IEEE 96 Conference on Robotics and Automation

Distributed Motion Control for Multiple Robotic Manipulators *

Kamal Souccar and Roderic A. Grupen
Laboratory for Perceptual Robotics
Department of Computer Science
University of Massachusetts

Abstract

This paper presents a method for coordinating the motion of multiple manipulators that share a common workspace. The approach employs independent motion controllers for each manipulator coupled by way of common configuration space constraints — each robot is represented as a constraint in the other configuration spaces. The harmonic function-based controller for each robot guarantees collision-free motion up to dynamic limits. Heuristic constraints are introduced to compensate for the lack of global information in the resulting distributed controller. The relationship between the distributed controller and the task planner is discussed and results of an experimental implementation are presented.

1 Introduction

The complexity of motion planning for multiple robotic systems grows exponentially as the number of degrees of freedom of the system increases. For this reason, it is clearly advantageous to decouple the control of multiple manipulators. In such systems, the manipulators often share goals that serve to coordinate their behavior and constraints arising from moving in a shared workspace. In a completely decoupled system, these constraints lead to scheduling approaches that plan motions for every manipulator independently and then schedule those motion plans in time to avoid collisions. In a loosely-coupled, distributed approach, instantaneous kinematic constraints derived from other cooperating robots can be imposed on each robot's controller. Potential problems associated with goal hiding or deadlock between loosely-coupled robots can be addressed by heuristics expressing general geometric characteristics of the multi-manipulator.

*This work is supported in part by NSF IRI-9116297, CDA-8922572 and IRI-9208920.

2 Related Work

Global methods generate complete trajectories for all robots *a priori* that guarantee collision-free paths to the goal. Erdmann and Lozano-Pérez [6] constructed the configuration space-time for multiple 2-dof planar manipulators. The trajectories of the manipulators are planned one at a time using the space-time swept volume of the previous trajectories as obstacles. Fortune, Wilfong, and Yap [7] found collision-free trajectories for two planar robots by characterizing the combinatorial structure of the configuration space of the two robots. These methods are computationally expensive since they have to consider both time and space. In addition, since the trajectories are generated off-line, they must be executed exactly and synchronously as to guarantee safety. Schwartz, Sharir, and Hopcroft's work [14, 10] on the closely related problem of planning trajectories for multiple moving objects has shown this problem to be PSPACE-hard: the time to achieve coordination grows exponentially with the number of moving objects.

Handey [13] employed a planner that produces a parametric trajectory for each robot. These solutions are then scheduled temporally to avoid collisions. Each robot contends for swept volumes of the workspace and blocks other robots that request overlapping volumes. This approach solves uncoupled spatial problems first, and then coordinates these solution in time. The spatial solution is constructed independently, however, not in the context of the desired coordinated motion. The resulting spatial trajectories could lead to frequent resource contention and while each plan is constructed in a relatively small configuration space, $n + 1$ plans (where n is the number of robots) must be constructed to satisfy constraints in space and time.

Local methods make instantaneous decisions on where each robot should be based on some continuously computed heuristic. Freund and Hoyer's controller coordinates multiple planar robots by incorporating collision constraints into the control system of the robots [8]. Tournassoud's technique for coordinating multiple moving robots defines separating planes at each moment and ensures that the robots stay in opposite half-spaces [15]. Since these methods are based on the instantaneous location of each robot, they can compensate for variations or inaccuracies in the execution of the trajectories. However, they cannot guarantee that each robot will reach its goal since they may reach a deadlock state where one robot is blocking the other.

The approach proposed in this paper makes use of distributed controllers, loosely coupled through kinematic constraints. As the configuration of the workspace changes, we advocate continuously updating constraints and incrementally recomputing controllers. Individual controllers are based on harmonic functions in their respective configuration spaces. We will show that in some applications, a coordinated motion plan can be computed by associating goals with controllers and need not concern the definition of trajectories at all.

