Proceedings of the 2009 IEEE
International Conference on Robotics and Biomimetics
December 19 -23, 2009, Guilin, China

Reactive Deformation of Path for Navigation Among Dynamic Obstacles

Anand TM

Abstract—**This paper presents experimental results of a reactive deforming path algorithm, based on elastic strips, in a 2D / 3D simulator. This algorithm generates motion plans that respect global constraints while making sure that the robot avoids collisions with all stationary, moving and unexpected obstacles. The problem is addressed by integrating a global planning algorithm with a local reactive approach. The planner performs path planning in two stages: (i) Global planning of the path is done using a heuristic search algorithm such as A*. (ii) In a dynamic environment, with several obstacles moving around in an unpredicted fashion, the planner deforms the path reactively by subjecting it to artificial forces. For making sure that the robot is always on a collision free path, the plan is incrementally updated with changes in the environment. The plan representation uses deformable links connecting the robot's configurations along the path, stretching or retracting the path as required to avoid collisions.**

***Index Terms*—motion/path planning, Reactive planning, Mobile robot, Robot dynamics, Robot kinematics, dynamic obstacles, elastic strip**
Type - Regular Paper

I. Introduction

MOTION planning is one of the most important and creative areas of robotics research [1]. The basic problem of motion planning consists of finding a collision-free navigation path for a robot in an environment among obstacles. This is a completely geometric problem that although appears to be simple, could be computationally very hard for robots with higher degrees of freedom. In 1969, Nilsson [2] used visibility graphs for mobile robot system that exhibited motion planning. A need for refining the motion planners was felt with introduction of mission level robot programming systems for assembly lines. Lozano-Perez and Wesley's work [3] extended Udupa's idea [4] of representing a robot by a point and designed a complete planner, which returns a collision free path whenever one exists and indicates that no such path exists otherwise. This led to the definition of configuration space (C-space) [5], which had a major influence on motion planning. The parameter space that encodes a robot's degrees of freedom and contains the robot or its end-effector as a point (a configuration) is called the configuration space. Each obstacle is represented as a prohibited configuration and hence, the complement of all the prohibited configurations forms the space in which the robot can navigate.

While these methods worked well for stationary obstacles, or obstacles with pre-defined or known trajectories [1], recent interest has inclined towards developing algorithms and planners that plan the robot's path in a very dynamic and unpredictable environment such as laboratory, factory floor, hospitals and in service / field robotics. Control theory enables a robot to use sensors to interact with the environment in real-time and use this knowledge to update its planned path.

In a dynamic environment, unforeseen obstacles may invalidate a previously planned path very often. Global planning algorithms like A* [6] take an order of minutes, ideally seconds [7], to determine a plan in complex environments, and hence, it would be impractical to use such algorithms for repeatedly generating plans to avoid collision with moving obstacles. Reactive approaches such as potential fields [8] can avoid dynamic obstacles but may get trapped in local minima. The reactive deforming path framework allows reactive obstacle avoidance behavior while preserving the global nature of the planned path (thereby avoiding local minima problems) and not suspending task execution, enabling robots to perform tasks in dynamic and unstructured environments.

The paper describes the development of the reactive planner and simulator for a 2D environment as well as implementation of the algorithm in a 3D simulator. The paper is arranged in following sections: Section II is a survey of past related work on reactive planning. Section III explains the high level system architecture of the reactive deforming path framework. Section IV presents the algorithm for this framework and explains the technique used for motion control with dynamic obstacle avoidance. Section V gives details of the implementation of simulators for the algorithm discussed in Section IV. Section VI presents the experimental results and capabilities of the algorithm and its implementation. Section VII describes the future vision for extending the current work. Section VIII concludes the paper by summarizing the approach and achievements.

II. Related Work

Various algorithms have been proposed for motion planning among dynamic obstacles. Some earlier algorithms make use of

Manuscript received May 2, 2009.
Anand TM is with the College of Computing, Georgia Institute of Technology, Atlanta, GA 30332 USA (phone: 404-775-2957; e-mail: Anand.TM@GaTech.edu).

978-1-4244-4775-6/09/$25.00 © 2009 IEEE.

known obstacle trajectories [9], while other algorithms utilize graphics hardware to plan the motion of rigid robots moving in R^3 [10]. The efforts to improve robot motion generation algorithms starts from the existence of complete but inefficient global planning algorithms or incomplete but efficient local approaches [11].

