Elastic Strips: A Framework for Integrated Planning and Execution

Oliver Brock Oussama Khatib
Robotics Laboratory, Department of Computer Science
Stanford University, Stanford, California 94305, U.S.A.
email: {oli,khatib}@cs.stanford.edu

Abstract: The execution of robotic tasks in dynamic, unstructured environments requires the generation of motion plans that respect global constraints imposed by the task while avoiding collisions with stationary, moving, and unforeseen obstacles. This paper presents the elastic strip framework, which addresses this problem by integrating global motion planning methods with a reactive motion execution approach. To maintain a collision-free trajectory, a given motion plan is incrementally modified to reflect changes in the environment. This modification can be performed without suspending task behavior. The elastic strip framework is computationally efficient and can be applied to robots with many degrees of freedom. The paper also presents experimental results obtained by the implementation of this framework on the the Stanford Mobile Manipulator.

1. Introduction

The automated execution of tasks by robots in dynamic, unstructured, and potentially populated environments requires sophisticated motion capabilities. In such environments unforeseen or moving obstacles may repeatedly invalidate a previously planned motion. The successful completion of a task will then require the frequent generation of revised motion plans that accommodate changes in the environment. In highly structured environments this problem can be avoided by imposing constraints on the motion of obstacles: it is common for motion planning algorithms to assume that obstacles are stationary or moving on predetermined trajectories [13]. These assumptions are unrealistic for environments outside the factory floor or the laboratory environment, like those encountered in robotic applications such as service or field robotics.

The necessity of frequent regeneration of motion plans for task execution in unstructured environments entails a computational difficulty for the generation of robot motion. Planning algorithms generally take on the order of minutes to determine a plan, under ideal circumstances still on the order of seconds [8]. Hence, using planning algorithms the avoidance of moving or unforeseen obstacles cannot be guaranteed. Reactive motion schemes, like the potential field approach [9], on the other hand, are able to avoid moving obstacles in real-time but might fail to achieve the task by getting trapped in local minima.

This dichotomy of complete or resolution complete but inefficient global planning algorithms and incomplete but efficient local, reactive execution approaches has been the starting point for various efforts to improve the performance of robot motion generation algorithms [11]. In one approach the concepts of potential field-based obstacle avoidance and approximate cell decomposition motion planning were used in conjunction to yield a reactive framework for planning and execution of robot motion [5]. Only partial knowledge of the environment is required and small, unforeseen obstacles or minor changes in the environment can be tolerated.

Whereas the previously mentioned approach can only be applied to mobile robots, the elastic band framework [14] is also suited for robot manipulators. In this approach planning and reactive execution are used in sequence: first a plan is generated using a conventional motion planner; subsequently this plan is modified incrementally as a reaction to changes in the environment. Real-time obstacle avoidance for robots with few degrees of freedom has been demonstrated.

For mobile robots the global dynamic window approach integrates planning and execution by incorporating an efficient global path planner into the control loop of a dynamics-based execution scheme [3]. This approach allows for high-speed navigation of a mobile base in an unknown and dynamic environment.

In another approach to integrated planning and control a plan is converted into a trajectory using a path-based parameterization, rather than a time-based one [15]. This allows the corresponding path-based controller to interrupt the execution of the trajectory, if an unforeseen obstacle is detected. Once the obstacle has been removed or an evasion maneuver has been specified by a human operator, the execution of the original trajectory is resumed.

The elastic strip framework [2] presented in this paper integrates planning and execution for robots with many degrees of freedom. This approach allows for reactive obstacle avoidance behavior that does not suspend task execution, enabling robots with many degrees of freedom to perform tasks in dynamic and unstructured environments. The paper also presents the experimental validation of this framework on the Stanford Mobile Manipulator, a six degree-of-freedom (DOF) PUMA 560 mounted on a 3 DOF holonomic mobile base.

2. Elastic Strip Framework

The elastic *strip* framework is very similar in spirit to the elastic *band* framework [14]. In the elastic *band* framework a previously planned robot motion is modeled as elastic material. A path between an initial and a final configuration can be imagined as a rubber band spanning the gap between two points in space. Obstacles exert a repulsive force on the trajectory, resulting in an incremental modification. This can be imagined as a moving obstacle pushing and deforming 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 leads to high computational complexity for high-dimensional configuration spaces. Furthermore, the specification of tasks for robots 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 that 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 volume swept by the robot along the trajectory must be known. The simplest representation of free space around a point 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

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

where $\rho(p)$ computes the minimum distance from p to an obstacle.