3 Distributed, Collision-Free Motion Control

3.1 Harmonic Functions

Harmonic functions are solutions to Laplace's equation. They can be used to advantage for potential-field robot path planning, since they are smooth and do not exhibit spurious local minima [3]. Harmonic functions are known to have a number of other properties which are essential to robotics applications. These include:

Completeness - If a path exists from the current state to the goal, then it will be found up to the completeness and resolution of the configuration space model.

Robustness - Gradient descent on the harmonic function computed over an isotropic configuration space grid minimizes hitting probability[1] [2].

Responsiveness - Efficient numerical procedures for computing harmonic functions lead to high bandwidth in the path controller. As result, the artificial potential can track bandlimited disturbances in the geometric constraints. Newly observed motion constraints can be incorporated to incrementally modify the harmonic potential function.

[1] Hitting probabilities are random walk statistics regarding the probability of encountering a constraint before reaching the goal.

These harmonic potentials can be computed over arbitrary, discretized environments by very fast relaxation techniques. The relaxation times for computing potentials in the examples presented in this paper are in the millisecond range. Detailed discussion of numerical relaxation techniques and examples of robot control applications using this approach can be found in [4].

3.2 Configuration Space Representations

The configuration variables $\{q_0, q_1, \ldots, q_n\}$ of an articulated mechanism constitute a natural representation in which to represent motion plans. In this *configuration space*, the manipulator is mapped to a point [12]. The path planning problem is then posed as the construction of an obstacle-avoiding path from a start point to a goal point in configuration space.

A bitmap representation of the workspace suffices for computing the desired harmonic function. Two bits are used to distinguish obstacles, goals, and freespace in the configuration space grid. Each configuration coordinate corresponds to a volume in Cartesian space occupied by the manipulator. If any subset of the Cartesian bitmap occupied by the manipulator contains obstacle constraints, then the configuration space bitmap coordinate is marked as an obstacle.

For two, 2-dof manipulators ($n = 4$) this configuration space bitmap is four-dimensional, $boundary\_map[\theta_{00}][\theta_{01}][\theta_{10}][\theta_{11}]$, where $(\theta_{00}, \theta_{01})$ and $(\theta_{10}, \theta_{11})$ are the configuration variables for manipulator 0 and 1, respectively. Nodes in this grid are labeled to reflect the accessibility of unique system configurations. Therefore, by binding values to $(\theta_{00}, \theta_{01})$, for instance, the remaining sub-space parameterized by $(\theta_{10}, \theta_{11})$ defines Ψ_1 — the instantaneous *operational space* for manipulator 1. Likewise for manipulator 0. It is within these subspaces that we will implement concurrent harmonic function motion controllers.

3.3 Energy-Referenced Control

We will consider the class of controllers consisting of a model-based controller whose command accelerations were obtained from the gradient of an artificial potential. If we assume there exists a perfect dynamic model of the system, then each degree of freedom in the system can be linearized and decoupled in a feedforward dynamic compensator. Under these conditions, we may assume that the system behaves as a unit mass "marble" moving across the harmonic potential surface. This system has the following artificial Hamiltonian function:

$$H(q, p) = \frac{1}{2}||p||^2 + \phi(q) \tag{1}$$

where ϕ is an artificial potential, and q and p are configuration and momentum, respectively. This results in the following phase space trajectory constraints:

$$\begin{aligned} \dot{q} &= \frac{\partial H}{\partial p} &= p \\ \dot{p} &= -\frac{\partial H}{\partial q} &= -\nabla\phi \end{aligned} \tag{2}$$

Note that Equation 1 is essentially the total energy function for the dynamically compensated system. Thus, the reference acceleration is proportional to the negative gradient of the potential ϕ.

In practice, an energy-referenced damper is used to regulate system energy. This can be used to dissipate energy during convergent motion ($H_{ref} = 0$). The reference acceleration of this system is:

$$\ddot{q}_{ref} = -\nabla\phi \; - \; G_{KE} \; p \tag{3}$$

where G_{KE} is the derivative gain.