A. Path Modification

Path modification methods allow a specified path to move or deform based upon obstacle motion to ensure a collision free path. "Elastic Bands" provide fast path modification for moving, dynamic obstacles [12]. Elastic bands represent the path as an elastic deforming structure. This idea was extended to Elastic Strips [13] in which most of the computation is performed in the workspace rather than in the configuration space. The connectivity between the milestones is recomputed and the milestones are moved during execution to avoid the obstacles.

B. Replanning

Rather than altering the path itself, many approaches replan, or rebuild the connectivity of the free configuration space at each step. The D* deterministic planning algorithm makes use of previous solutions rather than starting from scratch [14], [15]. Additionally, there have been several algorithms that perform a similar task for higher dimensional configuration spaces [16] – [20]. In general, these methods wait until the roadmap or robot's path becomes invalidated by obstacle motion, and replan a new path. In a complex environment, this may not be the best solution, since it would be slower than required.

III. System Architecture

The framework in this approach is an extension of that in [12] with three levels (as shown in fig. 1):

1) Global path planner: A global plan is generated for the specified task, defined by the start and goal configurations of the robot, as well the definition of the environment.

2) Reactive deformation engine: The globally planned path is warped according to motion of the dynamic obstacles in real-time, to generate a collision-free path.

3) Control: A position and velocity based control system is used to make the robot navigate along the trajectory.

By using the second level, that is the deformation engine, the system does not have to incur the high cost of replanning by using the global planner again.

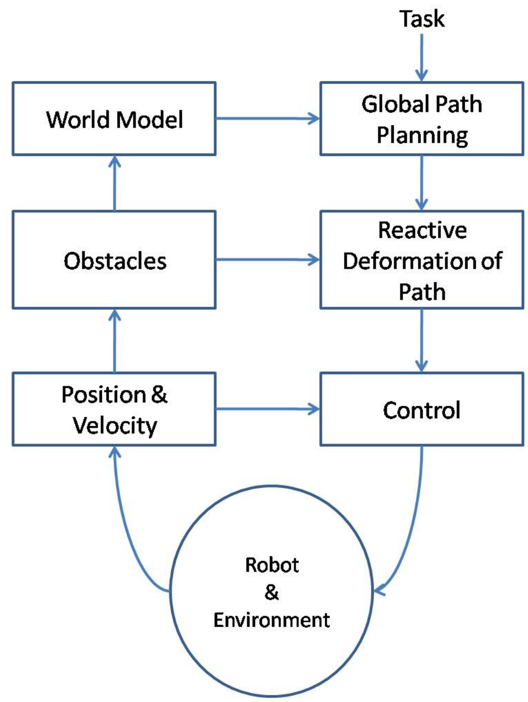

Fig. 1. Three level hierarchy of elastic path system

IV. The Algorithm

An already planned robot path (using A* for global planning), as shown in fig. 2, is modeled as if it is made of an elastic material as shown in fig. 3.

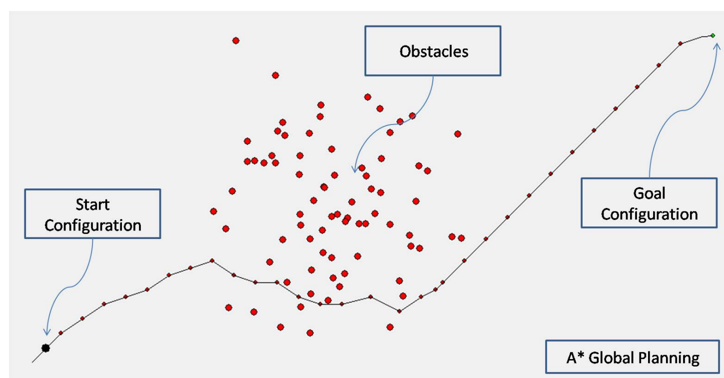

Fig. 2. Global Planning using A*

The path can be imagined as a rubber band connecting the start and goal position as in fig. 3(a) [13]. Each obstacle has a charge that is opposite to the charge on the path, and so, the obstacles exert a repulsive force on robot's trajectory, incrementally deforming it in the same way as an elastic strip would deform as in fig. 3(b). The path still remains an optimal trajectory. When the obstacle moves away, the path returns to its initial configuration.

