

# Motion Planning Under Uncertainty In Highly Deformable Environments

Sachin Patil

Jur van den Berg

Ron Alterovitz

Department of Computer Science, University of North Carolina at Chapel Hill, USA

Email: {sachin,berg,ron}@cs.unc.edu

**Abstract**—Many tasks in robot-assisted surgery, food handling, manufacturing, and other applications require planning and controlling the motions of manipulators or other devices that must interact with highly deformable objects. We present a unified approach for motion planning under uncertainty in deformable environments that maximizes probability of success by accounting for uncertainty in deformation models, noisy sensing, and unpredictable actuation. Unlike prior planners that assume deterministic deformations or treat deformations as a type of small perturbation, our method explicitly considers the uncertainty in large, time-dependent deformations. Our method requires a simulator of deformable objects but places no significant restrictions on the simulator used. We use a sampling-based motion planner in conjunction with the simulator to generate a set of candidate plans based on expected deformations. Our method then uses the simulator and optimal control to numerically estimate time-dependent state distributions based on uncertain parameters (e.g. deformable material properties or actuation errors). We then select the plan with the highest estimated probability of successfully avoiding obstacles and reaching the goal region. Using FEM-based simulation of deformable tissues, we demonstrate the ability of our method to generate high quality plans in two medical-inspired scenarios: (1) guiding bevel-tip steerable needles through slices of deformable tissue around obstacles for minimally invasive biopsies and drug-delivery, and (2) manipulating planar tissues to align interior points at desired coordinates for precision treatment.

## I. INTRODUCTION

Many tasks in robot-assisted surgery, food handling, manufacturing, and other applications require planning and controlling the motions of manipulators or other devices that must interact with highly deformable objects. Due to their difficulty, tasks involving uncertainty and highly deformable objects are still routinely completed manually rather than automatically or semi-autonomously using robot assistance. Automating these tasks could increase productivity and improve outcomes by decreasing the time and costs associated with manual operation while simultaneously increasing accuracy and precision.

Motion planning for tasks in highly deformable environments is challenging because it requires the robot to anticipate deformations in its environment while simultaneously considering uncertainty in those deformations and in its sensing of the system state. Prior work in motion planning has considered the effect of predictable deformations with no uncertainty. These methods often use a physically-based simulation of deforming objects to generate feasible motion plans [17, 4, 11, 24, 10]. These planners assume that the simulator accurately predicts deformations of the objects in the environment, which

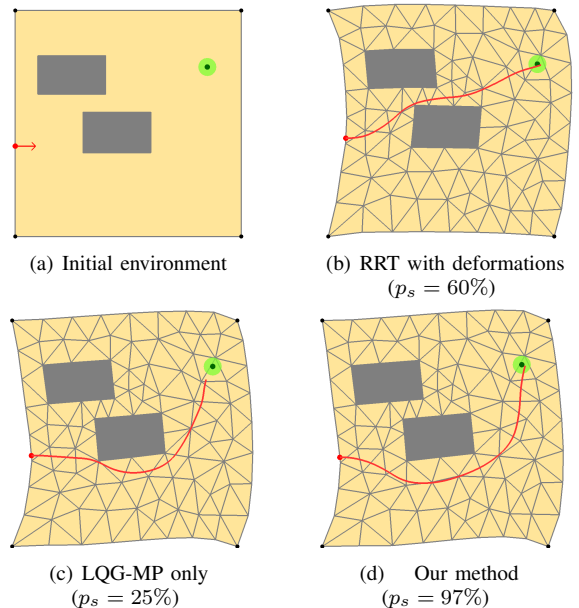


Fig. 1. We illustrate plans for guiding a mobile robot that moves on a deformable plane and pushes the surface forward as it advances. (This model is applicable to steering medical needles through soft tissue.) Current motion planning solutions for deformable environments either assume deterministic deformations (b), which may result in paths through narrow passageways that are highly likely to result in obstacle collision, or compute plans in a static world and consider deformations as a type of perturbation (c), which neglects the large time-dependent motions of the obstacles and target. Our unified framework (d), which accounts for uncertainty in deformation models, noisy sensing, and unpredictable actuation, results in a significantly higher probability of success ( $p_s$ ) in plan execution.

is seldom the case. Other prior work in motion planning has focused on uncertainty and modeled deformations as a type of uncertain perturbation, enabling the use of standard feedback controllers [3, 31]. However, these approaches will not work effectively for problems involving large, history-dependent deformations that fall outside the realm of small perturbations. For example, in robot-assisted surgery and food handling, the robot must interact with tissues that deform significantly and in a history-dependent manner, and these large deformations cannot be predicted with high accuracy due to uncertainty in the underlying tissue properties.