Obstacle potentials can be fixed at a uniform value, ϕ_{obs}, which is the maximum potential energy in the system. Under an energy-referenced control scheme, the system energy will never exceed the obstacle potential. If errors in the energy-reference controller are neglected, then as long as $H_{ref} < \phi_{obs}$, the system can not encounter a configuration space obstacle with non-zero velocity [5] .

3.4 Decomposition of Collision-Free Motion Control

Space Complexity — Consider an n dimensional system in which two $n/2$ dimensional robots interact. Assume that the grid resolution in each configuration space dimension is d. For each of d^n grid nodes, a boundary condition label indicates whether this configuration is an element of the constraint set, the goal set, or is free configuration space.

Consider the pair of 2-dof planar robots illustrated in Figure 1. The workspace map is represented in a single 4D array of boundary condition labels. Lookup tables for collision regions for each pair of robots are generated off-line and are indexed by the 4D configuration. Similar lookup tables were generated for fixed Cartesian workspace obstacles like tabletops, etc. In practice, a *point evaluation* method [11] is used to build these maps — the robot is placed in all possible configurations and intersections with objects are tested. Although very time consuming, it is simple and applicable to all types of manipulators. The boundary condition label can be encoded using 2 bits resulting in a space requirement $\frac{1}{4}d^n$ bytes.

The harmonic function is a double precision floating point computation requiring typically eight bytes to represent the artificial potential at each node in the grid. For the global, n dimensional grid this requires $8d^n$ bytes. In the $n/2$ dimensional operational spaces proposed, each robot requires $8d^{n/2}$ bytes.

Time Complexity — A similar savings can be realized in the time required to compute the harmonic function. The time complexity of the Successive-Over-Relaxation (SOR) procedure used to compute the harmonic function is roughly proportional to the total number of nodes, or $O(d^n)$ in the global operational space and $O(d^{n/2})$ in each of the $n/2$ dimensional operational spaces.

This approach greatly reduces both the space and time requirements of the planning and control problem. Moreover, the incremental cost of each robot is linear in both time and space for the distributed approach versus polynomial for the global problem.

3.5 Deadlock

In this context, deadlock occurs when the goals for a set of robots are mutually obscured by other robots in this set. In this situation, paths exist in the n dimensional configuration space that cannot be computed in the respective $\frac{n}{2}$ dimensional controllers. An example of this situation is presented in Figure 1. When no

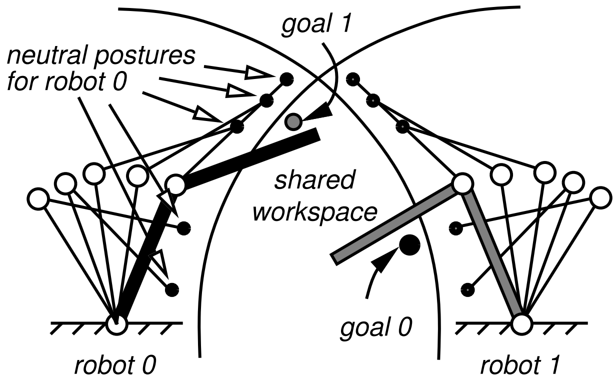

Figure 1: A Deadlock Situation Created By Mutually Obscured Goals.

goal is present in the accessible configuration space, the artificial potential at that point rises toward the obstacle potential. This results in a very simple means of testing for the existence of a path to a posted goal. However, instantaneously deadlocked robots may appear to have no goal when in fact those goals have been obscured by other robots.

To break deadlock and to introduce goals when no task related goals exist, a set of neutral positions are defined in each robot's configuration space. Figure 1 illustrates a set of candidate neutral postures for both robots. When a manipulator has no goal within its accessible workspace, heuristic goals are introduced at neutral postures just outside the shared workspace. Sets of candidate neutral postures can be arbitrarily large, but the harmonic function controller will always select goals that are "nearest" in the constrained configuration space. The result is a prioritized policy for each robot — (1) achieve a reference configuration, and (2) get out of the way of other robots if (1) is not possible.

