions, the robots maintain the same path they constructed earlier, but alter the velocities along their paths. Another example is [23], where a geometric based approach was taken to coordinate previously built trajectories. This algorithm demonstrated effective planning for a large number of robots in a confined area. However, as with the reactive style planners, these planners lack completeness.

Another drawback of decentralized planners is that they usually fail to find globally optimal solutions because they only plan to avoid local objects. Hence, many decentralized algorithms exist that search for near-optimal solutions. One example [10], uses the method of altering velocities described in [13] with D* to produce a distributed planner that tries to optimize trajectories. Also in [2], negotiations between localized groups of robots are used to assign priority orders to robots, that when applied to the planning algorithm, results in reduced trajectory lengths. The negotiation scheme in [2] demonstrates the benefits of localized inter-robot communication, and is the research most closely related to the robot network system presented later in this paper.

Ideally, one would like to use a centralized planner that provides optimality and completeness. However, the existing centralized planners assume unlimited communication abilities and, aside from randomized approaches, are computationally expensive and typically cannot be implemented in real-time. On the other hand, while decentralized planners have successfully demonstrated real-time planning, they still lack completeness and optimality.

This paper presents a system that exploits the advantages of both centralized and decentralized planning, while minimizing the disadvantages. It uses centralized randomized motion planning to increase completeness and still provide real time planning, but only does this within localized networks to avoid computational expense.

*2.2 Global versus Local Knowledge Availability*

While a large amount of attention has been paid to the differences between centralized and decentralized motion planning, less has been paid to the issue of whether or not a planner requires the availability of local or global knowledge.

Some planners assume global knowledge, e.g. are given maps or have global sensing, and global communication. In unknown environments, global knowledge is not available due to limited sensing capabilities. Also, in real systems global communication is often not possible due to range limitations and occlusions.

Other planners use only their local sensing for object detection and plan with only local knowledge of the environment. This avoids the problem of requiring global knowledge, but can lead to infeasible trajectories that were constructed using an incomplete world model.

Ideally, one would like robots to have available the world information necessary to plan with completeness. To increase the likelihood of robots having this information, the motion planning system proposed uses robot networks that exchange world model information whenever possible. For a given robot configuration, this maximizes the amount of world information available to each robot and allows for more complete plans.

## 3. MOTION PLANNING ALGORITHM

*3.1 Kinodynamic Randomized Motion Planning*
Probabilistic Road Maps (PRMs) are constructed by randomly selecting milestones from the robot's configuration space and connecting all milestones whose connecting paths are collision-free [12]. As described in [17], [11] and [16], this algorithm can be modified to accommodate kinodynamic constraints and accommodate moving obstacles by building a roadmap that includes time in the configuration space. This is known as Kinodynamic Randomized Motion Planning, (KRMP). Also shown in [11], is that under reasonable assumptions on the free space, the probability of not finding a plan when one exists decreases exponentially to 0 as the number of sampled milestones increases. The work demonstrated how randomized motion planners can successfully build kinodynamically-consistent trajectories for a single robot in real-time.

To enable the motion planning within networks, KRMP was used. Specifically, this paper extends the KRMP in [16] to accommodate several robots. Our previous work in [7], [8] uses a decentralized approach in which each robot independently constructs trajectories using KRMP. When robots enter each other's local area, they use a priority system to ensure they will remain collision-free**.** In this paper, the KRMP is extended further so that it may be used as a centralized planner within each local network of robots. By modifying the algorithm methods that 1) randomly select milestones in the roadmap for expansion, 2) generate new milestones for the roadmap, and 3) define the endgame region, the algorithm can be extended for use with multi-robot applications.

A key advantage that this system has over most distributed systems is that planning within the network is probabilistically complete. That is, each time a plan is created, the construction of that particular plan is probabilistically complete. Note that this does not mean that the system is probabilistically complete. Because robots may need to replan as they encounter new obstacles or robots, the overall completeness of the system is lost.

## 4. EXPERIMENTAL TEST-PLATFORM

*4.1 Micro-Autonomous RoverS Test-Platform*
Located in the Aerospace Robotics Lab at Stanford University, the Micro-Autonomous RoverS (MARS) test platform is used to model mobile robots in a two-dimensional workspace. The platform consists of a large 12' x 9' flat, granite table with six autonomous robots that move about the table's surface.

