Real-Time Obstacle Avoidance and Motion Coordination in a Multi-Robot Workcell

Oliver Brock   Oussama Khatib
Robotics Laboratory, Department of Computer Science
Stanford University, Stanford, California 94305
email: {oli, khatib}@CS.Stanford.EDU

Abstract

To exploit the potential benefit of multi-robot workcells, powerful motion planning and motion execution paradigms are necessary. The novel framework of elastic strips allows real-time obstacle avoidance and implicit motion coordination for multiple robots in a shared workspace. It augments motion plans with a reactive component allowing the avoidance of unpredictably moving obstacles. The obstacle avoidance behavior is task-dependent so that task behavior is not suspended to avoid obstacles. The motion coordination behavior of robots can also be specified in a task-dependent manner. Motion coordination can be achieved by regarding other robots as obstacles or by real-time modification of the trajectory's time parameterization. Multi-robot workcells can be programmed by planning the trajectories of all robots independently. Obstacle avoidance and motion coordination for the resulting trajectories are performed using elastic strips. The framework has been applied to the simulation of a multi-robot workcell.

1 Introduction

The flexibility and efficiency of the manufacturing process have a crucial impact on product development time and production cost. Multi-robot workcells can be employed to allow easy adaptation of the assembly process to new products, while maintaining high productivity. However, certain skills are required to allow the fast reconfiguration of such workcells. Since the location and the motion of parts within the workcell cannot be predicted precisely at planning time, robots need to exhibit reactive behavior to changes in the environment and unforeseen obstacles. In addition, the presence of multiple robots in the same workspace requires coordination of their motion to avoid collisions among them.

The generation of motion for articulated robots can be divided into a planning phase and an execution phase. Due to the high dimensionality of the conjunctive configuration-time space of multiple robots, the planning phase is computationally very expensive and algorithms have to assume that the environment is static or the motion of obstacles is known [5, 9, 10, 15]. If a change in the environment invalidates the current trajectory, a new trajectory has to be generated. The frequency of replanning can be reduced, if not eliminated, by augmenting the motion plan with a reactive component. The trajectory is then incrementally modified in accordance to the motion of obstacles.

A frequent approach to multiple-robot motion planning is to simplify the planning problem, resulting in unrealistically simple scenarios. One way to reduce the computational complexity of planning is to reduce the dimensionality of the configuration space, which can be achieved by projecting the robots into a plane [3, 11, 12]. These algorithms can only be applied to tasks in which the robot does not move above or beneath obstacles. A different way to simplify the planning problem is to restrict robots to move on independently planned roadmaps [10].

Assuming static obstacles, the motion planning problem of multiple robots can be solved using optimization techniques [2]. Gradient descent methods are applied to a quality function that incorporates kinematic constraints of the robot and stationary obstacles in the environment. Those methods are generally too computationally expensive to allow real-time control of a robot.

The problem of integrating the two phases mentioned above, the motion planning phase and the motion execution phase, was addressed by a layered multi-robot planning and control system [4]. A planner generates high level tasks that are then executed robustly on a low level controller. This abstraction increases flexibility, because the precise sequence of motion commands is determined at execution time, taking into account the latest information about the environment. Moving obstacles, however, could invalidate a plan and therefore require frequent replanning.

Once trajectories for multiple robots have been determined independently, they can be modified to avoid collisions between them. The velocity tuning approach [6] is a decoupled approach [9] to motion coordination that modifies the velocity of a robot along a fixed path so that all moving obstacles are avoided. The motion of these obstacles must be known. Path coordination [13] is also a decoupled approach to motion coordination. It performs velocity tuning for two robots in conjunction. It also assumes static obstacles. Independently planned trajectories for both robots are time-parameterized to avoid collisions.

Decoupled approaches are inherently incomplete. Centralized approaches, on the other hand, have to plan in the combined configuration-time space of the robots, a task that quickly becomes intractable.

The *elastic strip* framework [1] integrates motion planning and motion execution. The extensions to elastic strips presented in this paper result in a framework particularly well suited for motion execution for robots with many degrees of freedom in a shared and dynamic environment, like a multi-robot workcell. The augmented elastic strip framework allows for implicit motion coordination of multiple robots sharing the same workspace, combining the advantages of centralized and decoupled approaches.