4 Example: Distributed Motor Controllers in Blocks World

To illustrate the impact of the distributed controller on sensorimotor problem solving, consider the classical blocks world task where a stack of three blocks is to be disassembled and re-assembled in a new location.

Table 1: Blocks World Plans for 1, 2, and 3 Robots

step	1 robot 0	2 robots 0	2 robots 1	3 robots 0	3 robots 1	3 robots 2
1	get A	get A	get A	get A	get A	get A
	↓	↓		↓		
2	put A	put A	get B	∅	get B	get B
	↓	↓	↓	↓	↓	
3	get B	get C	∅	∅	∅	get C
	↓	↓	↓	↓	↓	↓
4	put B	put C	∅	∅	∅	put C
	↓	↓	↓	↓	↓	↓
5	get C	get A	put B	∅	put B	∅
	↓	↓	↓	↓	↓	↓
6	put C	put A	∅	put A	∅	∅
	↓					
7	get B					
	↓					
8	put B					
	↓					
9	get A					
	↓					
10	put A					

Table 1 summarizes plans generated by a simple A* search algorithm where the goal is to minimize the number of control steps and each action introduces a unit cost, empty goal sets (∅) are assumed to cost nothing. In our formulation, manipulators compete for the current set of goals — no manipulator trajectory in space or time, and no strict relationship between goals and manipulators must be established in the planner. These details are resolved at run-time by the distributed controller. For example, for two robots (0 and 1), "get A" is submitted to both robot 0 and to robot 1. The run-time system determines which robot will actually claim this goal — whichever robot blocks the other first wins. For clarity, the plans presented in Table 1 always assume that the lowest numbered robot competes successfully for the goal. Plans in Table 1 make no explicit resource allocations. The planner schedules a set of goals for each control step that satisfies precedence and resource constraints.

The performance of these plans is determined using the following cost analysis: $cost = \#actions + n(\#robots)$, where n is the number of unit cost per robot. Figure 2 summarizes this analysis using two values of n. It shows that either 2 or 3 robots appear to be plausible resource allocations depending on the relative expense of robot resources (a single robot will be preferable for very expensive robots). Allocating more than 3 robots will never be justifiable since the number of actions cannot be reduced further.

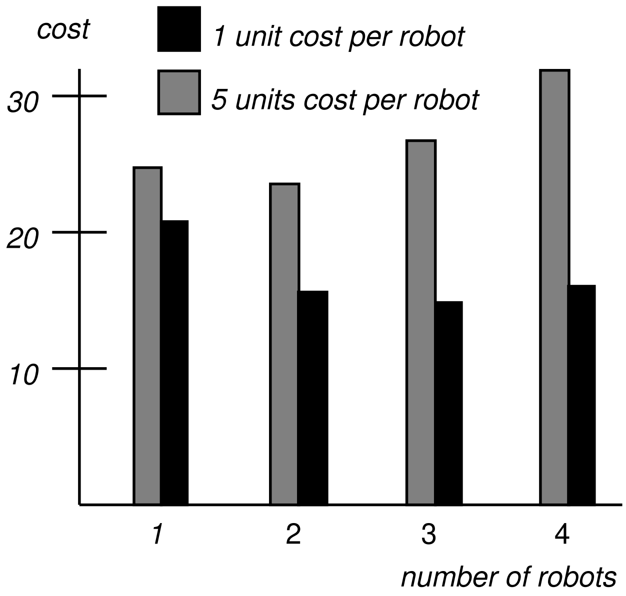

Figure 2: The Performance of Blocks World Plans

An experimental platform was constructed in which two, 4-dof GE-P50 robots manipulate blocks within a shared a workspace. The robot arms are equipped with two dextrous end effectors with which to handle the blocks. Details regarding grasp control and kinematic conditioning for this platform can be found in [1, 9]. The plans presented in Table 1 for 1 and 2 robots have been executed on this platform. A trace of the performance of the distributed controller is illustrated in Figure 3 for two manipulators.