Our goal is to compute motion plans that maximize the probability of success in challenging environments with uncertain deformations. We present a unified approach that combines motion planning with sensing and feedback control for

generation and execution of robust motion plans in deformable environments. We consider uncertainty due to noise in actuation and sensor measurements, as well as uncertainty in deformations arising from erroneous material model assumptions and inaccurate robot/environment interaction models. To the best of our knowledge, our method is the first framework that enables effective computation of motion plans in environments with both large deformations and substantial uncertainty.

Our method requires a simulator of deformable objects. We use the finite element method (FEM) for simulating deformations in our examples, but our approach is generally applicable to other simulation techniques as well. We use a sampling-based motion planner in conjunction with the simulator to generate a set of candidate motion plans that assume expected deformations. Our method then uses the simulator and optimal control to numerically estimate time-dependent state distributions based on uncertain parameters (e.g. deformable material properties or actuation errors). We use this information to generate an optimal linear-quadratic (LQG) feedback controller for each candidate plan to mitigate any uncertainty in the expected deformations that occur during the actual execution of the plan. Since computing an optimal controller using the full deformable system state is computationally prohibitive, we observe that it is possible to formulate the optimal control problem using a subset of the full state space. We then use an extension of the LQG-MP framework [31] to select the plan with the highest estimated probability of successfully avoiding obstacles and reaching the goal region.

We demonstrate the ability of our method to generate high quality plans in two challenging domains. In the first domain, we compute optimal motion plans for a nonholonomic mobile robot that traverses a deformable plane and pushes the surface forward as it advances (as shown in Fig. 1). This model is equivalent to the challenge of guiding bevel-tip steerable needles through highly deformable tissue around obstacles to clinical targets for biopsies and drug delivery under 2D image guidance. In the second domain, the objective is for a manipulator to automatically reposition targets in the interior of a deformable object by applying forces at given points on the object’s surface. This problem arises in manufacturing as well as in alignment of clinical targets in soft tissue for precision treatment. We demonstrate that our method significantly improves the probability of success compared to prior approaches based on standard feedback controllers or motion planners that do not simultaneously consider deformations and uncertainty.

## II. PRIOR WORK

**Modeling of deformable objects:** Physically-based simulation of deformable objects is a well-studied area in solid mechanics and computer graphics [21]. Mass-spring systems, boundary element methods, and finite element methods (FEM) are a few of the popular simulation techniques available. The choice of the simulation technique is application-specific and influences the accuracy of the estimated deformations.

**Motion planning for deformable objects:** Prior work has

investigated motion planners for deformable robots in static environments [17, 4, 11] and in deformable environments [2, 24, 10]. These methods assume deformations can be perfectly predicted and do not take into account uncertainty due to noisy actuation, noisy sensor measurements, or uncertainty in deformation modeling.

Manipulation of internal target points inside a deformable object by a robotic system is necessary in many medical and industrial applications. Wada et al. [33] and Das et al. [8] propose a robust control law for positioning the internal target points by manipulating the boundary points using robotic fingers. Mallapragada et al. [19] and Torabi et al. [30] propose a system to displace an internal target to a needle’s trajectory for image-guided biopsy procedures. These methods use a PD feedback controller that requires mass-spring systems or linear FEM as the deformation simulation technique. These methods do not address motion planning in complex environments with obstacles and large deformations.

