Real-Time Replanning in High-Dimensional Configuration Spaces Using Sets of Homotopic Paths

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

Abstract

Real-time replanning is a prerequisite for motion execution in unpredictably changing environments. This paper presents a framework that allows real-time replanning in high-dimensional configuration spaces. Initially, a planning operation generates a path. The path is augmented by a set of paths homotopic to it. This set is represented implicitly by a volume of free space in the work space. Effectively, this corresponds to delaying part of the planning operation for the homotopic paths until motion execution. During execution reactive control algorithms are used to select a valid path from the set of homotopic paths, using proximity to the environment as a simple and effective heuristic and thereby significantly pruning the search in the configuration space. Experimental results are presented to validate the real-time performance of this framework in high-dimensional configuration spaces.

1 Introduction

Algorithms to generate robot motion have historically been divided into motion planning algorithms [12] and control algorithms [7]. While this distinction is vague and some algorithms cannot be exclusively attributed to either of these categories [6, 13], it is based on several fundamental properties of those methods. Motion planning algorithms are considered to generate a path that respects *global* motion constraints, whereas control algorithms only use *local* information to determine a motion command. Furthermore, motion planning algorithms compute the path *prior* to the motion execution, whereas control algorithms are determining motion commands based on feedback *during* motion execution. In addition, planning methods are generally *complete* or resolution complete, while control algorithms only optimize a local objective function and may therefore cause suboptimal behavior. This also holds for reactive methods, such as the potential field approach that exhibits *local minima* [9]. Generally, motion planning methods are of *high computational complexity*, whereas the computations for most control algorithms can be performed *many times per second.*

In unpredictably changing environments the applicability of motion planning algorithms is limited. Moving obstacles can invalidate a previously generated motion by obstructing the computed path. Should such a situation occur, a planning operation is computationally too complex to continuously recompute a new path in real time. Control algorithms, on the other hand, can react to sudden changes in the environment, but might fail to achieve the desired goal configuration. This paper presents the latest results on the elastic strip framework [2, 4] which integrates planning and control algorithms into a motion generation approach, allowing real-time replanning in high-dimensional configuration spaces.

2 Integration of Planning and Control

The computational complexity of motion planning methods is mostly determined by the cost of computing the free space. This can be attributed to two properties: Firstly, the search space has to be explored *globally*, and secondly, the size of the search space grows exponentially with the dimensionality of the configuration space. Attempts to render motion planning algorithms more efficient have taken these two properties as starting points.

The most efficient planning algorithms for high-dimensional configuration spaces avoid the computation of an explicit free space representation. The free space is computed and represented implicitly, by applying sampling techniques to the configuration space

[8]. This does not eliminate the dependency of the computational complexity on the dimensionality of the configuration space. The required number of samples to accurately represent the free space depends on the clutteredness of the configuration space, which generally increases with its dimensionality. Furthermore, these algorithms still have to explore the entire configuration space.

In an attempt to reduce the computational complexity of planning algorithms, methods methods were devised that use heuristics to locally explore the configuration space until a path is found [1, 5]. Those heuristics take advantage of information about the workspace to guide the local search in configuration space. The search relies on iterative or recursive methods to compute the boundary of free space around a particular configuration. For these methods the amount of computation will increase with the number of degrees of freedom of the robot because the extent of the free space region must be explored along every dimension of the configuration space.

The framework presented in this paper is applicable to replanning and therefore addresses a narrower problem than the approaches mentioned above. Motion planning methods reduce the complexity of motion planning by representing configuration space obstacles implicitly and by using workspace information to guide the exploration of configuration space. The elastic strip framework presented in the next section goes one step further: it delays the exploration of the configuration space until execution. This exploration can be performed very efficiently using reactive control algorithms and proximity information from the workspace.

The underlying idea is to represent a set of homotopic paths by the workspace volume a robot would sweep out along them. This can be done without considering the kinematic properties of the robot, i.e. without exploring the configuration space. The prerequisite is the existence of a valid path, called *candidate path*. Assume a planner has generated such a candidate path and it lies entirely in free space. The free space around the candidate path must contain the volume swept by the robot along paths homotopic to the candidate path itself. Those paths are represented implicitly by an approximation of the free space around the candidate path, rather than computing them explicitly. During execution this set of homotopic paths can be searched very efficiently for an alternate valid path, should the candidate path be invalidated by changes in the environment.