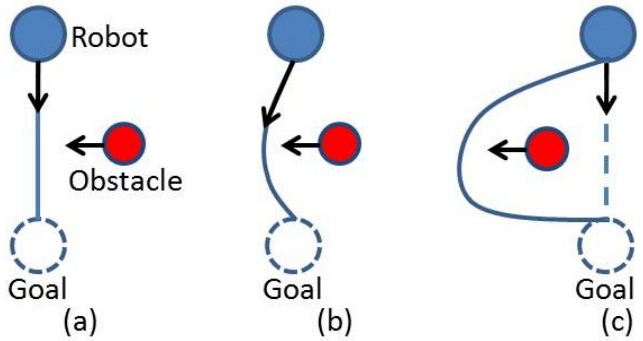

Fig. 3. Elastic Path becomes sub-optimal by deformation due to changes in environment

The obstacle can either move away by intersecting the path, and ultimately snapping the elastic strip by stretching it beyond its elastic limit, or just by moving away from the path. In the first case, the path regains its original shape due to snapping, while in the later case, it retracts due to a reduction in the repulsive forces between the path and the obstacle.

As shown in fig. 3(c), a straight-line trajectory is possible. This new path is the optimal plan. The original path would have to be invalidated by means of the notion of path elasticity, which states that if a path is stretched beyond its elastic limit, it shall snap in to a straight line. Global planner could also have been invoked to generate a new path, but since such obstacle behavior is likely to occur frequently in dynamic environments, we would like to be able to address it within the local planner itself. If the component of the repulsive force in the opposite direction of the internal force exceeds the magnitude of the external force, two adjacent configurations along the elastic strip start to separate. If this separation results in two disjoint components of the strip, the effect of repulsive forces at the separated configurations is suspended. The obstacle can then pass through the opening and internal forces will repair the strip, by joining the two disjoint pieces. Thus, the cost of calling the global planner is avoided by deforming the path when dynamic obstacles are moving around. Since it is the path, and not the robot, which is subjected to a charge, the local minima problem is avoided in this implementation.

A. *Free Space Representation*

The free space in the simulator was implemented by the equation in [13] to make sure that every time a deformed trajectory is generated, it should be in the free space. For this, the workspace is assumed to be filled with virtual spheres which represent free space. A virtual sphere with center ‘b’ represents the free space around point ‘b’ by the equation:

$$S(b) = \{a: ||b-a|| < d(b)\}$$

This equation defines the equation of a sphere S, with center ‘b’, and set of points ‘a’ such that the radius of the sphere is less than the minimum distance d(b) between the point ‘b’ and an obstacle near it.

A set of such spheres form the free space around each configuration of the robot along the path as shown in fig. 4. Every time the path is deformed, it needs to be checked that each link connecting two adjacent configurations is enclosed within the free space, that is, within the set of spheres around the two adjacent configurations. Such a path would guarantee to be free from obstacles.

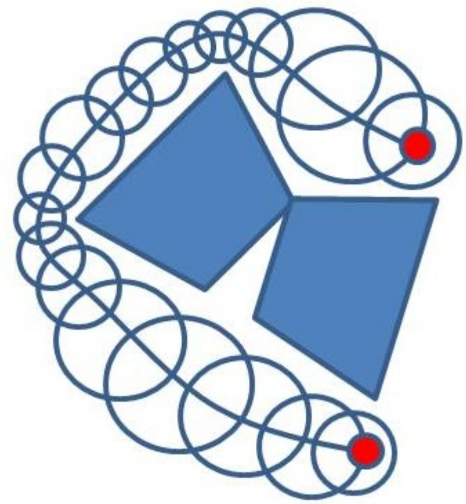

Fig. 4. Virtual Spheres representing free space around robot’s configurations along the elastic path

B. *The Forces*

The two forces applied to the path are: (i) internal contraction force and (ii) external repulsive force. The contraction force simulates tension in the planned path and is used to determine if the elastic limit of the path is reached and if it should snap. The path is repelled from the obstacles to generate the deformation. The deformation process stops when these two forces are in equilibrium. With a presence of new obstacles in the vicinity, the equilibrium is disturbed, and these forces make the path deform as necessary to avoid the obstacles.
The potential function used in the simulator is defined in [13] as:

$$V_{ext}(p) = \tfrac{1}{2}\, k_r(d_0 - d(p))^2, \text{ if } d(p) < d_0$$
$$= 0, \text{ otherwise}$$

where d_0 is the maximum distance around an obstacle up to which it can exert the repulsive force and k_r is called the repulsive gain. The repulsive force is $F_{ext}(p) = -\nabla V_{ext}(p)$ [13]

