

Elastic Roadmaps: Globally Task-Consistent Motion for Autonomous Mobile Manipulation in Dynamic Environments

Yuandong Yang Oliver Brock
Department of Computer Science
University of Massachusetts Amherst

Abstract—The autonomous execution of manipulation tasks in unstructured, dynamic environments requires the consideration of various motion constraints. Any motion performed during the manipulation task has to satisfy constraints imposed by the task itself, but also has to consider kinematic and dynamic limitations of the manipulator, avoid unpredictably moving obstacles, and observe constraints imposed by the global connectivity of the workspace. Furthermore, the unpredictability of unstructured environments requires the continuous incorporation of feedback to reliably satisfy these constraints. We present a novel feedback motion planning approach, called elastic roadmap framework, capable of satisfying all of the motion constraints that arise in autonomous mobile manipulation and their respective feedback requirements. This framework is validated with simulation experiments using a mobile manipulation platform and a stationary manipulator.

I. INTRODUCTION

Autonomous robots are beginning to address real-world tasks in unstructured and dynamic environments. Today these robots predominantly perform tasks based on mobility. But the potential of augmenting autonomous mobility with manipulation skills is significant [4]. To achieve autonomous mobile manipulation, robots have to perform complex, task-oriented motion in unpredictably changing environments. To address this problem, we present a novel motion generation technique, called *elastic roadmap*. Elastic roadmaps generate robust and globally task-consistent motion in dynamic environments. To validate the approach, we present experiments with a mobile manipulator and a stationary robot arm.

In the context of autonomous mobile manipulation, motion is subject to numerous constraints. 1) The task imposes constraints that can be expressed as position or force constraints at the end-effector. 2) Kinematic and dynamic constraints, such as joint and actuation limitations, restrict the manipulator’s motion capabilities. 3) Postural constraints represent secondary task constraints that can improve some performance metric of the manipulator. Generally, they express a desired, subordinate behavior that should not interfere with task execution. 4) Reactive obstacle avoidance imposes motion constraints in response to changes in the environment. These constraints prevent collisions but cannot address global connectivity constraints. 5) Global motion planners consider constraints imposed by the global connectivity of free space. Only global consideration of connectivity can prevent the susceptibility to local minima.

The maintenance of these five types of motion constraints requires feedback to accommodate modeling error, uncertainty in sensing and execution, and the dynamic aspects of the

environment. The required frequency of feedback depends on the constraint and the application. In autonomous mobile manipulation, feedback requirements range from several hundred times per second for task constraints to about once per second for global motion constraints. A successful motion generation approach for autonomous mobile manipulation has to address all five types of motion constraints and in addition satisfy their respective feedback requirements.

We propose a novel framework for feedback motion planning [26] that combines the advantages of planning and control to address the motion requirements in autonomous mobile manipulation. From motion planning this framework takes the concept of a roadmap [22] to represent global connectivity information. Every milestone in the roadmap represents the attractor of a local potential function. By sequencing attractors in accordance with global connectivity information, we can generate a global navigation function. This navigation function permits the integration of control-based methods with global connectivity information through the use of task-level, multi-objective control. The resulting motion framework addresses all motion constraints that arise in autonomous mobile manipulation and their respective feedback requirements.

II. RELATED WORK

The literature concerning robot motion is extensive. We restrict the discussion to work that incorporates multiple motion constraints relevant to autonomous mobile manipulation.

a) *Control*: Control methods [15] specify motion behavior using potential functions defined in the robot’s state space. Each potential function represents a specific motion constraint or motion objective. Objectives can be combined into multi-objective motion by combining the respective potential functions. Control-based methods descend the gradient of such a combined potential function to achieve the desired motion behavior. The use of feedback during gradient descent renders motion generation robust to external disturbances at high rates of feedback, but also makes it susceptible to local minima in the combined potential landscape.

The operational space framework [23] achieves complex, multi-objective behavior [30] for manipulation tasks by combining potential functions with nullspace projections. These projections ensure that subordinate motion objectives do not interfere with superior ones. The resulting motion satisfies the motion constraints and their feedback requirements in autonomous mobile manipulation, with the exception of global connectivity constraints. Control-based methods are subject to

local minima and cannot guarantee the successful attainment of a particular motion objective.

b) Motion planning: Motion planners construct a global representation of the free configuration space to determine a valid motion. Among a large number of global motion planning techniques [12], [26], sampling-based motion planners [22], [24] currently represent the dominant planning paradigm. Global motion planners have been extended to specifically address constraints arising in manipulation [32] or in the context of dynamic environments [16], [17], [20], [21], [27], [34], [35]. However, these methods are currently not computationally efficient enough to incorporate feedback at rates required to satisfy task constraints in the context of autonomous mobile manipulation, such as force control, for example. The acceleration of global motion planning techniques has thus been an active area of research [7]. One of the promising directions in this area is the inclusion of workspace information to guide sampling [25], [33], [38]. The elastic roadmap approach presented in this paper will make extensive use of workspace information to improve the feedback rates for the computation of global connectivity information.