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

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

Not every point 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 q. An example of a protective hull around the Stanford Mobile Manipulator is shown in Figure 1 a).

Along the trajectory $\mathcal{U}$ a sequence of configurations $q_0, \cdots, q_n$ 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

$$V_{\mathcal{U}}^{\mathcal{R}} \subseteq \mathcal{T}_{\mathcal{S}}^{\mathcal{R}} = \bigcup_{0 \leq i \leq n} \mathcal{P}_i^{\mathcal{R}}, \tag{1}$$

where $V_{\mathcal{U}}^{\mathcal{R}}$ is the workspace volume of robot $\mathcal{R}$ swept along the trajectory $\mathcal{U}$. The union $\mathcal{T}_{\mathcal{S}}^{\mathcal{R}}$ of protective 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 1 b). Three configurations represent snapshots of the motion along a trajectory. The union of the protective hulls around those configurations form an elastic tunnel. It contains the volume swept by the robot along the trajectory.

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

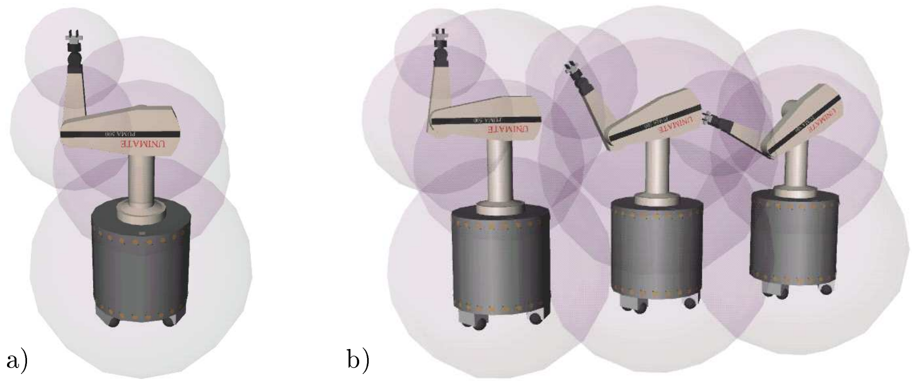

Figure 1. a) A protective hull around the Stanford Mobile Manipulator b) An elastic tunnel formed by several overlapping protective hulls

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 p, this potential function is defined as

$$V_{ext}(p) = \begin{cases} \frac{1}{2}k_r(\rho_0 - \rho(p))^2 & \text{if } \rho(p) < \rho_0 \\ 0 & \text{otherwise} \end{cases},$$

where ρ_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 described in section 2.3.

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.

The integration of planning and execution paradigms in the elastic strip framework consists of the incremental and reactive modification of a global plan. As the forces acting on the elastic strip do not change the topological properties of the represented trajectory, the global properties of the plan are maintained. The robot can be guaranteed to reach the goal as long as the plan remains topologically feasible. Hence, an efficient, reactive scheme has been integrated with a global plan.

2.3. Motion Behavior

A task can consist 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 F_{task} is specified in operational space [10] 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 $F_{task} = -\nabla V_{task}$ acting at the end-effector point e using the Jacobian $J(q)$ at that point for configuration q:

$$\Gamma_{task} = J^T(q) F_{task}. \tag{2}$$

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. No particular motion behavior is required to accomplish the task and the joint torques Γ_{task} can be computed by adding the torques resulting from internal and external forces to equation 2:

$$\Gamma_{task} = J^T(q) F_{task} + \sum_{p \in \mathcal{R}} J_p^T(q) F_p,$$

where F_p is the sum of internal and external forces action at point p and $J_p(q)$ is the Jacobian at that point in configuration 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: $F_p = -\nabla V_{int} - \nabla V_{ext}$.

Redundancy of a robot with respect to its task can be exploited to integrate task behavior and obstacle avoidance behavior. Some tasks may require 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

$$\Gamma_{task} = J^T(q) F_{task} + \left[I - J^T(q) \overline{J}^T(q) \right] \Gamma_0, \tag{3}$$

$$\text{with} \qquad \Gamma_0 = \sum_{p \in \mathcal{R}} J_p^T(q) F_p \qquad \text{and} \qquad \overline{J}(q) = A^{-1}(q) J^T(q) \Lambda(q),$$

where $\overline{J}(q)$ is the dynamically consistent generalized inverse [10]. The term $[I - J^T(q)\overline{J}^T(q)]$ corresponds to the null space of the Jacobian $J(q)$. This relationship provides a decomposition of joint torques into two dynamically decoupled control vectors: joint torques resulting in forces acting at the end-effector $(J^T(q)F_{task})$ and joint torques that only affect internal motions $\left(\left[I - J^T(q)\overline{J}^T(q)\right]\Gamma_0\right)$ [12]. This allows to control the end-effector by a force (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.