The internal forces can be thought of as the forces caused due to virtual springs connecting adjacent configurations along the path. Net force acting on a point p would then be defined as: $F(p) = -\nabla V_{ext}(p) - \nabla V_{int}(p)$ [13]
where $V_{int}(p)$ is the potential at point p due to the virtual springs

The force due to the virtual springs is computed in [13] by the equation:

$$F_{int} = \sum_{i=1 \text{ to } n-1} k_c((q_{i-1} - q_i / \| q_{i-1} - q_i \|) + (q_{i+1} - q_i / \| q_{i+1} - q_i \|))$$
where,

k_c is the contraction gain
q_i is the i^{th} configuration of the robot along the path represented by a vector (x,y,θ). In all there are 'n' configurations, q_0 to q_n.

The total internal contraction force is the normalized sum of the contraction force in each virtual spring connecting each pair of adjacent configurations, and represents the tension in the elastic path, making the path taut as shown in fig. 5.

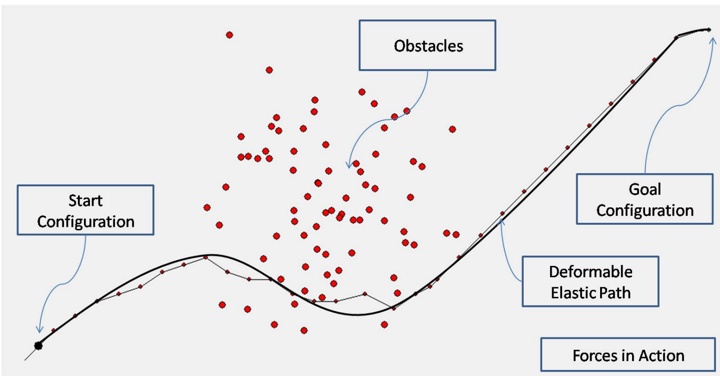

Fig. 5. Internal Contraction Force and External Repulsive Force together generate Elastic Path

Whenever obstacles approach closer to the trajectory, the free space reduces, and the trajectory deforms such that it can pass through the available free space as shown in fig. 6. New configurations are added to the path if necessary. In the opposite scenario, when obstacles move away, due to reduction of the repulsive forces, the trajectory regains its original shape by retracting as shown in fig. 7 or snapping as shown in fig. 8, and any extra configurations that may have been added are now removed.