c) Feedback motion planning: Feedback motion planners construct a local minima-free potential function based on global information [26]. This can be achieved with navigation functions [29], numerical navigation functions [2], harmonic potential functions [14], or by composing a series of local potential functions based on global information [8], [10], [11], [13], [36]. Once such a global potential function has been computed, these methods are able to satisfy all of the motion constraints mentioned above. However, changes in the environment or task will invalidate a computed global potential function. Since the computation of this function for robots with many degrees of freedom is computationally complex, this type of approach cannot satisfy the feedback requirements of autonomous mobile manipulation. The proposed elastic roadmap framework overcomes this limitation.

d) Integration of local and global methods: The elastic band framework [28] assumes that a global motion planner has determined a motion that satisfies the task-requirements. Local methods subsequently modify this motion incrementally in response to feedback from the environment. The elastic strip framework [6] extends the elastic band framework and has been used to demonstrate global, task-consistent obstacle avoidance and posture behavior on a mobile manipulator [6]. Both of these approaches combine the advantages of control-based approaches with a predetermined global motion. However, neither of these methods can recover from an invalidation of the global motion.

Decomposition-based motion planning [5] also combines global and local methods. Initially, global connectivity information in the workspace is captured in a navigation function for the manipulator's end-effector. This navigation function is combined with control-based methods to generate the robot's motion, satisfying all considered motion constraints and their feedback requirements. However, the fact that the navigation function is computed in the workspace and only for the end-

effector limits the applicability of the approach.

The proposed elastic roadmap framework can be seen as a natural progression in this line of research. It combines local and global methods into a reactive approach to feedback motion planning.

III. ELASTIC ROADMAP FRAMEWORK

The elastic roadmap framework is a robust and efficient framework for feedback motion planning in dynamic environments. It addresses all motion constraints for autonomous mobile manipulation and their respective feedback requirements. To accomplish this, the elastic roadmap framework relies on the following main ideas:

a) Elastic roadmap: Conventional roadmaps capture global connectivity information in graphs consisting of collision-free vertices (milestones) and edges. Once added to the roadmap, neither milestones nor edges are changed. In contrast, an elastic roadmap moves its milestones and updates their connectivity to adapt to changes in the environment. The modification of milestones is performed in a task-consistent manner, relying on methods from task-level control. The roadmap thus always represents task-consistent motions. The visual effect resulting from the continuous modification of the roadmap gives rise to the name *elastic* roadmap. An elastic roadmap consists of a set of milestones $M = \{m_1, \dots, m_n\}$ and an ordered binary relation $C : M \times M \rightarrow \{0, 1\}$. The relation $C(m_i, m_j)$ replaces the notion of edges; it holds for two milestones m_i, m_j if a feedback controller is able to move the robot from m_i to m_j .

b) Navigation function: The relation $C(m_i, m_j)$ holds for two milestones if a feedback controller is able to move the robot from m_i to m_j . In this view of a roadmap, every milestone m_i is associated with a local potential function for which it is the attractor. Milestone m_j is connected to m_i if m_j is within the region of attraction of the potential function associated with m_i . An elastic roadmap thus defines a hybrid system of potential functions. Given a particular goal state, the connectivity of the roadmap determines how a hybrid system can compose a set of local potential functions into an approximate, global navigation function.

c) Using workspace information to determine milestones: To improve the efficiency of global motion planning, many researchers have directed their efforts towards making the construction of roadmaps more efficient. Notable advances in this area have come from the consideration of workspace information [25], [33], [37], [38]. The elastic roadmap framework determines the entire roadmap directly in workspace, avoiding the computationally costly construction of a configuration space roadmap. This step is the key to satisfying the feedback requirements for global motion constraints.

The proposed framework's efficiency in maintaining global connectivity information comes at the price of completeness. In fact, the framework consciously trades completeness for computational efficiency. We believe that within the context of a particular application, such as autonomous mobile manipulation, this tradeoff is justified. The evaluation of

such an incomplete, application-specific method then has to be performed in the context of application-specific planning problems. To validate the proposed approach, we present three experimental scenarios in Section V. These scenarios can be viewed as characteristic of a broad class of tasks in autonomous mobile manipulation. Our results obtained in these scenarios demonstrate that the degree of completeness achieved by the proposed framework remains adequate for our application. At the same time, they demonstrate that the framework is able to satisfy all motion constraints and their feedback requirements. A more detailed discussion of completeness is given in Section IV-H.

IV. AN ELASTIC ROADMAP IMPLEMENTATION

We now describe a specific implementation of the elastic roadmap framework proposed in Section III. This implementation should be viewed as a proof of concept. The specific algorithmic components of the implementation will be subject of future investigations.

A. Task-level control

Task-level control [23] is a convenient and powerful method of generating multi-objective behavior for robotic systems. Rather than specifying joint trajectories, this framework permits direct control of the manipulator’s end-effectors, greatly facilitating programming for kinematically redundant robots. Task-level control also permits the task-consistent execution of subordinate behaviors, exploiting nullspace projections. Given an end-effector task, expressed as a force F_{task} acting on the end-effector, and given an arbitrary subordinate behavior, expressed as a vector of joint torques Γ_0 , we can determine the torque Γ to achieve task and subordinate behavior as follows:

$$\Gamma = J_{\text{task}}(q) F_{\text{task}}^T + N_{\text{task}}^T(q) \Gamma_0, \quad (1)$$

where N_{task}^T represents a projection into the nullspace of the end-effector Jacobian J_{task} . This projection ensures that the subordinate behavior will not alter task behavior, i.e., it will result in task-consistent motion.