2 Elastic Strip Framework

The elastic *strip* framework [1] is very similar in spirit to the elastic *band* framework [14]. A previously planned robot trajectory is modeled as elastic material. A path between an initial and a final configuration can be imagined as a rubber band spanning the space between two points in space. Obstacles exert a repulsive force on the trajectory, resulting in an incremental modification of the trajectory. This can be imagined as an obstacle pushing the rubber band. When the obstacle is removed, the trajectory will return to its initial configuration, just as the rubber band would.

An elastic *band* is represented as a one-dimensional curve in configuration space. This results in high computational complexity for high-dimensional configuration spaces. Furthermore, the specification of tasks for robots, particularly in assembly, is most naturally done in workspace. Elastic bands, however, represent a path in the configuration space.

The elastic *strip* framework operates entirely in the workspace in order to avoid aforementioned problems. The characterization of free space becomes more accurate in the workspace than in configuration space, resulting in a more efficient description of trajectories. In addition, by avoiding configuration space computation, the framework becomes applicable to robots with many degrees of freedom. The trajectory and the task are both described in workspace.

In the elastic strip framework a trajectory can be imagined as elastic material filling the volume swept by the robot along the trajectory. This strip of elastic material deforms when obstacles approach and regains its shape as they retract.

2.1 Free Space Representation

To guarantee that the current trajectory is entirely in free space or to modify it due to the motion of an obstacle, the free space around the swept volume of the robot along the trajectory must be known. The simplest representation of free space around a point $\mathbf{p}$ in the workspace is a sphere: it is described by four parameters and can be computed with just one distance computation. Such a sphere is called *bubble* [14] and is defined as

$$\mathcal{B}(\mathbf{p}) = \{\, \mathbf{q} : \|\mathbf{p} - \mathbf{q}\| < \rho(\mathbf{p})\},$$

where $\rho(\mathbf{p})$ computes the minimum distance from point $\mathbf{p}$ to the closest obstacle.

A set of bubbles is used to describe the local free space around the configuration q of a robot $\mathcal{R}$. This set $\mathcal{P}_q^{\mathcal{R}}$ is called *protective hull* $\mathcal{P}$ and is defined as

$$\mathcal{P}_q^{\mathcal{R}} = \bigcup_{\mathbf{p} \in \mathcal{R}} \mathcal{B}(\mathbf{p}).$$

Not every point $\mathbf{p}$ needs to be covered by a bubble. A heuristic is used for selecting a small set of points yielding an accurate description of the free space around configuration $\mathbf{q}$. An example of a protective hull around the Mitsubishi PA-10 manipulator is shown in Figure 1.

Along the trajectory $\mathcal{U}$ a sequence of configurations $q_1, q_2, \cdots, q_n$ is chosen. This sequence is called an elastic strip $\mathcal{S}_{\mathcal{U}}^{\mathcal{R}}$ if the union of the protective hulls $\mathcal{P}_i^{\mathcal{R}}$ of the configurations $q_i, 1 \leq i \leq n$ fulfills the condition

$$V_{\mathcal{U}}^{\mathcal{R}} \subseteq \mathcal{T}_{\mathcal{U}}^{\mathcal{R}} = \bigcup_{1 \leq i \leq n} \mathcal{P}_i^{\mathcal{R}}, \qquad (1)$$

where $V_{\mathcal{U}}^{\mathcal{R}}$ is the workspace volume of robot $\mathcal{R}$ swept along the trajectory $\mathcal{U}$. The union $\mathcal{T}_{\mathcal{U}}^{\mathcal{R}}$ of protective

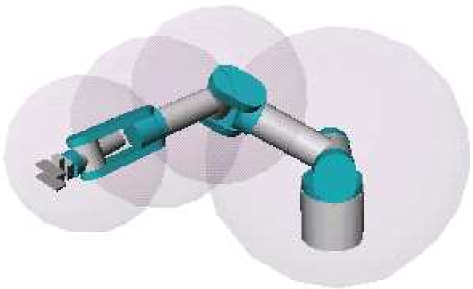

Figure 1: Protective Hull

hulls is called *elastic tunnel*. It can be imagined as a tunnel of free space within which the trajectory can be modified without colliding with obstacles.

An example of an elastic tunnel is shown in Figure 2. Three configurations represent snapshots of the motion along the trajectory. The union of the protective hulls around those form an elastic tunnel. It contains the volume swept by the robot along the trajectory. A gantry (not shown) moves the arm in the plane.