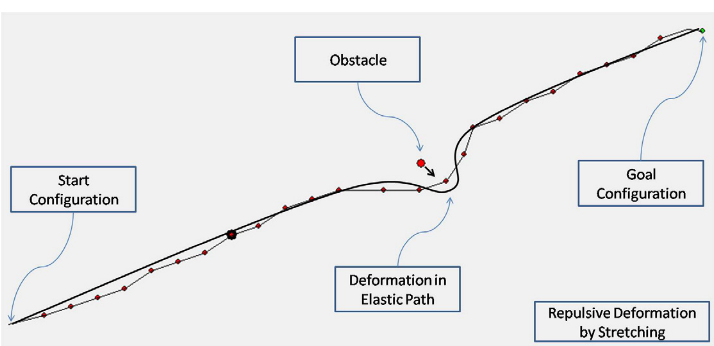

Fig. 6. Deformation of Elastic Path by Stretching

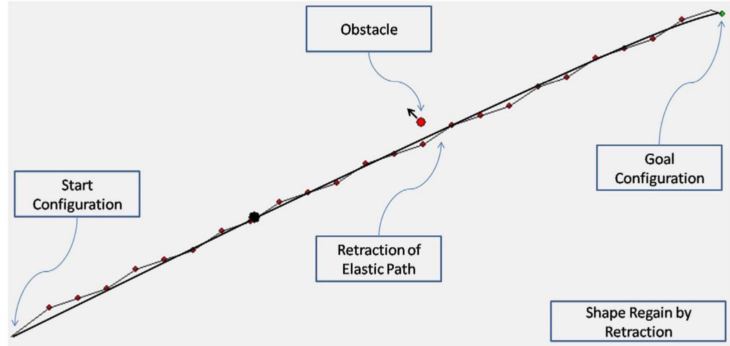

Fig. 7. Elastic Path regains shape by Retraction

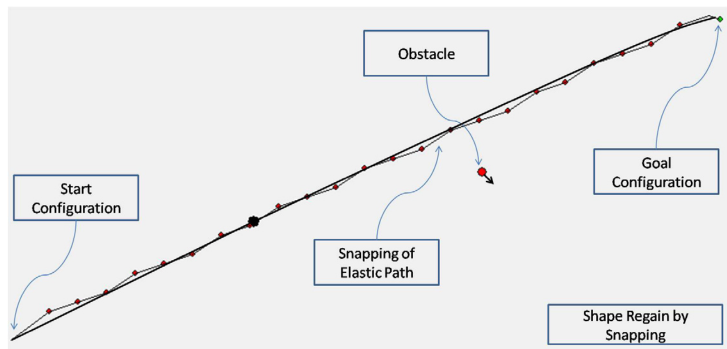

Fig. 8. Elastic Path regains shape by Snapping

C. *Dynamic Trajectory Generation*

As described in [13], the elastic strip represents a sequence of discrete configurations from start to goal. Due to the changes in the forces acting on each configuration along the path, the strip is deformed. Once the resultant force on each configuration is calculated, the configurations are displaced in the direction of this force, until the resultant force is zero, that is, equilibrium is established.

It is possible that the difference between current shape of the path and the shape resulting after establishing equilibrium due to the effect of nearby obstacles is large. In such cases, the robot may not be able to keep up with such large changes. This problem can be solved by limiting the trajectory's deformation. However, this is not very practical as it is infeasible to identify what a good limit would be in an unknown environment, thereby, causing the path to be invalidated more often than necessary leading to more replanning and less efficient execution. To avoid this, the solution proposed in [21] is to allow deformation without any limitations. This solution is also used in the simulator. The final trajectory is obtained by merging the originally planned path with the newly generated deformed path. In effect, this simulates snapping of the elastic path when it might have been stretched by more than its elastic limit. In extreme cases, either it is found that it is not possible to stretch any further and the configurations are returned to the originally planned set of configurations, or an effort is made by inserting additional configurations to stretch the path further.

V. Implementation

A 2D multi-threaded simulator was developed in C++ to implement the reactive deforming path planner algorithm described in the previous section. The simulator allows the user to set the start and goal positions of the robots, as well as, disperse obstacles in the environment. These obstacles follow an unknown trajectory, changing their direction and velocity of motion at random intervals of time. The virtual spheres are replaced by virtual circles in 2D. Once the environment is set, the planning phase begins in which first of all the global planner (A*) is invoked to generate a path. If it was known that none of the obstacles are stationary, we could as well generate the shortest path and then use the reactive planner to avoid dynamic obstacles. But there may be some stationary obstacles present in the environment and hence, we use A* to generate a plan. An efficient implementation of A* using priority queues was implemented to reduce the computation overhead as much as possible. After this initial phase, the multi-threaded nature of the simulator is apparent as one thread monitors the obstacles and their positions with respect to the trajectory while another thread controls the robot's motion along the trajectory. As soon as the sensor thread detects changes in the environment, it deforms the trajectory as necessary and as defined in the algorithm, and the control thread accordingly moves the robot along the new trajectory. The control thread is also responsible for making sure that the elastic path has not snapped, and if it has, retract the path back to its original form. As soon as new trajectory is computed, it is checked for validity. Validity is defined by each configuration being valid and in a free space.

The reactive deforming path algorithm was also implemented on the SNU Robotics Library - a simulation library for rigid multi-body dynamics. The implementation details are very similar to those for the previous simulator. There is some scope for improvement in the integration of the algorithm within the SNU simulation library.