This principle of nullspace projections can be extended to cascade an arbitrary number of hierarchical behaviors [30]. If behavior i results in torque Γ_i , the torque

$$\Gamma = \Gamma_1 + N_1^T(q) (\Gamma_1 + N_2^T(q) (\Gamma_3 + N_3^T(q) (\dots))) \quad (2)$$

combines these behaviors in such a way that behavior i does not affect behavior j if $i > j$. In equation 2, N_i^T is the nullspace projection associated with the task Jacobian of behavior i . Here, we adopt the more compact notation of the control basis [19] to describe such cascaded nullspaces. We associate a control primitive ϕ_i with each torque Γ_i . If a control primitive ϕ_i is executed in the nullspace of the control primitive ϕ_j , we say that ϕ_i is performed subject to ϕ_j , written as $\phi_i \triangleleft \phi_j$. We can now rewrite equation 2 as

$$\dots \triangleleft \phi_3 \triangleleft \phi_2 \triangleleft \phi_1. \quad (3)$$

This task-level framework with nullspace projections serves as the underlying control scheme for the elastic roadmap framework. Torques computed using this framework are applied to the robot to generate its motion.

B. Creating milestones

A roadmap has to capture the connectivity of free configuration space to allow the solution of motion queries. The adequate placement of milestones critically determines the quality of a roadmap. For sampling-based multi-query motion planning, prior work shows that the adequacy of milestones is largely determined by their visibility properties, i.e., the amount of free configuration space “visible” to them [18]. Milestones with large visibility provide better coverage of configuration space. This has motivated heuristics for placing samples in configuration space [31]. Another sampling heuristic attempts to place milestones close to the boundary of configuration space obstacles [1], following the intuition that solution paths circumnavigate these obstacles. When sampling in configuration space, however, visibility properties and obstacle boundaries are unknown, making it difficult to a priori select milestones with favorable properties.

The elastic roadmap framework generates milestones based on workspace information about obstacles. More specifically, milestones are generated for configurations in which the robot is in proximity to workspace obstacles. For such configurations, the corresponding point in configuration space must be close to the boundary of the corresponding configuration space obstacle. Our technique for generating milestones is therefore a special case of other obstacle boundary sampling heuristics [1].

It is not sufficient for the elastic roadmap framework to generate milestones that are collision free. Adequate milestones also have to satisfy task constraints. These task constraints are specified in terms of the robot’s end-effector and may limit its position or orientation. This is illustrated in Figure 1, where the end-effector is constrained by the task to remain positioned on the horizontal line, effectively constraining two positional degrees of freedom.

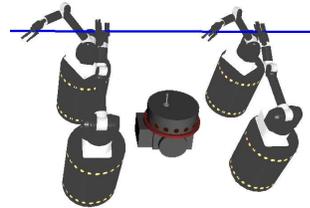


Fig. 1. Four task-consistent milestones associated with a simple obstacle. The task consists of moving the end-effector along the line.

To describe milestone generation in the context of the elastic roadmap framework, we distinguish between two kinds of tasks: end-effector placement (the end-effector position is only constrained at the final configuration) and position-constrained end-effector motion (at least one of the translational degrees of freedom of the end-effector is constrained by the task). For end-effector placement tasks, we reduce the notion of configuration space coverage to workspace reachability for the

end-effector: if an elastic roadmap allows us to reach every workspace location with the robot’s end-effector, we have achieved workspace coverage. The motion for the remaining links of the robot will be generated by powerful task-level controllers [30].

Our ability to place the end-effector in any position and orientation is limited by obstacles. Our milestone creation method for placement tasks thus considers possible end-effector placements in the proximity to the surface obstacles. We need to determine a roadmap that permits the motion of the end-effector from any feature on the obstacle to any other feature on the same obstacle. Since the motion between obstacles is by definition obstacle free, a collection of such roadmaps for all obstacles would allow us to achieve workspace coverage for end-effector placement tasks.

We now describe a naive approach to the creation of milestones for end-effector placement tasks based on workspace obstacles. This approach serves as a proof-of-concept for the elastic roadmap framework. More sophisticated milestone creation methods will be the subject of future investigations.