Using such a representation, planning and control can be integrated very tightly: The path generated by a motion planner is transformed into a more general representation by augmenting every path with an implicit representation of paths homotopic to it. Control algorithms can then be used during execution to efficiently search that space to find a valid path, should the original one become invalid due to changes in the environment. This is the underlying principle of the elastic strip framework, described in the next section.

3 Elastic Strips

The elastic strip framework allows real-time replanning by integrating planning and control. This integration relies on two important components: a simple representation of sets of homotopic paths and an efficient method of selecting a valid path from this set. Those components will be described in this section.

3.1 Sets of Homotopic Paths

An elastic strip represents the free space around an existing path using a set of overlapping balls of workspace. Using only one distance computation, the free space around a point p in the workspace can be computed. Such a ball of free space is called *bubble* [13]; it contains all points q that are close to p than the closest obstacle and is defined as

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

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

Figure 1: Protective hull around the Stanford Mobile Manipulator

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 $p \in \mathcal{R}$ 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.

Along the path $\mathcal{U}$ 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

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

where $V_{\mathcal{U}}^{\mathcal{R}}$ is the workspace volume of robot $\mathcal{R}$ swept along the path $\mathcal{U}$. The union $\mathcal{T}_{\mathcal{S}}^{\mathcal{R}}$ of protective hulls is called *elastic tunnel*. An example of an elastic tunnel is shown in Figure 2. Five configurations of the Stanford Mobile Manipulator along a given path are displayed. The overlapping bubbles of free space are shown as transparent spheres. The spherical obstacle in the middle is restricting the size of the bubbles. For clarity, other obstacles are not shown.

Figure 2: Elastic tunnel in the presence of an obstacle

Relating the elastic strip framework to the discussion in section 2, the initial path $\mathcal{U}$ corresponds to the candidate path, and the elastic tunnel implicitly represents a set of paths homotopic to it. Referring to figure 2, it is easy to imagine how the volume swept by the robot along the path $\mathcal{U}$ is contained within the elastic tunnel $\mathcal{T}$. It is also intuitive that the volume swept by the robot along a path $\mathcal{U}'$ that is homotopic to $\mathcal{U}$ and was obtained by a slight modification of $\mathcal{U}$, would also be contained within the elastic tunnel. Therefore, an elastic tunnel can be viewed as a representation of a set of paths homotopic to the candidate path. As the elastic tunnel can be computed very efficiently with very few distance computations, this constitutes the first component necessary for the integration of planning and control described in the previous section.

3.2 Selection of a Path

Given a candidate path and an implicit representation of a set of paths homotopic to it, an algorithm is required to efficiently select a path from the elastic tunnel, if the candidate path is invalidated by changes in the environment. For this algorithm to be as efficient as possible, a simple potential field-based control algorithm is used. Rather than exploring the entire configuration space, it maps proximity information from the environment into the configuration space, using the kinematics of the manipulator.

The robot is exposed to forces, acting in the workspace, that affect the selection of a path by incrementally modifying the candidate path. These forces are derived from three potential functions, the external, internal, and posture potential, V_{ext}, V_{int}, and V_{pos}, respectively. The external, repulsive potential V_{ext} is defined as a function of proximity to obstacles. Minimizing this potential effectively maximizes the clearance the path has to obstacles in the environment. For a point p on the robot the external potential is defined as follows:

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

where $d(p)$ is the distance from p to the closest obstacle, d_0 defines the region of influence around obstacles, and k_r is the repulsion gain. The force resulting from this potential that acts on point p is then given by:

$$F_p^{ext} = -\nabla V_{ext} = k_r (d_0 - d(p)) \frac{\vec{d}}{||\vec{d}||},$$

where $\vec{d}$ is the vector between p and the closest point on the obstacle. Intuitively, the repulsive potential pushes the robot away from obstacles, if it is inside their influence region. Using the external forces to select a new candidate path is different from a purely reactive approach, since the global properties of the path are maintained and local minima can be avoided.