A significant body of work exists on motion planning and control for bevel-tip steerable needles. Tissue deformation (without uncertainty) has been taken into account using 2D FEM simulation of soft tissue [2]. Motion uncertainty has been considered in 2D for cases with negligible deformations [3, 1]. Planners for static 3D environments with obstacles have been proposed [9, 22]. Hauser et al. [13] use a fast feedback controller for guiding steerable needles in deformable tissue but do not address obstacle avoidance. Recently, Van den Berg et al. [32] proposed an LQG feedback controller for addressing motion and sensing uncertainty in steerable needle insertion, but this work does not take into account displacement of the target and obstacles due to tissue deformation and uncertainty resulting from deformation. No prior work has successfully computed motion plans in highly deformable environments with uncertainty and obstacles.

**Motion planning under uncertainty:** Uncertainty in motion planning typically originates from noisy actuation and sensing, partial state observations, and uncertainty about the environment. Planners that specifically take into account motion uncertainty include [3, 15]. Roy et al. [25] also consider sensing uncertainty while planning in dynamic environments. Other approaches blend planning and control by defining a global control policy over the entire environment using Markov decision processes (MDPs) [3] and partially-observable MDPs (POMDPs) [16], but these suffer from issues of scalability. Another class of planners focuses on uncertainty about the environment and obstacles [5, 12]. Recently, several methods have been proposed that take into account motion and sensing uncertainty to optimize a user-specified planning objective [23, 28, 31]. These methods do not explicitly take into account large, time-dependent deformations in the environment.

## III. PROBLEM DEFINITION

We consider a robot interacting with a deformable environment. We assume that time is discretized into stages of equal duration. At each time step  $t$ , the robot executes a control  $\mathbf{u}_t$  from its control input space  $\mathcal{U} = \mathbb{R}^q$ . Our objective, formalized

below, is to compute a motion plan and associated feedback controller to maximize the probability of success.

Our planner requires as input a simulator of the deformable system that models the motions and deformations of the robot and the surrounding environment. Given a model of objects in the environment, a set of problem-specific simulation parameters  $\mathbf{s}$ , and a control input  $\mathbf{u}_t$  for the robot, the simulator  $g$  computes the expected motions and deformations of the robot and environment. We define the state space of the deformable system to be  $\mathcal{Y}$ . A deformable system state  $\mathbf{y}_t \in \mathcal{Y}$  at time  $t$  encodes all necessary information to save the state of the simulator and be able to restart it, including deformations, dynamics parameters, friction states, the robot configuration, etc. Formally, the simulator  $g$  evolves as:

$$\mathbf{y}_{t+1} = g(\mathbf{y}_t, \mathbf{u}_t, \mathbf{s}). \quad (1)$$

The simulator  $g$  acts as a deterministic function, but the deformations and motions of the robot and environment may be highly uncertain. The material properties of deformable objects (e.g. Young’s modulus and Poisson’s ratio<sup>1</sup>), interaction parameters such as friction, and deviation from commanded actuation are all uncertain and can significantly affect the motions and deformations of the robot and environment. Due to this uncertainty, we assume  $\mathbf{s}$  comes from some known distribution  $S$ . For simplicity of notation, we assume that  $\mathbf{s}$  is defined such that its expected value is  $\mathbf{0}$ . Selecting different values of  $\mathbf{s} \sim S$  allows us to use the deterministic simulator to consider variations in outcome.

The dimension of the deformable system state space  $\mathcal{Y}$  is very high, possibly containing thousands of elements for large, complex meshes of deformable objects. Computing optimal control policies directly in this high-dimensional state space would be computationally prohibitive. Furthermore, while the initial geometry may be known, it is unlikely that it will be possible to sense and track the entire deformation over time when executing a plan. Instead of working with the entire deformable state  $\mathcal{Y}$ , we pose the control problem by considering a reduced-dimensional space  $\mathcal{X} \in \mathbb{R}^p$  that only contains the attributes that define the state of the robot and/or points in the deformable environment that are necessary for collision detection or goal detection. For instance, when manipulating deformable objects, the reduced state only includes the robot configuration and select points, on the boundary of the deformable objects, whose position can be sensed.