Workspace obstacles are decomposed into convex regions; these regions are then approximated by bounding boxes. We select each of the corners of the bounding box as well as the centers of the edges as obstacle features. These features are chosen so that the end-effector can move freely between adjacent features. To create milestones associated with these obstacle features, we pick a nearby configuration and drag the end-effector towards the feature. The controller used to achieve this is given by:

$$\phi_{\text{posture}} \triangleleft \phi_{\text{avoidance}} \triangleleft \phi_{\text{task}} \triangleleft \phi_{\text{feature}} \triangleleft \phi_{\text{collision}} \triangleleft \phi_{\text{kinematic}}, \quad (4)$$

where ϕ_{posture} describes a posture potential for kinematic conditioning of the robot, $\phi_{\text{avoidance}}$ performs reactive obstacle avoidance, ϕ_{task} describes a task potential, ϕ_{feature} is the potential that maintains proximity between the robot and the obstacle feature, $\phi_{\text{collision}}$ is able to prevent imminent collisions, and $\phi_{\text{kinematic}}$ prevents the manipulator from reaching its joint limits. The resulting milestones are added to the roadmap. If the obstacle feature can be reached by the end-effector without collisions, the milestone is considered *valid*.

The resulting set of milestones place the end-effector on obstacle features. Milestones associated with adjacent features are likely to be very similar. Furthermore, the motion of the end-effector between two different features is unobstructed. For most of the adjacent features it will therefore be possible to employ a task-level controller to generate the motion from one milestone to an adjacent milestone. This intuitive argument will later be confirmed by our experimental results. It is apparent that the selection of milestones plays a critical role in the performance of the proposed framework and warrants significant further investigation.

Milestones for position-constrained end-effector motion are created in a similar fashion. Instead of dragging the end-effector towards the feature, we drag the closest point on the robot towards the feature, while maintaining the task constraints. In equation 4, the order of ϕ_{feature} and ϕ_{task} is changed.

If the resulting milestone satisfies the task constraints, it is considered *task-consistent*. Four task-consistent milestones for task-constrained end-effector motion are shown in Figure 1.

C. Maintaining milestones

Milestones continue to be controlled by their respective controllers (equation 4). This permits them to react to changes in the environment. In particular, when the obstacle associated with the feature moves, the milestone will move with it. If these changes cause the milestone to violate task constraints (for task-constrained end-effector motion), the milestone changes its status from *task-consistent* to *valid*. If it violates collision avoidance constraints or kinematic constraints, it is labeled as *invalid*.

The computational complexity of milestone creation and milestone maintenance is $O(nm)$ [9], where n is the number of degrees of freedom of the robot and m is the number of milestones generated. This implies that the computational complexity of milestone maintenance is proportional to the geometric complexity of the workspace obstacles.

D. Determining connectivity among milestones

To complete the computation of an elastic roadmap, we need to determine the connectivity relation $C(m_i, m_j)$, where m_i, m_j are milestones.

In our current implementation, we employ workspace visibility as a criterion for the connectivity of two milestones. If two milestones m_i and m_j are mutually “visible,” we add $C(m_i, m_j)$ to the connectivity relation of the elastic roadmap. Visibility between milestones is determined by evaluating if designated handle points (origins of the coordinate frames attached to each link) [37] on the respective milestones can be connected by straight, collision-free line segments. This criterion is computationally efficient and relatively accurate, i.e., if the criterion determines that two milestones are visible, a valid trajectory can be determined using task-level, multi-objective control. In Section IV-G, we discuss how the planning method recovers from failure, should the visibility criterion erroneously label two milestones as connected.

While this approximation of connectivity compromises completeness (see Section IV-H), it allows us to leverage the power of task-level, reactive control as a local planner among milestones. As we will see in our experimental validation of the framework (see Section V), this simple heuristic for the connectivity relation can solve challenging motion planning problems in dynamic environments in real time. In future work, we will investigate more sophisticated and complete workspace criteria to determine the connectivity of milestones in the elastic roadmap.

E. Extracting a navigation function from the elastic roadmap

An elastic roadmap does not represent explicit configuration space trajectories. Instead, it continuously maintains a graph, consisting of task-consistent milestones (vertices) and hypothesis about the connectivity of these milestones (edges). Similarly to other roadmap-based motion planning

approaches, graph search algorithms can be used to extract a path in this graph that connects the initial and the final configuration of a motion planning problem (see Section IV-F). In the elastic roadmap approach, this path represents a sequence m_1, m_2, \dots, m_n of milestones. The motion between two milestones m_i and m_{i+1} can be generated using the following task-consistent, multi-objective controller:

$$\phi_{\text{posture}} \triangleleft \phi_{\text{avoidance}} \triangleleft \phi_{\text{global}} \triangleleft \phi_{\text{task}} \triangleleft \phi_{\text{collision}} \triangleleft \phi_{\text{kinematic}} \cdot \quad (5)$$

Compared to equation 4, we have removed the control primitive ϕ_{feature} and added the control primitive ϕ_{global} , which is responsible for the global motion towards the next milestone.

We view the extracted sequence of milestones as a hybrid system. Controllers are used to generate the motion from m_i to m_{i+1} until the robot has approached milestone m_{i+1} . The hybrid system then discretely switches to the controller that moves the robot from m_{i+1} to m_{i+2} , until the goal milestone m_n is reached. Note that throughout the entire motion all milestones as well as the motion between them remain consistent with all motion constraints. The hybrid system represents an approximate, local-minima free navigation function for the given motion problem. The navigation function is composed from simple, local potential functions by considering the global connectivity information captured in the roadmap [8], [10], [11], [13], [29], [36].