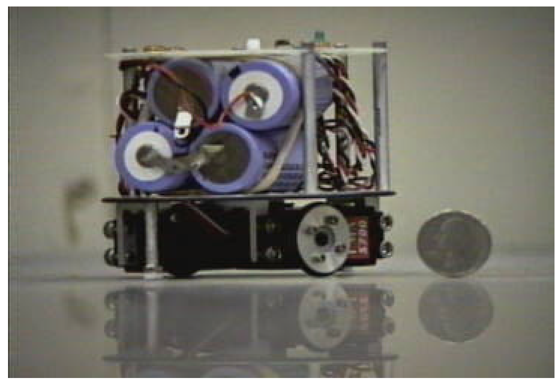


Figure 3: A rover from the Micro Autonomous RoverS test platform.

The robots are cylindrical in shape and use two independently driven wheels that allow them to rotate on the spot, but inhibit lateral movement so as to induce the nonholonomic constraint. Each robot is equipped with it's own Planner and Controller that are located off-board. To provide local sensing information to the robots, an overhead vision system with filtering is used.

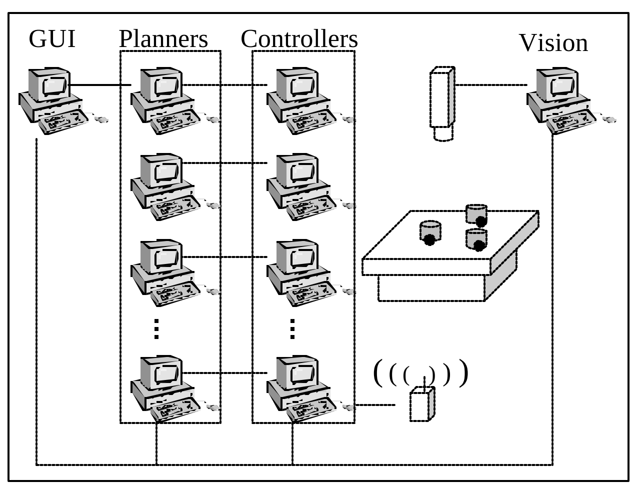



Figure 4: Network architecture of the MARS test platform

*4.2 Network Communication*
With all of the processing done off-board, the platform requires several computers distributed on a Local Area

Network (LAN). All communication within the LAN is accomplished with Real Time Innovation's Network Data Delivery Service (NDDS) software. Because a LAN is used for inter-robot communication instead of a wireless medium, there are no physical barriers to limit the range of communication. Hence the communication barrier is simulated.

NDDS is based on a publish/subscribe architecture. To broadcast messages by flooding a robot network, the sender will publish a message to which all robots subscribe. Before robots can receive their subscriptions, the messages are filtered so that only robots within some predetermined range of the sender will receive the message. This effectively simulates a discrete physical communication range.

*4.3 Sensors*

Not all robots on the MARS test platform are equipped with sensors. Instead, an overhead camera vision system is used to track the states of all objects on the table. The vision system processor calculates these states and publishes them to all applications that subscribe. This makes global state information available to all robots. To simulate a physical sensing range that occurs when using local sensors, the object states are filtered such that robots only receive state information regarding objects within some predetermined range of the robot.

## 5. EXPERIMENTS

To validate the planner's performance, it was implemented in both simulations and real robot experiments using the MARS test-platform. Simulation results demonstrated the system's ability to plan for up to 8 robots in cluttered environments involving 5 stationary and 5 moving obstacles. Results are summarized in Table 1.

| Number of Robots | Average Plan Time | Average Trajectory Time Constructed | Average Trajectory Time Used | Average number of Robots per plan |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |

Table 1: Simulation results

For 20 simulations involving 80 plans, the average time to plan was less than 10 ms for all plans recorded above.

To illustrate the advantage of using centralized versus decentralized planning within the networks, a classical example is given in which there is a very low probability of successfully coordinating independently constructed trajectories. One solution constructed by the planner is shown below.

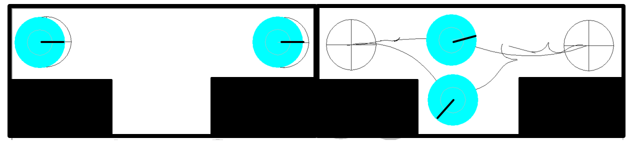


Figure 4: Two robots are within a confined workspace. Each robots goal location is coincident with the other robots start location. Using centralized KRMP within this network of two robots leads to a feasible plan.