External potential alone would suffice in most cases to select a new candidate path, as repulsive forces keep the robot in free space. If an obstacle deforming a path would recede, however, the path would never shorten. Virtual springs attached to control points on consecutive configurations of the robot along the elastic strip, as depicted in Figure 3, can achieve this effect. Let p_j^i be the position vector of the control point attached to the j-th joint of the robot in configuration q_i. The internal contraction force caused by the spring connecting to joint i to joint j is defined as:

$$F_{i,j}^{int} = k_c \left(\frac{d_j^{i-1}}{d_j^{i-1} + d_j^i} (p_j^{i+1} - p_j^{i-1}) - (p_j^i - p_j^{i-1}) \right),$$

where d_j^i is the distance $\|p_j^i - p_j^{i+1}\|$ in the initial, unmodified trajectory and k_c is a constant determining the contraction gain of the elastic strip.

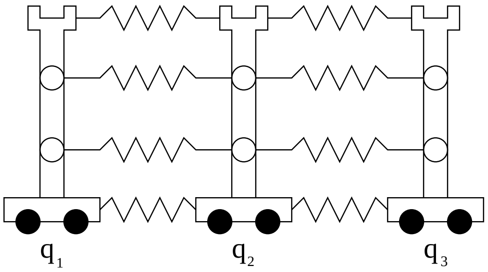

Figure 3: Principal structure of elastic strip

Using the external and internal forces, the elastic strip behaves like a rubber band. Obstacles cause it to deform, and as obstacles recede it assumes its previous shape. The forces are mapped to joint displacements using a dynamic model of the manipulator. This replaces configuration space exploration with a directed search, guided by workspace forces. This procedure is virtually independent of the dimensionality of the configuration space.

In case of redundant manipulators, a third potential is introduced. The potential $V_{posture}$ can be used to define a preferred posture for the robot in absence of other constraints. This allows, for example, to maintain the most stable or energy-conserving position when manipulating a heavy load. Based on the dynamically consistent decoupling of task execution and null space motion [3, 10], the motion resulting from these three potentials can be combined with the motion required for task execution. Motion in the null space of the task is used to avoid obstacles, shorten the trajectory, and achieve a desired posture without influencing task execution.

3.3 Updating the Sets

A roadmap captures the connectivity of the free space as a network of one-dimensional curves. As we are interested in dynamic environments that give rise to the need of replanning, it cannot be assumed that the roadmap will remain accurate. As a consequence for the replanning paradigm described in this paper, the sets of homotopic paths need to be updated in accordance with changes in free space connectivity. For example, an obstacle crossing a hallway from one side to another, introduces a second candidate path through the hallway in the roadmap, as it moves away from the wall and can now be passed on either side. After contact with the opposing wall is made, the two candidate paths are merged again.

The motion of obstacles in the environment can cause substantial changes in the connectivity of the free space. Ultimately, the reinvocation of a motion planner is needed to guarantee completeness. In most practical environments, however, the motion of obstacles translate to small changes of overall free space connectivity. Since the goal is to avoid a costly motion planning operation, local planning methods or heuristics can be used to update the roadmap. This subsection presents two of those local replanning operations that are integrated with the elastic strip framework. For further details please refer to [2].

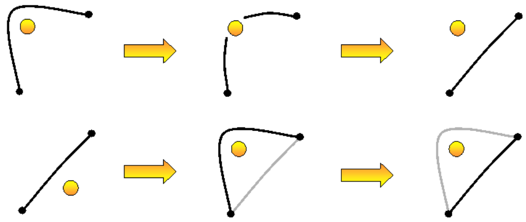

Figure 4: Local replanning

In populated environments, the most frequent type of change impacting the candidate path, is an obstacle crossing the path. As a result, it is deformed, as can be seen in the first image of Figure 4. By imposing constraints on the internal forces acting between two adjacent configurations along the elastic strip, the obstacle can be allowed to "pop through" the elastic strip, as shown in the top row of images of Figure 4. Internal forces then will reconnect the path, after the obstacle has passed. If internal forces cannot reconnect the path, a different local replanner or a global motion planner can be invoked [2].