F. Updating the elastic roadmap

The update of the elastic roadmap consists of three parts: milestone maintenance, connectivity update, and path extraction. Milestone maintenance (Section IV-C) is performed continuously at high frequencies to ensure the maintenance of task constraints. The resulting motion of the milestones may invalidate the connectivity information represented in the roadmap. We now describe how the connectivity is updated and how a path is extracted. These parts of the algorithm are also performed continuously, but at a low frequency that is sufficient to satisfy the feedback requirements of global motion constraints.

Given a roadmap with n milestones, there are potentially $O(n^2)$ visibility tests to perform during a connectivity update. Due to the small computational cost of a connectivity check and the low update frequency associated with global motion constraints, this is feasible in practice. In the experiments presented in Section V, the computational cost of connectivity checks was dwarfed by the cost of milestone maintenance. In much larger environments, the cost of $O(n^2)$ visibility tests can be reduced to $O(n)$ (assuming a uniform spatial distribution of milestones) by restricting the adjacency of milestones based on spatial proximity.

Graph algorithms, such as A*, can be used to extract a path from a roadmap [3]. To select paths with certain favorable properties, the relationship between two connected milestones is labeled with the associated traversal cost. During the execution of a task-constrained motion, only task-consistent milestones are considered during path extraction. For end-effector placement tasks, it suffices for milestones to be valid,

i.e., collision-free. Invalid milestones are excluded from the connectivity update as well as from the path extraction but continue to be modified based on their controllers.

G. Recovering from failure

We distinguish three failure modes. First, a motion can fail because the task-level controller is unable to find a motion between two milestones that are connected in the roadmap. Second, changes in the environment may force the robot to give up the task constraints, leading to task failure. For both of these failures we describe recovery strategies below. The inability to find a valid path, even after these recovery strategies have been applied, constitutes the third failure mode. This last failure has to be attributed to the incompleteness of our method (see Section IV-H).

The first failure mode occurs when the robot is moving between two milestones. If a robot fails to make progress without having reached the next milestone, the connection is labeled as invalid. The continuous path extraction process will automatically obtain a new path and the robot will start moving along this path. Currently, the two milestones remain unconnectable, but one could reconsider the connection after the environment has changed.

The second failure mode occurs during task-constrained motion. If the robot or any of the milestones in the current path change their label from task-consistent to valid or invalid, a new path has to be computed. This occurs automatically during path extraction. If a new path with task-consistent milestones can be found, it is executed. If no such path can be found, path extraction considers the shortest possible recovery path through valid milestones to a task-consistent milestone. To follow the recovery path, we use the following controller:

$$\phi_{\text{posture}} \triangleleft \phi_{\text{avoidance}} \triangleleft \phi_{\text{task}} \triangleleft \phi_{\text{global}} \triangleleft \phi_{\text{collision}} \triangleleft \phi_{\text{kinematic}} \cdot \quad (6)$$

Once the robot reaches a task-consistent milestone, the original controller (Equation 5) is used to resume task behavior.

H. Completeness

The elastic roadmap framework is an approach to feedback motion planning that is incomplete by design. It does not possess any of the completeness properties of sampling-based planners. Nevertheless, it is tempting to compare the elastic roadmap framework to sampling-based planning methods. Such a comparison, however, is difficult to make, since the planning problems addressed by these two methods are fundamentally different. The elastic roadmap framework explicitly addresses task constraints and feedback requirements of a specific application and permits the execution of motion in dynamic environments under these constraints. It is able to do so precisely because it sacrifices completeness. To our knowledge, no sampling-based method is able to either consider task constraints in the generality proposed here or to address explicit timing constraints.

A characterization of completeness of the elastic roadmap would be most useful in the context of workspace properties. This has been proposed in [5], where a minimum clearance

about the solution path is required for the method to be complete. Such a notion of completeness could be established for the elastic roadmap framework by showing that the selected milestones can be reached from the entire configuration space, excluding areas that do not provide sufficient clearance for the robot. Furthermore, such a notion would require workspace connection strategies with provable performance to ensure that when the regions of attraction of milestones overlap, task-level planning will find a connection trajectory. We will investigate such notions of workspace completeness in our future work.

V. EXPERIMENTAL EVALUATION

We demonstrate the performance of the elastic roadmap framework by performing simulation experiments. An adequate experiment for the evaluation of the elastic roadmap approach should exhibit the following characteristics: 1) The robot has to maintain an end-effector task throughout the entire motion. 2) The motion should be subject to kinematic or dynamic constraints. 3) A solution has to depend on global connectivity information. 4) The environment should be dynamic and the motion of obstacles should interfere with the motion. 5) The robot should be kinematically redundant to permit the execution of multi-objective behavior. Following these criteria, we have devised three experiments for the two simulated robots shown in Figure 2.



Fig. 2. Left: The UMass mobile manipulator with ten degrees of freedom. Middle: Model of the real platform. Right: Stationary robot with twelve degrees of freedom.