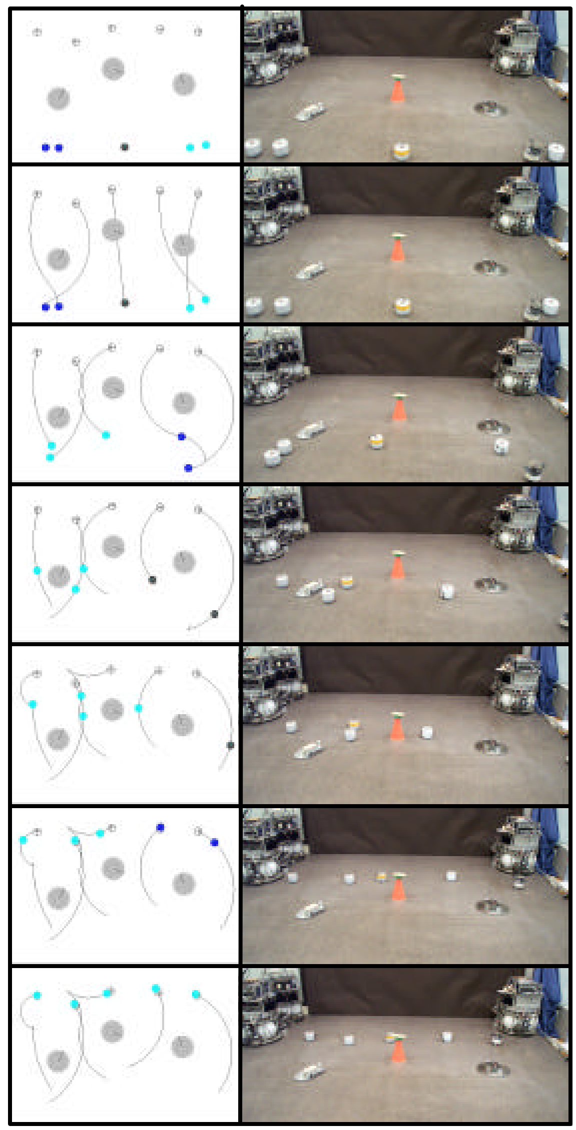


*Figure 5: Example experiment on the MARS test-platform involving 5 robots and 3 obstacles.*

To illustrate the applicability of the planner to a physical system, real robot experiments with up to 5 robots have been carried out. One example of such an experiment is illustrated in Figure 5. The left photos are screen-shots of the GUI taken throughout the experiment. The right photos show the physical hardware, and were taken at the same time as the corresponding GUI screen-shots. In the GUI, robots are depicted as small circles and obstacles are depicted as larger circles. Robot goal locations are indicated by cross-hairs, and lines leading to the goal locations depict the trajectories. When robots form a network as described in Section 2, it is indicated by a color change. Hence robots within a network have a common color, and this color will differ between networks.

In the experiment presented, all five robots are initially located at the close end of the table (i.e. bottom of the GUI screen). Communication and sensing ranges were limited to 0.75 m. Robot colors indicate that 2 networks have formed, one with the 2 robots in the bottom left and one with the 2 robots in the bottom right. As the experiment progresses, the robots follow their trajectories to eventually reach their goal locations at the far end of the table. Along the way, networks are continually changing as illustrated by the robots changing colors between frames. A result of this is real-time re-planning. This is illustrated by the fact that trajectories change between frames. Throughout the experiment, robots planned an average of 3.4 times, and planning times were an average of 9 ms.

## 6. CONCLUSIONS

The motion planner presented has demonstrated its effectiveness in planning for multiple mobile robots within a bounded workspace. It planned with a high probability of success, even in cluttered environments involving robots, stationary obstacles and moving obstacles. Planning times of less than 100 ms allowed the robots to re-plan in real-time and react quickly to changes in the environment.

Future work includes incorporating more sophisticated methods of modeling the environment into the communication system. Another future direction will be to investigate the effects of varying the ratio between sensor range and communication range. Also, while the application of the motion planner to surface rovers has been discussed, it should be noted that the planner is extendible to three-dimensional workspaces.

## BIBLIOGRAPHY


[1] T. Arai & J. Ota, "Motion Planning of multiple mobile robots," Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent robots and Systems, pg 1761-1768, 1992.

[2] K. Azarm & G Schmidt, "Conflict-Free Motion of Multiple Mobile Robots Based on Decentralized Motion Planning and Negotiation," *Proceedings of IEEE International Conference on Robotics and Automation*, pg 3526-3533, 1997.