This framework is integrated with the elastic strip approach by simulating the effect of internal and external forces on the robot at the configurations $\{q_0, \cdots, q_n\}$ representing the elastic strip. The elastic strip can now be modified in a way that avoids obstacles by exploiting the redundant degrees of freedom of the robot but leaves the end-effector motion unaltered. 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.

3. Trajectory Generation and Execution

An elastic strip $\mathcal{S}$ represents a sequence of discrete configurations $\{q_0, \ldots, q_n\}$ along a path from q_0 to q_n. The conversion of such a sequence into a time-parameterized trajectory is a well-studied problem [1, 6]. In the elastic strip framework, however, this sequence is changing in discrete steps as the elastic strip is modified in reaction to changes in the environment. Let q_t and $\dot{q}_t$ be the vectors of joint positions and joint velocities of the robot at time t. They resulted from the execution of the trajectory described by the elastic strip $\mathcal{S}_{t-1}$ at time $t-1$. The modified elastic strip $\mathcal{S}_t$ will represent a different trajectory with a desired velocity of $\dot{q}'_t$ at time t and a potentially unattainable position q'_{t+1} and velocity $\dot{q}'_{t+1}$ at time $t+1$. Due to the comparatively slow rate at which the update of the elastic strip occurs, the difference between the current and desired position and velocity, $\|q_t - q'_t\|$ and $\|\dot{q}_t - \dot{q}'_t\|$, can be large. Hence, the application of conventional approaches to trajectory execution would not result in desirable behavior.

This problem could be solved by either requiring the initial portion of the elastic strip to remain constant or by limiting modification to those trajectories achievable, given the dynamic constraints of the robot. These solutions have two disadvantages: For one, invalidating the path represented by the elastic strip requires a computationally expensive replanning operation, which should be avoided if at all possible. Hence, it is desirable for the elastic strip to represent a valid path, even if inconsistent with the current state of the robot. Secondly, when executing a motion on a mobile base, such as the Stanford Mobile Manipulator, execution errors accumulate for the mobile base due to slippage of the wheels. To reduce this error various relocalization schemes can be employed. The error accumulated between different relocalizations can be expected to be large enough to invalidate a trajectory originally incorporating dynamic constraints imposed by the robot. Insufficient actuator capabilities could also be regarded as a source of error.

To avoid these disadvantages a novel approach to the execution of changing trajectories is presented in this section. The general idea is to maintain a valid trajectory that corresponds to the elastic strip but ignores the dynamic state of the robot. The initial configuration q_0 of the strip will be constrained to coincide with the configuration of the robot. Hence, the trajectory represented by the strip will be almost correct, only ignoring dynamics. The final trajectory then results from merging the robot's current motion with the previously computed trajectory. These two steps are detailed in the two following subsections for a single degree of freedom. The extension to many degrees of freedom is trivial.

3.1. Generating the Trajectory

The elastic strip represents a sequence of configurations $\{q_0, \ldots, q_n\}$ that are connected by straight-line segments in joint space. The discontinuous velocity change that can occur at a configuration q_i along the piecewise linear trajectory cannot be executed by the robot without coming to rest at q_i. As this is not desirable, an interpolation technique is applied to convert the piecewise linear trajectory into one with a continuous first derivative. As we can expect frequent velocity changes during the execution of a changing trajectory, this interpolation is performed with cubic polynomials, which also guarantee a continuous second derivative within a given segment of the trajectory.

When using a standard scheme for trajectory generation with cubic polynomials [6], the trajectory passes through a set of via points, corresponding to the configurations q_i along the elastic strip. This may results in a large deviation from the straight-line trajectory between two adjacent configurations. Due to the free space description with protective hulls, however, this is the portion of the trajectory where the free space description is most narrow. It is hence desirable for the robot to follow this portion of the trajectory as closely as possible. To achieve this, the straight-line segments will be connected by *cubic turns*. The maximum allowed duration of a turn at configuration q_i can be inferred from the local free space and determines the velocity along the adjacent line segments from q_{i-1} to q_i to q_{i+1}.