In the first experiment, the mobile manipulation platform has to move its end-effector along a horizontal, linear path, indicated by the horizontal line in images 1a) and 1e) of Figure 3. During task execution, several obstacles move into the platform’s path. The direction of motion for the obstacles is indicated by the arrows. The sequence of images 1b)–1d) in Figure 3 illustrates how the elastic roadmap maintains task-consistent workspace connectivity and repeatedly generates new sequences of milestones in response to changes in the environment. By following the approximate navigation function constructed from the roadmap, the robot moves to the goal configuration in a task-consistent manner, i.e., by restricting the end-effector position to the line. Control primitives prevent collisions and keep the robot away from its joint limits. All computations are performed during the simulation and the motion is generated in real time. The pictures represent snapshots of the ongoing execution.

Note that the narrow passages between the moving obstacles render this motion problem challenging for motion planners

operating in configuration space. Furthermore, these planners also have difficulties to generate task-consistent motion, as the computation of the manifold of task-consistent configurations is computationally complex.

In a second experiment with the mobile manipulator, we demonstrate two additional capabilities of the framework. First, we show that the framework is capable of generating task-consistent motion even for force-controlled tasks, i.e., motion in contact with the environment. This stands in contrast with other approaches to motion generation [6], [28], which require the entire manipulator to move in free space. In the elastic roadmap framework, the only requirement is that motion continuously satisfies all motion constraints. These constraints may include force constraints. Second, we demonstrate that the elastic roadmap framework is able to automatically recover from the violation of task-constraints.

In this experiment, the end-effector of the manipulator tracks the unknown motion of an object based on force control. This can initially be achieved using task-level control alone, since the goal configuration is connected to the robot. As this direct connection is invalidated by moving obstacles forming a boxed canyon around the manipulator, the manipulator is unable to follow the moving object and has to violate task constraints by letting go of the object. Image 2b) in Figure 3 illustrates how the elastic roadmap framework then determines a path through valid but not task-consistent milestones to re-attain the task constraints in image 2c).

In a third experiment, we demonstrate the effectiveness of the elastic roadmap framework for a stationary twelve degree-of-freedom manipulator. The task consists of moving the end-effector to a goal location, while maintaining its orientation (task constraint). This motion is performed in an environment that contains a truss moving from right to left, as indicated by the arrows in image 3a) in Figure 3. The sequence of images 3a)–3d) illustrates how the robot reaches its goal location, while avoiding the moving truss and maintaining its end-effector orientation. As the truss keeps moving, it forces the manipulator to deviate from its goal location, repeatedly triggering a replanning operation that results in a repeated motion, similar to the one shown in images 3a)–3d).

All simulations were performed on a PentiumIV 3.2GHz PC with 1GB RAM and a 64MB DDR Radeon 300 graphics card. The computations associated with the maintenance of the elastic roadmap and the extraction of a path can be performed at a frequency of approximately 5-10Hz, thereby satisfying the feedback requirements for global motion in the context of autonomous mobile manipulation.

These experiments represent realistic autonomous manipulation scenarios. The elastic roadmap framework is able to maintain all required motion constraints while satisfying their respective feedback requirements. These experiments therefore demonstrate the effectiveness of the elastic roadmap framework for the generation of constraint-consistent motion for autonomous mobile manipulation in dynamic environments.