We assume that  $\mathcal{X} \subset \mathcal{Y}$ . Given a state  $\mathbf{x}_t^* \in \mathcal{X}$  at time  $t$  and the applied control input  $\mathbf{u}_t^* \in \mathcal{U}$ , the *expected* state at time  $t + 1$ ,  $\mathbf{x}_{t+1}^* \in \mathcal{X}$ , evolves as:

$$\mathbf{x}_{t+1}^* = f(\mathbf{x}_t^*, \mathbf{u}_t^*, \mathbf{s}) \quad (2)$$

where  $f$  is based on the simulation  $g$  (Eqn. 1). It is important to note that during the actual execution of a given plan, the *true*

<sup>1</sup>The Young’s modulus characterizes the stiffness of elastic materials and the Poisson’s ratio characterizes the compressibility of a deformable material [21]. These quantities are used by simulators to sufficiently describe isotropic, linearly elastic material properties, though additional physical quantities may be needed to describe the behavior of nonlinear elastic and plastic materials.

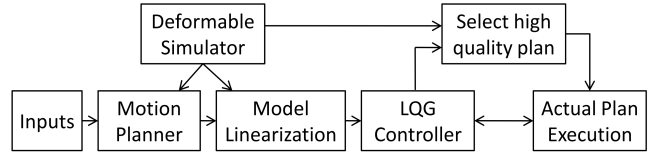


Fig. 2. Schematic overview of our method. The deformable simulator is used for motion planning, model linearization and for selecting a plan that maximizes the probability of success.

state  $\mathbf{x}_t$  departs from the expected output of the deterministic simulation because of uncertainty arising from noisy control inputs and uncertainty in the deformation models.

During the actual plan execution, we also assume that sensors provide us with partial and noisy information about the state according to a given stochastic observation model:

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{n}_t), \quad \mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, N_t) \quad (3)$$

where  $\mathbf{z}_t$  is the measurement obtained at time  $t$  that relates to the true state  $\mathbf{x}_t$  through function  $h$ , and  $\mathbf{n}_t$  is the measurement noise that we assume is drawn from a zero-mean Gaussian distribution with known variance  $N_t$ . It should be noted that some regions of the environment may have better sensing than others and our method automatically takes this into account.

The motion planning problem under uncertainty in deformable environments can be formally stated as follows:

**Objective:** Given a start state  $\mathbf{x}^{\text{start}} \in \mathcal{X}$  and a goal region  $\mathcal{X}^{\text{goal}} \subset \mathcal{X}$ , generate a motion plan and associated feedback controller that maximizes the probability of successfully avoiding obstacles and reaching the goal region.

**Input:** Deformable system simulator  $g$ , deformable system model parameters  $\mathbf{s}$  and their distribution  $S$ , sensor observation model, start state  $\mathbf{x}^{\text{start}}$ , and goal region  $\mathcal{X}^{\text{goal}} \subset \mathcal{X}$ .

**Output:** A motion plan composed of a series of states and corresponding control inputs,  $\pi : (\mathbf{x}_0^*, \mathbf{u}_0^*, \dots, \mathbf{x}_\ell^*, \mathbf{u}_\ell^*), 0 \leq t < \ell$  such that  $\mathbf{x}_0^* = \mathbf{x}^{\text{start}}, \mathbf{x}_\ell^* \in \mathcal{X}^{\text{goal}}$ , and the associated feedback controller for handling uncertainty arising from simulation errors, noisy sensing, and unpredictable actuation.

#### IV. APPROACH

We provide a schematic overview of our method in Fig. 2. We use a simulation-based RRT motion planner [18] to generate a set of candidate plans. We then use the simulator and numerical methods to linearize the model around each computed plan to compute a linear-quadratic Gaussian (LQG) controller [27]. We use an extension of the LQG-MP framework [31] to select the plan with the highest estimated probability of successfully avoiding obstacles and reaching the goal region.

##### A. Motion Planning

Given a start state  $\mathbf{x}^{\text{start}} \in \mathcal{X}$  and a goal region  $\mathcal{X}^{\text{goal}} \subset \mathcal{X}$ , we use the RRT motion planner [18] to generate a set of feasible motion plans  $\Pi$  that connect the start state and the goal region and avoid obstacles in the environment. The plans are generated assuming expected deformations (i.e. there is no uncertainty in the deformations).