To allow the execution of trajectories in tight spaces, a *turning cubic* will always begin by accelerating with the maximum acceleration $\ddot{q}_{max}$. Taking into account that the acceleration during a cubic can be described by a line, the duration d of the turning cubic is computed as follows:

$$\Delta\dot{q} = |\dot{q}_{i-1} - \dot{q}_i| = \int_0^d \ddot{q}(t)\, dt = \frac{d\, \ddot{q}_{max}}{2} \qquad \rightarrow \qquad d = \frac{2 \cdot |\Delta\dot{q}|}{\ddot{q}_{max}},$$

where $\dot{q}_i$ denotes the velocity along the line segment between q_i and q_{i+1}. The points at which the turning cubic connects with the adjacent line segments can be computed by equating their motion equations:

$$q_i + (d_i - d_{\Delta i})\dot{q}_i + \Delta q = q_{i+1} + (d - d_{\Delta i})\dot{q}_{i+1},$$

where d_i denotes the duration of the line segment between q_i and q_{i+1} assuming constant velocity $\dot{q}_i$, and Δq stands for the change in joint position between beginning and ending of the turning cubic. Solving for $d_{\Delta i}$, the parameters of the turning cubic are computed. Its execution will begin at time $t_{i+1} - d_{\Delta i}$ and end at time $t_{i+1} + (d - d_{\Delta i})$, where t_i denotes the time at which execution of the ith segment begins. Using those values the starting and ending point of the turning cubic with respect to the line segments can be computed.

3.2. Merging the Robot's Motion with the Trajectory

The trajectory resulting from the method described in the previous subsection does not take into account the current velocity of the robot. Also, the robot's position might have changed due to the correction of accumulated execution error. To connect the robot to this trajectory, an iterative optimization algorithm could be used [7]. This algorithm retains the original time parameterization of the trajectory, which causes the robot to "catch up" with the trajectory. In dynamic environments it is unreasonable to impose a time constraint for the robot to reach the goal configuration. Therefore we will reconnect the robot's current state to the trajectory according to its dynamic capabilities and then adapt the time-parameterization of the remainder of the trajectory.

The cubic segment that merges the current state of the robot with a segment on the trajectory can be computed by equating the position and the velocity equations of the merging segment and the segment on the trajectory. When merging with a line segment the following equations result:

$$\begin{aligned} q_0 + d\, \dot{q}_0 &= q_r + \dot{q}_r\, d + a_1\, d^2 + a_2\, d^3 \\ \dot{q}_0 &= \dot{q}_r + 2\, a_1\, d + 3\, a_2\, d^2, \end{aligned}$$

where q_r and $\dot{q}_r$ are the position and the velocity of the robot, d is the duration of the *merging cubic*, and a_1, a_2 are its coefficients. When attempting to merge with another cubic segment of the original trajectory, a solution can be found in a similar fashion. The duration d of the merger has to be determined such that the acceleration constraints of the robot are not violated.

4. Experimental Results

The elastic strip framework was implemented and tested on the Stanford Mobile Manipulator, a 9 degree-of-freedom (DOF) robotic system, consisting of a PUMA 560 arm mounted on a holonomic mobile base. The robot was controlled using the Robotics Library [4] running on a dedicated on-board 90 MHz Pentium PC. The algorithms presented in this paper were executed on a 400 MHz Pentium PC. In the example shown below, update rates of the elastic strip varied between 10 and 100 Hz.

To demonstrate the elastic strip framework, the Stanford Mobile Manipulator was commanded to move five meters along the x-axis of the global coordinate frame, while keeping the arm's posture. During the execution of this plan an unforeseen obstacle, another Stanford Mobile Manipulator, forces the first robot to deviate from the original plan. It is modified using the elastic strip framework to avoid collision while achieving the desired goal configuration. Two different perspectives of the simulated modification of the trajectory are shown in Figure 2. A sequence of snapshots from the execution on the real robot can be seen in Figure 3. The plot of the base trajectory, as well as the trajectory for the first three joints of the PUMA 560 manipulator are shown in Figure 4.