VI. Experimental Results

The simulator was run with one mobile robot and varying number of dynamic obstacles with an unpredicted motion in an 800 x 1000 sq unit workspace. The update rate of the elastic path was varied between 10 and 100 Hz. The simulator was run several times to observe the performance, and it was found that the algorithm was successfully able to deform the path to avoid obstacles. In each run, the start and destination configurations of the robot were set. Next, the global planner would invoke A* to compute the global plan and the obstacles would interfere with this plan, making the reactive planner to deform the plan as necessary, while the robot is navigating along the latest path. The results and performance of the algorithm in the simulator can be seen at:

1) With fewer than 10 obstacles -
http://www.cc.gatech.edu/~anand/Videos/RDP/Anand_Elastic_Demo_Less.avi
2) With more than 20 obstacles -
http://www.cc.gatech.edu/~anand/Videos/RDP/Anand_Elastic_Demo_More.avi

VII. Future Work

Next step would focus on using the algorithm for navigation module of a humanoid / mobile robot in a dynamic environment. The challenge would be to make use of sensor data and generate deformations in real-time. Handling uncertainty in sensor readings would also be necessary while performing navigation in a dynamic environment. The current work can be extended to implement elastic roadmaps which can recover from an invalidation of the global motion.

VIII. Conclusion

The reactively deforming path planner implements an efficient elastic strip algorithm that allows obstacle avoidance behavior without suspending the task. Planning and execution methods are efficiently integrated in to the algorithm. A global planner computes the path between start and goal configurations. Incremental adjustments are made to the path in the dynamic environment while maintaining the global path. Thus, this framework provides the benefits of reactive planning without giving up on global planning.

References

[1] J.C. Latombe, *Robot Motion Planning*, Boston, MA: Kluwer Academic Pub, 1991.
[2] N.J. Nilsson, "A Mobile Automaton: An Application of Artificial Intelligence Techniques", Proc. 1st Int. Joint Conf. on Artificial Intelligence, Washington D.C., pp 509-520, 1969.
[3] J.C. Latombe, *Robot Motion Planning*, Boston, MA: Kluwer Academic Pub, 1991.
[4] N.J. Nilsson, "A Mobile Automaton: An Application of Artificial Intelligence Techniques", Proc. 1st Int. Joint Conf. on Artificial Intelligence, Washington D.C., pp 509-520, 1969.
[5] T. Lozano-Perez, and M.A. Wesley, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles", Comm. ACM, volume 22, issue 10 pp. 560-570, 1979.
[6] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, "A formal basis for the heuristic determination of minimum cost paths", IEEE Transactions of Systems Science and Cybernetics, SSC-4(2):100–107, July 1968.
[7] L.E. Kavraki, P. Svestka, J.C. Latombe, and M.H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces", IEEE Transactions on Robotics and Automation, 12(4):566-580, 1996.
[8] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", IEEE International Conference on Robotics and Automation, St. Louis, Missouri, pp. 500-505, March 25-28, 1990.
[9] T. Fraichard, "Dynamic trajectory planning with dynamic constraints: A "state-timespace" approach", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), July 1993.
[10] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Maocha, "Interactive motion planning using hardware accelerated computation of generalized voronoi diagrams", Proceedings of IEEE Conference of Robotics and Automation, 2000.

[11] O. Khatib, "Towards integrated planning and control", Proceedings of IFAC Symposium on Robot Control, volume 1, pp 305-313, 1994.
[12] S. Quinlan, and O. Khatib, "Elastic bands: Connecting path planning and control," Proc. of IEEE Conf. on Robotics and Automation, 1993.
[13] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," Int. Journal of Robotics Research, vol. 18, no. 6, pp. 1031–1052, 2002.
[14] A. Stentz, "The focussed D* algorithm for real-time replanning," Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1995.
[15] S. Koenig and M. Likhachev, "Improved fast replanning for robot navigation in unknown terrain," Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), May 2002.
[16] P. Leven and S. Hutchinson, "Toward real-time path planning in changing environments," Proceedings of the fourth International Workshop on the Algorithmic Foundations of Robotics (WAFR), 2000.
[17] T.-Y. Li and Y.-C. Shie, "An incremental approach to motion planning with roadmap management," Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), May 2002.
[18] M. Kallmann and M. Mataric, "Motion planning using dynamic roadmaps," Proceedings of the IEEE Conference on Robotics and Automation (ICRA), April 2004.
[19] L. Jaillet and T. Simeon, "A PRM-based motion planning for dynamically changing environments," Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2004.
[20] D. Ferguson, N. Kalra, and A. Stentz, "Replanning with RRTs," Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), May 2006.
[21] O. Brock, and O. Kathib, "Elastic Strips: A framework for integrated planning and execution", Proceedings of the International Symposium on Experimental Robotics, volume 250 of Lecture Notes in Control and Information Sciences, pp. 328-338, 1999.