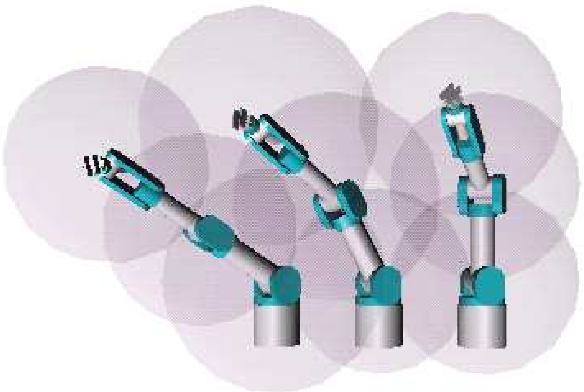

Figure 2: Elastic Tunnel

This representation of free space is the key to the performance of the elastic strip framework. It can be computed very efficiently, while giving a good approximation of the the actual free space.

2.2 Elastic Strip Modification

The elastic strip $\mathcal{S}$ is subjected to external forces that keep the trajectory free of collision and to internal forces that result in a short and smooth trajectory.

External forces are caused by a repulsive potential associated with obstacles. For a point $\mathbf{p}$ this potential function is defined as

$$V_{ext}(\mathbf{p}) = \begin{cases} \frac{1}{2}k_r(d_0 - \rho(\mathbf{p}))^2 & \text{if } \rho(\mathbf{p}) < d_0 \\ 0 & \text{otherwise} \end{cases},$$

where d_0 defines the region of influence around obstacles and k_r is the repulsion gain.

Internal forces are caused by virtual springs attached to control points on consecutive configurations along the elastic strip. How external and internal forces are used to modify the trajectory in accordance with the task specification is explained in section 3.

As obstacles approach the trajectory, the size of protective hulls decreases. If this leads to the violation of equation 1, intermediate configurations are inserted into the elastic strip, until the swept volume of the robot is again entirely covered by protective hulls. The retraction of obstacles, on the other hand, leads to the enlargement of protective hulls. In this case redundant configurations are removed.

3 Motion Behavior

An assembly task in a multi-robot workcell consists of different subtasks, each potentially requiring different motion behavior. Using the elastic strip framework these subtasks can be described in a very intuitive way. A force $\mathbf{F}_{task}$ defining the task is specified in operational space [7] at the end-effector. This force can be derived from the potential V_{task} associated with the task. Joint torques Γ_{task} required to accomplish the task can be computed by a simple mapping of $\mathbf{F}_{task} = \nabla V_{task}$ acting at the end-effector point $\mathbf{e}$ using the Jacobian $J^T(\mathbf{q})$ at that point for configuration $\mathbf{q}$:

$$\Gamma_{task} = J^T(\mathbf{q})\mathbf{F}_{task}. \qquad (2)$$

Operational space control of redundant manipulators can accommodate different kinds of motion behavior. In the simplest case a part has to be moved on any trajectory between two locations. To implement obstacle avoidance, the existing trajectory can be modified to accommodate unforeseen or moving obstacles, or to evade another robot sharing the same workspace. No particular motion behavior is required to accomplish the task and the joint torques $\Gamma_{task'}$ can be computed by adding the torques resulting from internal and external forces to equation 2:

$$\Gamma'_{task} = J^T(\mathbf{q})\mathbf{F}_{task} + \sum_{\mathbf{p}\in\mathcal{R}} J^T_{\mathbf{p}}(\mathbf{q})\mathbf{F}_{\mathbf{p}}, \qquad (3)$$

where $\mathbf{F}_{\mathbf{p}}$ is the sum of internal and external forces action at point $\mathbf{p}$ and $J^T_{\mathbf{p}}(\mathbf{q})$ is the Jacobian at that point in configuration $\mathbf{q}$. Internal forces are caused by the potential function V_{int} associated with the virtual springs of the simulated elastic material and external forces can be derived from the potential function V_{ext} resulting from the proximity of obstacles: $\mathbf{F_p} = \nabla V_{int} + \nabla V_{ext}$.

Redundancy of a robot with respect to its task can be exploited to integrate task behavior and obstacle avoidance behavior. Holding a subassembly, inserting a subassembly, or tracing the surface of an assembly requires the end-effector to remain stationary or to move along a given trajectory without deviation. This entails that obstacle avoidance cannot alter the motion of the end-effector.