Alternatively, the roadmap computed by the planner could always be maintained as a set of alternative routes. These routes can become invalid, if obstructed by obstacles. The obstacle will cause portions of the the original roadmap to change. If the obstruction is removed, however, the path in the original roadmap can again become a valid candidate path. This is illustrated in Figure 4, where a moving obstacle causes a path to be deformed. Further obstacle motion unblocks the original candidate path, which now is valid again and represents the shortest path between the two points.

4 Experimental Results

The elastic strip framework was tested in simulation and experimentally on various robotics [2]. Figure 5 shows two experiments with the Stanford Mobile Manipulator, a nine degree-of-freedom robot, consisting of a holonomic base and a six degree-of-freedom PUMA 560. The first series of images in Figure 5 shows how four PA-10 robots move into the path of the Stanford Mobile Manipulator. The first image of this sequence shows the candidate path, as computed by the planner. The subsequent pictures show how a new path is selected in real-time, as the motion of obstacles invalidates the original path. The replanning operation in the nine-dimensional configuration space is performed about 10 times per second on a 400 MHz Pentium PC. For simpler examples replanning rates of up to 100 Hz were achieved. The path is smoothended during execution. The other sequence of images shows a similar experiment for two Stanford Mobile Manipulators.

5 Conclusion

Conventionally, the link between planning and control is a path. To execute the path, it is first converted into a trajectory and then executed by control algorithms. Such a path can be represented as a set of via points, a parametric descriptions of joint positions as a function of time, or as a navigation function [11].

The elastic strip framework relies on the existence of a valid candidate path, previously obtained by a planner. Using the notion of elastic tunnel as an implicit representation of sets of homotopic paths as the connecting link, the elastic strip framework realizes a much tighter integration of planning and control than previous approaches. This integration allows to delay the exploration of the configuration space necessary for plan generation until execution, rendering replanning significantly more efficient. Experiments have proven the elastic strip framework to be an effective and efficient way of performing replanning in real-time, even in high-dimensional configuration spaces.

The replanning operation performed by the elastic strip framework is incomplete and may result in suboptimal paths. It can be augmented with other replanning primitives that allow to handle most practical situations. Ultimately, elastic strips cannot replace planning itself: if changes in the environment are substantial and the elastic strip framework cannot find a valid candidate path, the reinvocation of a complete planner becomes necessary.

Acknowledgments

The authors would like to thank Kong-Sok Chang, Bob Holmberg, and Diego Ruspini for their helpful insights and discussion in preparing this paper. The financial support of Boeing, Nomadic Technologies, General Motors, and NSF (grant IRI-9320017) is gratefully acknowledged.

References

[1] Jérôme Barraquand, Bruno Langlois, and Jean-Claude Latombe. Robot motion planning with many degrees of freedom and dynamic constraints. In *Robotics Research*, volume 5. Springer Verlag, 1989.

[2] Oliver Brock. *Generating Robot Motion: The Integration of Planning and Execution*. PhD thesis, Stanford Unviversity, 1999.

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

[4] Oliver Brock and Oussama Khatib. Elastic strips: A framwork for integrated planning and execution. In Peter Corke and James Trevelyan, editors, *Proc. of the Int. Symp. on Experimental Robotics*, volume 250 of *Lecture Notes in Control and Information Sciences*, pages 328–38. Spinger Verlag, 1999.

[5] John F. Canny and Ming C. Lin. An opportunistic global path planner. *Algorithmica*, 10:102–120, 1993.

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

[7] Gene F. Franklin, J. David Powell, and Michael L. Workman. *Digital Control of Dynamic Systems*. Addision-Wesley, 3rd edition, 1998.

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

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

[10] 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. of the Int. Conf. on Intelligent Robots and Systems*, volume 2, pages 546–53, 1996.

[11] D. E. Koditschek. Exact robot navigation by means of potential functions: Some topological considerations. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 1–6, 1987.