The figure enumerates plan steps from top to bottom and action sequences determined on-line from left

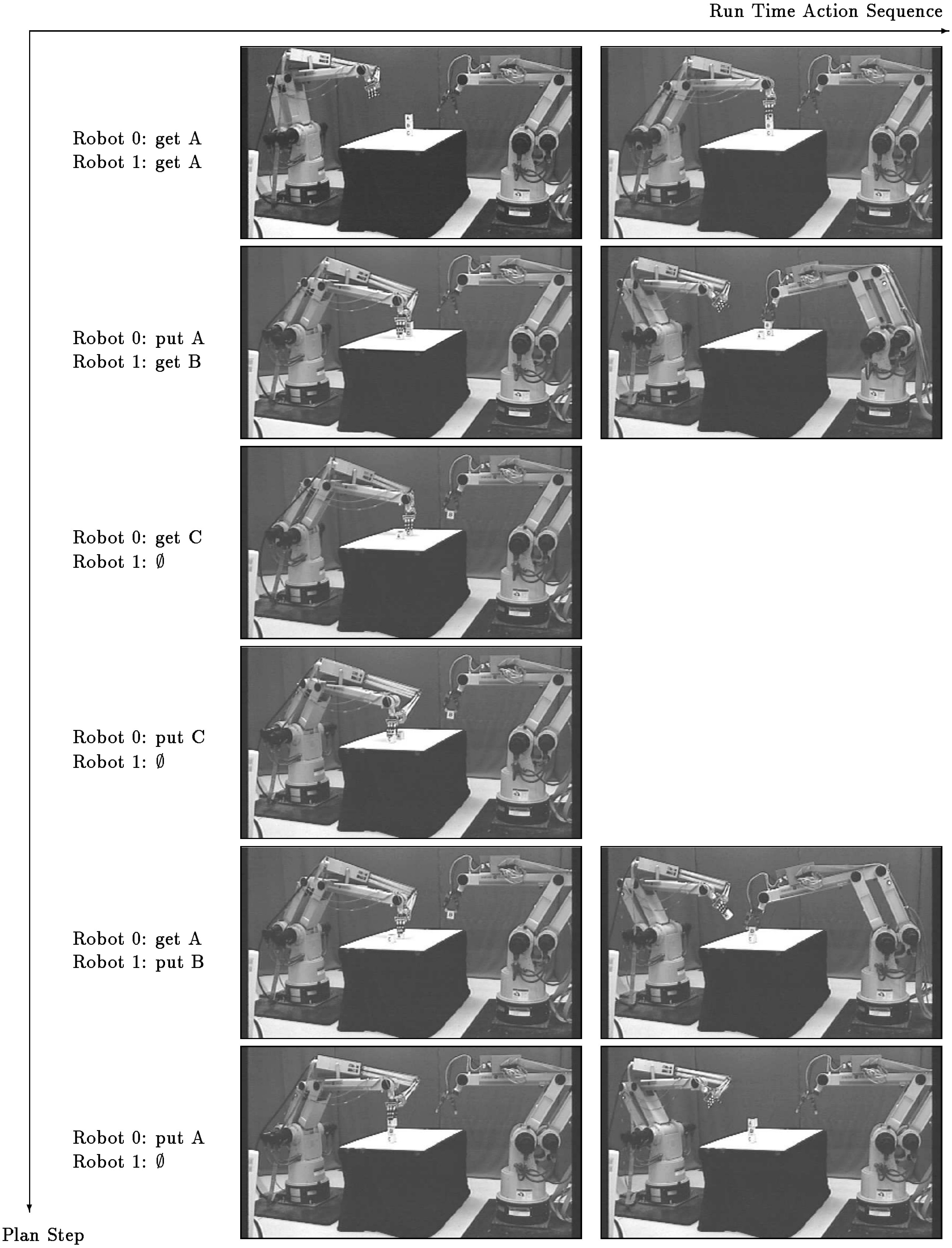

Figure 3: Coordinating the Behavior of Two 4-dof GE-P50 Robots in Blocks World.