For redundant systems the relationship between joint torques and operational forces is given by

$$\Gamma''_{task} = J^T(\mathbf{q})\mathbf{F}_{task} + \left[I - J^T(\mathbf{q})\overline{J}^T(\mathbf{q})\right]\Gamma_0, \qquad (4)$$

with $$\Gamma_0 = \sum_{\mathbf{p}\in\mathcal{R}} J^T_{\mathbf{p}}(\mathbf{q})\mathbf{F_p}$$

and $$\overline{J}(\mathbf{q}) = A^{-1}(\mathbf{q})J^T(\mathbf{q})\Lambda(\mathbf{q}),$$

where $\overline{J}(\mathbf{q})$ is the dynamically consistent generalized inverse [7]. The term $[I - J^T(\mathbf{q})\overline{J}^T(\mathbf{q})]$ corresponds to the null space of the Jacobian $J(\mathbf{q})$. This relationship provides a decomposition of joint torques into two dynamically decoupled control vectors: joint torques resulting in forces acting at the end-effector ($J^T(\mathbf{q})\mathbf{F}_{task}$) and joint torques that only affect internal motions $\left(\left[I - J^T(\mathbf{q})\overline{J}^T(\mathbf{q})\right]\Gamma_0\right)$ [8].

Using the decomposition of equation 4, the end-effector can be controlled by a force ($\mathbf{F}_{task}$) in operational space, whereas internal motions can be independently controlled by joint torques (Γ_0) that do not alter the end-effector's dynamic behavior.

Using this framework, obstacle avoidance behavior is implemented by exploiting the redundant degrees of freedom of the robot without influencing end-effector motion. If kinematic constraints of the robot make obstacle avoidance impossible the task needs to be aborted or modified. This can occur when the robot reaches the border of its workspace, for example.

4 Motion Coordination

In a realistic setting for a flexible, multi-robot workcell few assumptions can be made about the motion of obstacles. This creates the need for a constantly updated representation of local free space around the robot and its trajectory. Motion coordination could be achieved by simply regarding other robots and their trajectories as obstacles that influence the local free space and therefore modify the elastic strip. This approach is called *trajectory modification*. In tight environments, however, independently planned trajectories of different robots might pass through the same narrow passage. Hence, this approach would lead to unavoidable collisions and require replanning.

4.1 Velocity Tuning

Given an elastic strip, the current dynamic state of the robot, and its dynamic limitations, an approximate time-parameterization of the trajectory can be computed. This temporal information about the trajectories of multiple robots can be exploited for motion coordination. Using *velocity tuning* [6, 13] for multiple robots, collisions can be avoided by simply changing the time-parameterization of the elastic strip and the trajectory it represents.

Figure 3 shows an example of velocity tuning. Two independently planned trajectories are shown in part a). The time-parameterization is indicated by numbers. At time $t = 3$ a conflict is detected. An arbitrary or task-dependent prioritization of the trajectories is imposed to avoid circular dependencies. By changing the time parameterization of the robot with lower priority, as shown in Figure 3b), the conflict is resolved.

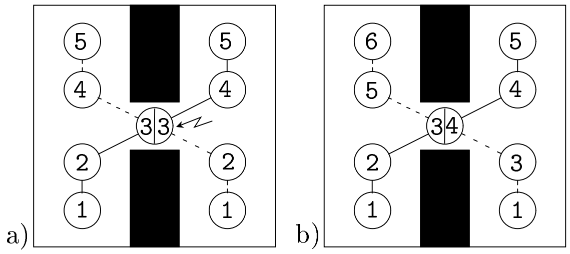

Figure 3: Velocity Tuning

Collision checking between configurations of different robots is efficiently performed using the protective hull of a configuration. To determine a conflict only configurations that represent the robot's configuration at overlapping time intervals need to be considered for collision. Nevertheless, for n robots there are $O(tn^2)$ possible collisions that have to be considered, where t is the number of time intervals. As a result, this approach becomes computationally inefficient for a large number of robots. In a multi-robot workcell, however, n is usually a small number and this algorithm remains practical.

The velocity tuning approach will fail, if two robots are attempting to switch positions or if two trajectories have the same priority, indicating that the task does not permit a change of time-parameterization.

4.2 Velocity/Trajectory Tuning

Velocity tuning and trajectory modification by themselves can fail to resolve a conflict. Velocity tuning might also result in a suboptimal conflict resolution when one robot has to wait for a long time for another robot to pass the region of conflict. These disadvantages can be eliminated by combining velocity tuning and trajectory modification.