[3] M. Bennewitz & W. Burgard, "A Probabilistic Method for Planning Collision-free Trajectories of Multiple Mobile Robots," *14th European Conference on Artificial Intelligence*, 2000.

[4] M. Bennewitz, W. Burgard & S. Thrun., "Optimizing Schedules for Prioritized Path Planning of Multi-Robot Systems," *Proceedings of the International Conference on Robotics and Automation*, 2001.

[5] Z. Bien & Jihong Lee, "A Minimum-Time Trajectory Planning Method for Two Robots," *IEEE Transactions on Robotics and Automation*, pg 443-450, 1992.

[6] S. Carpin & E. Pagello, "A Distributed Algorithm for Multi-robot Motion Planning," *Proceedings of the Fourth European Workshop on Advanced Mobile Robots*, pg 207-214, Lund (Sweden), September 2001

[7] C. Clark & S. Rock, "Randomized Motion Planning for Groups of Nonholonomic Robots," *International Symposium of Artificial Intelligence, Robotics and Automation in Space*, 2001.

[8] C. Clark, T. Bretl & S. Rock, "Kinodynamic Randomized Motion Planning for Multi-Robot Space Systems," *Proceedings of IEEE Aerospace Conference*, 2002.

[9] M. Erdmann & T. Lozano-Perez, "On Multiple Moving Objects," *Proceedings of IEEE International Conference on Robotics and Automation*, pg 1419-1424, 1986.

[10] Y. Guo & L. E. Parker, "A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots," *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pg 2612-2619, 2002.

[11] D. Hsu, R. Kindel, J. C. Latombe, & S. Rock, "Randomized Kinodynamic Motion Planning with Moving Obstacles," *Workshop on the Algorithmic Foundations of Robotics*, 2000.

[12] D. Hsu, J. C. Latombe, & R. Motwani, "Path planning in expansive configuration spaces," *Proceedings of the IEEE International Conference on Robotics and Automation*, pg 2719-2726, 1997.

[13] K. Kant, & S. Zucker, "Toward efficient Trajectory Planning: The path-velocity decomposition," *The International Journal of Robotics Research*, 5-3, pg 72-89,1986.

[14] S. Kato, S. Nishiyama, & J. Takeno, "Coordinating mobile robots by applying traffic rules", *Proceedings of the 1992 IEEE/RSH International Conference on Intelligent Robots and Systems*, pages 1535-1541, 1992.

[15] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *International Journal of Robotics Research*, 5, 1, pg 90-98, 1986.

[16] R. Kindel, D.Hsu, J. C. Latombe, & S. Rock. "Kinodynamic Motion Planning Amidst Moving Obstacles," *Proceedings of the IEEE International Conference on Robotics and Automation*, pg 537-544, 2000.

[17] S. M. Lavalle & S. A. Hutchinson, "Optimal Motion Planning for Multiple Robots Having Independent Goals," *IEEE Transactions on Robotics and Automation*, pg 14:912-925, 1998.

[18] S. M. Lavalle & J. J. Kufner, "Randomized Kinodynamic Planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.

[19] Lee, Lee, & Park, "Trajectory Generation and Motion Tracking for the Robot Soccer Game," *Proceedings of the 1999 IEEE International Conference on Intelligent Robots and Systems*, pg 1149-1154, 1999.

[20] V. J. Lumelsky & K. R. Harinarayan, "Decentralized Motion Planning for Multiple Mobile Robots: The Cocktail Party Model," *Autonomous Robots Journal*, No. 4, pg 121-135, 1997.

[21] T. Y. Li, & J. C. Latombe, "On-line manipulation planning for two robot arms in a dynamic environment", *Proceedings of the IEEE International Conference on Robotics and Automation*, 1995

[22] D. Parsons & J. Canny, "A Motion Planner for Multiple Mobile Robots", *Proceedings of IEEE International Conference on Robotics and Automation*, pg 8-13, 1992

[23] T. Simeon, S. Leroy & J.P. Laumond, "Path Coordination for Multiple Mobile Robots: a geometric algorithm," *Proceedings of the International Joint Conference on Artificial Intelligence*, 1999.

[24] P. Svestka, & M. H. Overmars, "Coordinated Path Planning for Multiple Robots," *Technical Report UU-CS-1996-43, Utrecht University, The Netherlands*, 13, 1996.

[25] C. W. Warren, "Multiple Path Coordination using Artificial Potential Fields," *Proceedings of the IEEE International Conference on Robotics and Automation*, pg 500-505, 1990.