to right. The trace unfolds in time from left to right and top to bottom. Initially, blocks A and B are stacked on top of block C at the far side of the table. The goal is to stack the blocks in the same order at the near side of the table. The sequence of operations taken by the system starts with the first robot removing block A from the top of the stack while the second robot is waiting until block B is clear. Block B is then picked up by the second robot as the first robot moves block A to a temporary position. Then, the second robot remains holding block B waiting for block C to be moved to its final position by the first robot. The second robot is then able to place block B in its final position atop block C as the first robot picks up block A from its temporary spot and places it on the top of the stack achieving the goal. A total of six steps were necessary to accomplish the goal with an execution time of about 3 minutes.

5 Conclusions

The main issue addressed in this paper is the coordination of multiple manipulators operating in a common workspace. A means of decomposing global configuration space into coupled subspaces was developed to achieve the coordination of distributed controllers for collision-free motion in multi-manipulator systems based on harmonic functions. In the context of assembly tasks, this method alleviates the burden on the planner to specify details of manipulator trajectories in space and/or time, reducing the complexity of the planner. In conjunction with the distributed controller, the planner reasons exclusively about precedence constraints and resource limitations in the multi-manipulator system.

References

[1] J.A. Coelho Jr. and R.A. Grupen. Control preimaging for multifingered grasp synthesis. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, CA, May 1994.

[2] C.I. Connolly. Harmonic functions and collision probabilities. In *Proc. 1994 IEEE Int. Conf. Intelligent Robots Systems*, pages 3015–3019, May 1994.

[3] C.I. Connolly and R.A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, Oct. 1993.

[4] C.I. Connolly and R.A. Grupen. The applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, October 1993.

[5] C.I. Connolly, K. Souccar, and R.A. Grupen. A hamiltonian framework for kinodynamic planning and control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, Aichi, Japan, May 1995.

[6] M.A. Erdmann and T. Lozano-Peréz. On multiple moving objects. *Algorithmica*, 2:477–521, 1987.

[7] S. Fortune, G. Wilfong, and C. Yap. Coordinated motion of two robot arms. In *Proc. 1986 IEEE Int. Conf. Robotics Automat.*, volume 1, pages 1216–1223, San Francisco, CA, 1986.

[8] E. Freund and H. Hoyer. Real-time pathfinding in multirobot systems including obstacle avoidance. *International Journal of Robotics Research*, 7(1), February 1988.

[9] R.A. Grupen and K. Souccar. Manipulability-based spatial isotropy: A kinematic reflex. In *Proceedings of the International Workshop on Mechatronical Computer Systems for Perception and Action*, pages 157–163, Halmstad, SWEDEN, June 1993.

[10] J.E. Hopcroft, J.M. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independant objects. *International Journal of Robotics Research*, 3(4):76–88, 1984.

[11] Y.H. Hwang and N. Ahuja. Gross motion planning – a survey. *ACM Computing Surveys*, 24(3):219–291, September 1992.

[12] T. Lozano-Pérez. Automatic planning of manipulator transfer movements. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(10):681–698, October 1981.

[13] T. Lozano-Peréz, J. L. Jones, E. Mazer, and P. A. O'Donnell. *Handey: A Robot Task Planner*. The MIT Press, Cambridge, MA, 1992.

[14] J.T. Schwartz, M. Sharir, and J. Hopcroft, editors. *Planning, Geometry, and Complexity of Robot Motion*. Ablex Publishing Corporation, Norwood, NJ, 1987.

[15] P. Tournassoud. A strategy for obstacle avoidance and its application to multirobot systems. *IEEE Transactions on Robotics and Automation*, pages 1224–1229, 1986.