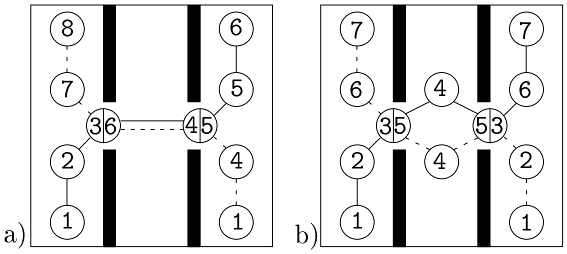

Figure 4: Velocity/Trajectory Tuning

In Figure 4a) two independently planned trajectories are shown after the existing conflict has been resolved using velocity tuning only. Robot 2 has wait for robot 1 to pass through the second room. Figure 4b) shows the trajectories that result from trajectory modification.

Depending on the tasks of the robots, velocity tuning or trajectory modification are the preferred approach to motion modification. If one approach should fail, however, both approaches can be applied in conjunction to attempt conflict resolution. Motion coordination is performed strictly based on task-specification. Potential motion conflicts and moving obstacles do not need to be considered during planning. The elastic band incorporates velocity tuning and trajectory modification to result in implicit motion coordination.

Velocity/trajectory tuning combines decoupled planning with centralized trajectory modification. This results in a motion coordination approach that avoids the computational expense of planning in the combined configuration-time space of multiple robot. However, it resolves motion conflicts beyond the capabilities of decoupled approaches. The trajectories for all the robots in the workcell can determined independently and are coordinated by the elastic strip framework.

The need for replanning, however, is not eliminated entirely. For two robots exchanging their position in a narrow corridor the planning algorithm has to generate a motion for both robots into an open area to perform the exchange. The need for replanning can be detected using equation 1. During the modification of the elastic strip external forces and the insertion of intermediate protective hulls assure a collision-free path. If equation 1 remains violated the chosen trajectory is topologically not feasible any more and replanning is necessary.

5 Experimental Results

The elastic strip framework has been implemented and tested in simulation with different robotic platforms, such as the Stanford Mobile Platform, a mobile base equipped with a PUMA 560 robot [1]. In Figure 5 the application to a multi-robot workcell is shown. The workcell consists of two Mitsubishi PA-10 robotic arms on gantries with nine degrees of freedom each, sharing the same workspace. The gantries are not explicitly shown; they enable the arms to move in the plane of the floor of the assembly cell. The gantry increases the redundancy of the robot with respect to the task and hence allows better obstacle avoidance behavior. The elastic strip framework can equally well be applied to stationary manipulators.

Figure 5 shows the incremental modification of the trajectory of a robot to avoid a collision with another manipulator, moving into its path. The configurations of the elastic strip are indicated by lines connecting the joint frames of the robot. Notice how more intermediate configurations are used to represent the trajectory where the robots are closer to each other. This is due to the decreasing size of protective hulls. Figure 5a) shows the initial configuration. Figure 5b) and 5c) show the modification of the elastic strip as a reaction to the right robot raising its end-effector; the two configurations show the first robot just before and after passing the rising arm. The last image shows the robot in the goal position defined by the elastic strip. The base of the right robot remains stationary throughout the entire experiment.

The current implementation runs on a single 400MHz PentiumII. In wide open spaces only few configurations are required to represent the trajectory and the elastic strip can be updated at rates exceeding 100Hz. In very tight and complex environments this update rate can drop to 5-10Hz.

6 Conclusion

The elastic strip framework is an efficient real-time obstacle avoidance approach for robots with many degrees of freedom in dynamic environments. It integrates planning and execution to result in a motion ex-

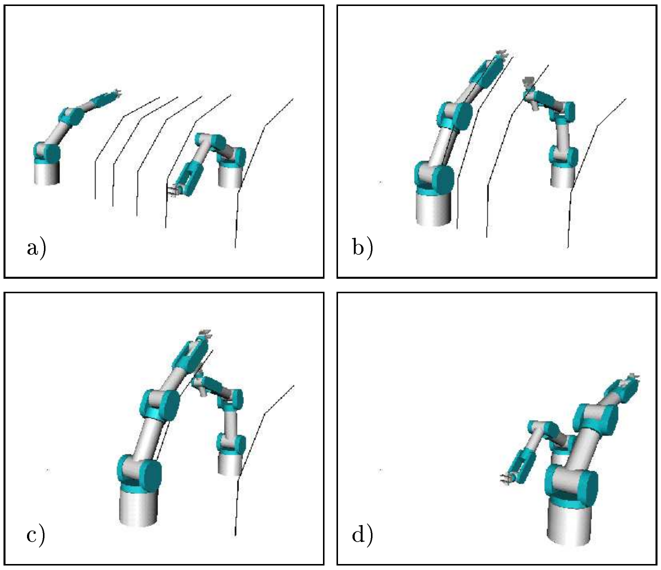

Figure 5: Two Mitsubishi PA-10 arms in a workcell

ecution algorithm that incorporates task constraints.