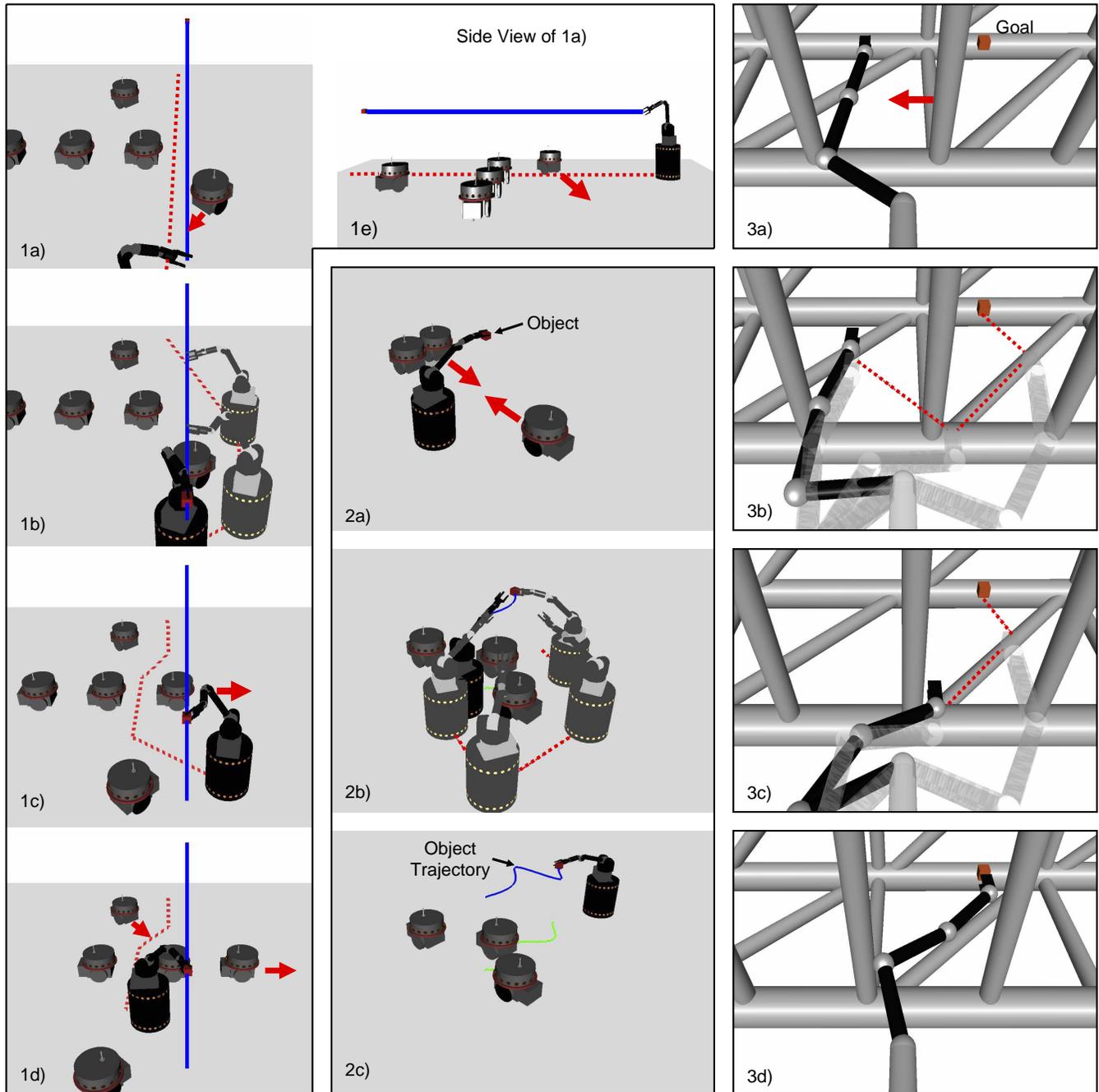


Fig. 3. Three experiments to validate the elastic roadmap approach. In all experiments, lighter versions of the robot represent milestones of the roadmap that are part of the current solution paths. The connectivity of these milestones is indicated by a dashed line. The darker robots represent the actual robot in motion. The direction of motion of obstacles is indicated by arrows. *Experiment 1:* Images 1a) and 1e) show two perspectives of the same scene. The robot follows the line with its end-effector, while moving obstacles invalidate the solution. The elastic roadmap framework repeatedly generates global, task-consistent motion plans until the robot reaches the goal. *Experiment 2:* The task consists of following an object moving on an unknown trajectory. The following task is achieved based on force control. Moving obstacles force the robot to suspend the force control task, losing contact with the object. The elastic roadmap framework computes a path to re-attain the force control task, shown in image 2c). This image also shows the trajectory taken by the object and its projection onto the floor. *Experiment 3:* A stationary robot, operating under a moving truss, reaches for a goal location while maintaining a constant orientation with its end-effector. The sequence of images shows how the goal can be reached. Continued motion by the truss will repeatedly force the robot to move away from the goal location to avoid collision. The elastic roadmap framework repeatedly generates motions such as those shown in images 3a)–3c) to re-attain the goal location.

VI. CONCLUSION

Motion in the context of autonomous mobile manipulation is subject to numerous constraints. These constraints are imposed by the task, by kinematic and dynamic limitations of the robot, by moving obstacles in the environment, by global the global connectivity of the workspace, and by subordinate behaviors, such as posture control. Existing approaches to motion generation for autonomous mobile manipulation either fail to address all motion constraints simultaneously or do not meet the respective feedback requirements. We have presented the elastic roadmap framework as a new approach to feedback motion planning. This framework satisfies all of the aforementioned constraints and their feedback requirements. Furthermore, the framework is capable of generating constraint-consistent motion in dynamic environments in real time. To achieve the computational efficiency, the elastic roadmap framework makes several approximations that cause it to lose the provable completeness many other planning techniques possess. However, our experimental results indicate that the elastic roadmap framework is able to solve challenging motion generation problems for autonomous mobile manipulators and stationary manipulator arms.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation (NSF) under grants CNS-0454074, IIS-0545934, and MIT/NASA cooperative agreement NNJ05HB61A. The authors would like to thank the anonymous reviewers and Nancy Amato for their helpful comments.