5. Conclusion

The elastic strip framework is an efficient method for motion execution for robots with many degrees of freedom in dynamic environments. It allows obstacle avoidance behavior without the suspension of task execution. Hence,

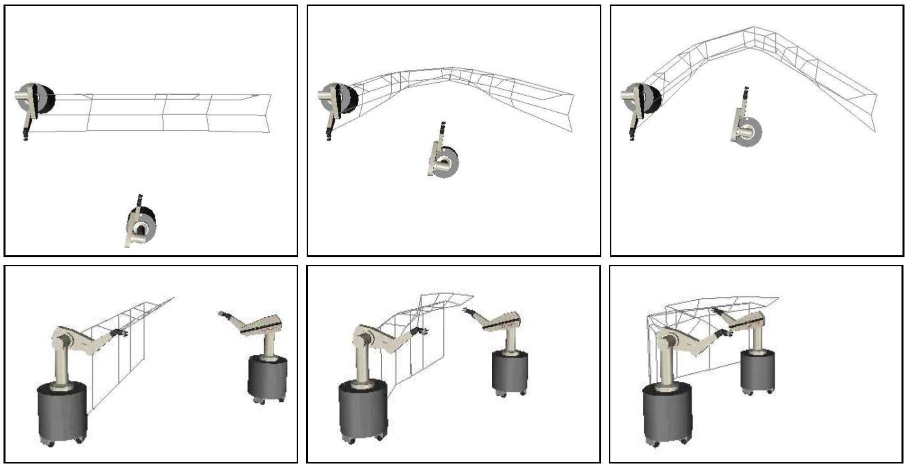

Figure 2. The elastic strip, shown as gray lines, is modified incrementally in order to maintain a valid path while avoiding a moving obstacle

Figure 3. Execution of a plan using the elastic strip framework; the path is modified in real-time to avoid the obstacle

this approach is well suited for mobile manipulation. The effectiveness of the proposed algorithm is derived from an efficient integration of planning and execution methods. The validity of the algorithm has been experimentally verified using the Stanford Mobile Manipulator, a 9 DOF robotic system, consisting of a mobile base and a manipulator arm.

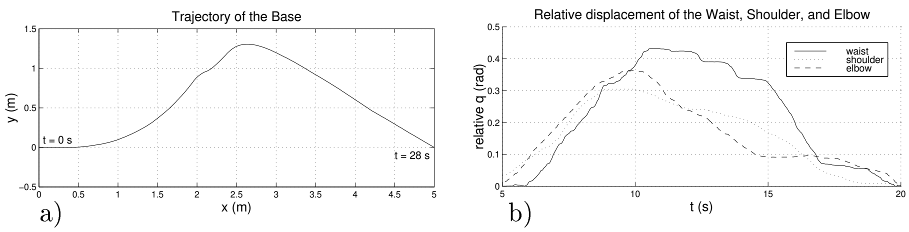

Figure 4. a) Trajectory of the base b) Relative displacement of the arm

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. In particular we would like to thank Kyong-Sok Chang for helping to gather the experimental data. The financial support of Boeing and NSF (grants IRI-9320017 and CAD-9320419) is gratefully acknowledged.

References

[1] James. E. Bobrow, S. Dubowsky, and J. S. Gibson. Time-optimal control of robotic manipulators along specified paths. *Int. J. of Robotics Research*, 4(3):3–17, 1985.

[2] 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, 1998.

[3] Oliver Brock and Oussama Khatib. High-speed navigation using the global dynamic window approach. In *Proc. Int. Conf. on Robotics and Automation*, 1999. To appear.

[4] Kyong-Sok Chang. *Robotics Library*. Stanford University, 1998. Dynamic control and simulation library.

[5] Wonyun Choi and Jean-Claude Latombe. A reactive architecture for planning and executing robot motions with incomplete knowledge. In *Proc. Int. Conf. on Intelligent Robots and Systems*, volume 1, pages 24–29, 1991.

[6] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 2nd edition, 1989.

[7] Elisabeth A. Croft, Robert G. Fenton, and Beno Benhabib. Time-optimal interception of objects moving along topologically varying paths. In *Proc. Int. Conf. on Systems, Man, and Cybernetics*, volume 5, pages 4089–94, 1995.

[8] Lydia E. Kavraki, Peter Švestka, Jean-Claude Latombe, and Mark H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.

[9] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. of Robotics Research*, 5(1):90–8, 1986.

[10] 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.

[11] Oussama Khatib. Towards integrated planning and control. In *Proceedings of IFAC Symposium on Robot Control*, volume 1, pages 305–13, 1994.

[12] 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.

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

[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] Tzyh-Jong Tarn. Path-based approach to integrated planning and control for robotic systems. *Automatica*, 32(12):1675–87, 1996.