Elastic strips have been extended to allow motion coordination of multiple robots operating in a shared workspace. Velocity tuning as well as trajectory modification are used in concert to generate powerful, implicit motion coordination that does not violate constraints imposed by the tasks of the individual robots.

The approach to motion coordination is decoupled in the sense that planning is performed for each individual robot without considering interactions. It hence avoids complex computation in combined configuration-time space. Its performance in practice, however, can be compared with centralized approaches, as replanning can be avoided in most situations. Programming of a multi-robot workcell is performed by planning the trajectory of each robot independently, ignoring collisions with moving obstacles and other robots. Obstacle avoidance and motion coordination are performed in real-time using the elastic strip framework. Experimental results were presented using a multi-robot workcell with two Mitsubishi PA-10 arms on gantries.

Acknowledgments

The authors would like to thank Kyong-Sok Chang, Bob Holmberg and Diego Ruspini for their helpful insights and discussion in preparing this paper. The financial support of Boeing and Nomadic Technologies is gratefully acknowledged.

References

[1] Oliver Brock and Oussama Khatib. Executing motion plans for robots with many degrees of freedom in dynamic environments. In *Proc. Int. Conf. on Robotics and Automation*, volume 1, pages 1–6, Leuven, Belgium, 1998.

[2] Arben S. Çela and Yskandar Haman. Optimal motion planning of a multiple-robot system based on decomposition coordination. *IEEE Trans. on Robotics and Automation*, 8(5):585–96, 1992.

[3] H. Chu and H. A. ElMaraghy. Real-time multi-robot path planner based on a heuristic approach. In *Proc. Int. Conf. on Robotics and Automation*, volume 1, pages 475–80, 1992.

[4] H. Chu and H. A. ElMaraghy. Integration of task planning and motion control in a muli-robot assembly workcell. *Robotics and Computer-Integrated Manufacturing*, 10(3):235–55, 1993.

[5] Yong K. Hwang, Pang C. Chen, Chong W. Lee, and Mun S. Kim. Complete motion planner for time-varying environments. In *Proc. Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 513–19, 1996.

[6] K. Kant and S. W. Zucker. Toward efficient trajectory planning: Path velocity decomposition. *Int. J. of Robotics Research*, RA-2(3):135–45, 1986.

[7] Oussama Khatib. A unified approach to motion and force control of robot manipulators: The operational space formulation. *Int. J. of Robotics Research*, 3(1):43–53, 1987.

[8] Oussama Khatib, Kazu Yokoi, Kyong-Sok Chang, Diego Ruspini, Robert Holmberg, and Arancha Casal. Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In *Proc. Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 546–53, 1996.

[9] Jean-Claude Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1991.

[10] Steve M. Lavalle and Seth A. Hutchinson. Optimal motion planning for multiple robots having indepenent goals. In *Proc. Int. Conf. on Robotics and Automation*, volume 3, pages 2847–52, 1996.

[11] Tsai-Yen Li and Jean-Claude Latombe. On-line manipulation planning for two robot arms in a dynamic environment. *Int. J. of Robotics Research*, 16(2):144–167, 1997.

[12] Claudio Mirolo, Enrico Pagello, and Wen-Han Qian. Simplified motion planning strategies in flexible manufacturing. In *Proc. Int. Symp. on Assembly and Task Planning*, pages 394–9, 1995.

[13] P. A. O'Donnell and Tomás Lozano-Pérez. Deadlock-free and collision-free coordination of two robot manipulators. In *Proc. Int. Conf. on Robotics and Automation*, pages 484–89, 1989.

[14] Sean Quinlan and Oussama Khatib. Elastic bands: Connecting path planning and control. In *Proc. Int. Conf. on Robotics and Automation*, volume 2, pages 802–7, 1993.

[15] Charles W. Warren. Multipe path coordination using artificial potential fields. In *Proc. Int. Conf. on Robotics and Automation*, volume 1, pages 500–5, 1990.