REFERENCES

- [1] N. Amato, B. Bayazit, L. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Robotics: The Algorithmic Perspective*. AK Peters, 1998.
- [2] J. Barraquand and J.-C. Latombe. Robot motion planning: A distributed representation approach. *Intl. J. of Robotics Research*, 10(6):628–649, 1991.
- [3] R. Bohlin and L. E. Kavraki. Path planning using Lazy PRM. In *Proc. Intl. Conf. on Robotics and Automation*, volume 1, pages 521–528, 2000.
- [4] O. Brock and R. Grupen. Final report for the NSF/NASA Workshop on Autonomous Mobile Manipulation (AMM). <http://www-robotics.cs.umass.edu/amm/results.html>, November 2005.
- [5] O. Brock and L. E. Kavraki. Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces. In *Proc. Intl. Conf. on Robotics and Automation*, 2001.
- [6] O. Brock and O. Khatib. Elastic strips: A framework for motion generation in human environments. *Intl. J. of Robotics Research*, 21(12):1031–1052, 2002.
- [7] B. Burns and O. Brock. Toward optimal configuration space sampling. In *Proceedings of Robotics: Science and Systems (RSS)*, 2005.
- [8] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *Intl. J. of Robotics Research*, 18(6):534–555, 1999.
- [9] K.-S. Chang and O. Khatib. Operational space dynamics: Efficient algorithms for modelling and control of branching mechanisms. In *Proc. Intl. Conf. on Robotics and Automation*, pages 850–856, 2000.
- [10] P. C. Chen and Y. K. Hwang. SANDROS: A dynamic graph search algorithm for motion planning. *IEEE Trans. on Robotics and Automation*, 14(3):390–403, 1998.
- [11] W. Choi and J.-C. Latombe. A reactive architecture for planning and executing robot motions with incomplete knowledge. In *Proc. Intl. Conf. on Intelligent Robots and Systems*, volume 1, pages 24–29, 1991.
- [12] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of Robot Motion*. MIT Press, 2005.
- [13] D. C. Conner, A. A. Rizzi, and H. Choset. Composition of local potential functions for global robot control and navigation. In *Proc. Intl. Conf. on Intelligent Robots and Systems*, pages 3546–3551, 2003.
- [14] C. I. Connolly and R. A. Grupen. On the applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, 1993.
- [15] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, third edition, 1994.
- [16] M. Garber and M. C. Lin. Constraint-based motion planning using Voronoi diagrams. In *Proc. of the Workshop on the Algorithmic Foundations of Robotics*, 2002.
- [17] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock. Randomized kinodynamic motion planning with moving obstacles. In *Proc. of the Workshop on the Algorithmic Foundations of Robotics*, pages 247–264, 2000.
- [18] D. Hsu, J.-C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. In *Proceedings of the International Symposium of Robotics Research*, 2005.
- [19] M. Huber and R. A. Grupen. A feedback control structure for on-line learning tasks. *Robotics and Autonomous Systems*, 22(3-4):303–315, 1997.
- [20] L. Jaillet and T. Siméon. A PRM-based motion planner for dynamically changing environments. In *Proc. Intl. Conf. on Intelligent Robots and Systems*, 2004.
- [21] M. Kallmann and M. Mataric. Motion planning using dynamic roadmaps. In *Proc. Intl. Conf. on Robotics and Automation*, pages 4399–4404, 2004.
- [22] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.
- [23] O. Khatib. A unified approach to motion and force control of robot manipulators: The operational space formulation. *International Journal of Robotics and Automation*, 3(1):43–53, 1987.
- [24] J. J. Kuffner and S. M. LaValle. RRT-connect: An efficient approach to single-query path planning. In *Proc. Intl. Conf. on Robotics and Automation*, volume 2, pages 995–1001, 2000.
- [25] H. Kurniawati and D. Hsu. Workspace importance sampling for probabilistic roadmap planning. In *Proc. Intl. Conf. on Intelligent Robots and Systems*, pages 1618–1623, 2004.
- [26] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [27] P. Leven and S. Hutchinson. A framework for real-time path planning in changing environments. *Intl. J. of Robotics Research*, 21(12):999–1030, 2002.
- [28] S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. In *Proc. Intl. Conf. on Robotics and Automation*, volume 2, pages 802–807, Atlanta, USA, 1993.
- [29] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential fields. *IEEE Trans. on Robotics and Automation*, 8(5):501–518, 1992.
- [30] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robots*, 2(4):505–518, 2005.
- [31] T. Siméon, J.-P. Laumond, and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. *Journal of Advanced Robotics*, 14(6):477–494, 2000.
- [32] T. Siméon, J.-P. Laumonde, J. Cortés, and A. Sahbani. Manipulation planning with probabilistic roadmaps. *Intl. J. of Robotics Research*, 23(7-8):729–746, 2004.
- [33] J. P. van den Berg and M. H. Overmars. Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners. In *Proc. Intl. Conf. on Robotics and Automation*, pages 453–460, 2004.
- [34] J. P. van den Berg and M. H. Overmars. Roadmap-based motion planning in dynamic environments. *IEEE Trans. on Robotics and Automation*, 21(5):885–897, 2005.
- [35] J. Vannoy and J. Xiao. Real-time adaptive and trajectory-optimized manipulator motion planning. In *Proc. Intl. Conf. on Intelligent Robots and Systems*, volume 1, pages 497–502, 2004.
- [36] L. Yang and S. M. LaValle. The sampling-based neighborhood graph: A framework for planning and executing feedback motion strategies. In *Proc. Intl. Conf. on Robotics and Automation*, 2003.
- [37] Y. Yang and O. Brock. Adapting the sampling distribution in PRM planners based on an approximated medial axis. In *Proc. Intl. Conf. on Robotics and Automation*, pages 4405–4410, 2004.
- [38] Y. Yang and O. Brock. Efficient motion planning based on disassembly. In *Proceedings of Robotics: Science and Systems (RSS)*, 2005.