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

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

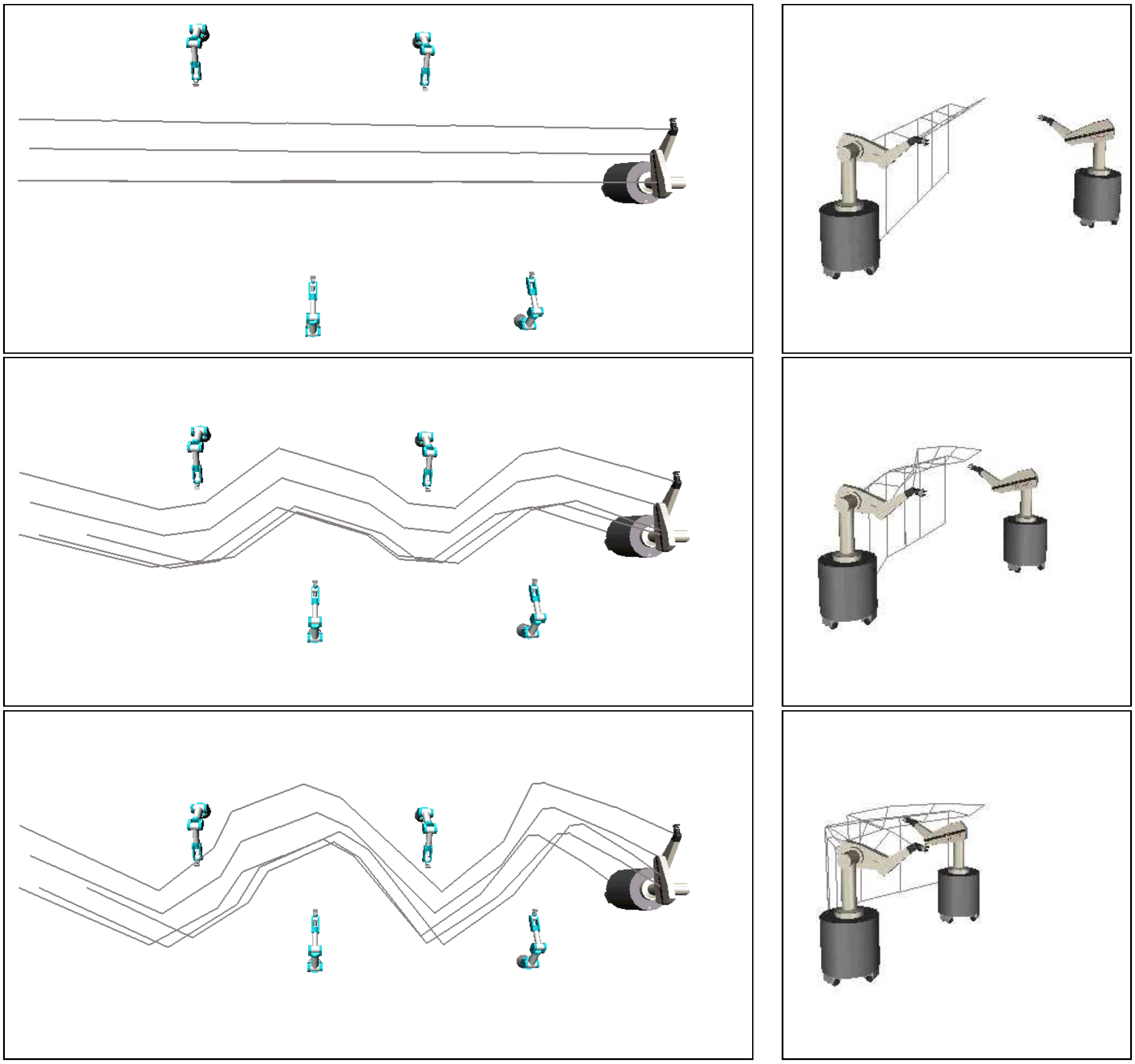

Figure 5: On the left: Four manipulator arms move into the path of the Stanford Mobile Manipulator; replanning using elastic strips is performed in real-time to maintain a valid path. The path is indicated by lines connecting points on conscutive configurations along the elastic strip. It is converted into a smooth trajectory during execution. On the right: A Stanford Mobile Manipulator moves into the path of another one. The path is updated in real-time to avoid a collision.