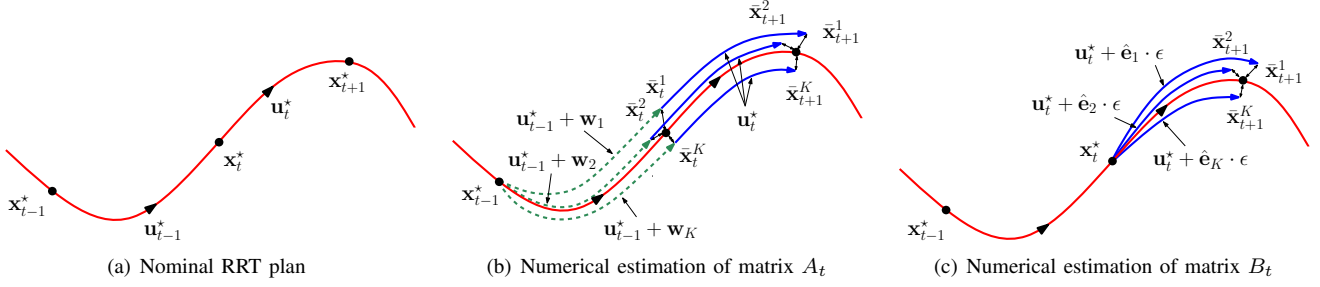


Fig. 3. Model linearization computed numerically using the simulator. (a) Nominal plan  $\pi : (\dots, \mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, \mathbf{x}_t^*, \mathbf{u}_t^*, \mathbf{x}_{t+1}^*, \mathbf{u}_{t+1}^*, \dots)$  computed using the motion planner. The Jacobian matrices  $A_t$  and  $B_t$  are estimated separately using the simulator. (b) We perform  $K = 2p$  independent simulation runs from state  $\mathbf{x}_{t-1}^*$  where control input  $\mathbf{u}_{t-1}^*$  perturbed by Gaussian noise  $\mathbf{w}_k, k \in \{1, 2, \dots, K\}$  and record the deviation from state  $\bar{\mathbf{x}}_t^k = (\mathbf{x}_t - \mathbf{x}_t^*)$  for each run. We then apply the nominal control input  $\mathbf{u}_t^*$  to each of these perturbed states and record the deviation from the subsequent state  $\bar{\mathbf{x}}_{t+1}^k$ . These deviations are used to solve a over-determined linear system to solve for all elements of the  $A_t$  matrix. (c) We perform  $K = q$  independent simulation runs from the nominal state  $\mathbf{x}_t^*$  by perturbing individual elements of the control input  $\mathbf{u}_t^*$  by  $\hat{\mathbf{e}}_k \cdot \epsilon, k \in \{1, 2, \dots, K\}$ , where  $\hat{\mathbf{e}}_k : (\dots, 0, 0, 1_k, 0, 0, \dots)$  and record the deviation from the subsequent state  $\bar{\mathbf{x}}_{t+1}^k$ . These deviations are used to determine the individual columns of the matrix  $B_t$ .

The RRT algorithm incrementally builds a tree-like structure over the state space  $\mathcal{X}$ . Each node in the RRT tree stores the state  $\mathbf{x}$  as well as the full deformable system state  $\mathbf{y} \in \mathcal{Y}$  to enable continuation of the simulation if the node is expanded. At each iteration of the algorithm, we generate a random state  $\mathbf{x}^{\text{sample}} \in \mathcal{X}$  and use a nearest-neighbor algorithm to identify the node in the tree closest to  $\mathbf{x}^{\text{sample}}$ . We then attempt to expand the tree towards  $\mathbf{x}^{\text{sample}}$  by choosing the best known control input  $\mathbf{u} \in \mathcal{U}$  obtained by sampling. For each node expansion, we simulate the deformations in the environment starting from the deformed system state stored at the node.

The RRT tree is grown until the maximum number of permissible iterations is exceeded. A nominal plan composed of a sequence of states and corresponding control inputs  $\pi : (\mathbf{x}_0^*, \mathbf{u}_0^*, \dots, \mathbf{x}_\ell^*, \mathbf{u}_\ell^*)$  can be extracted by traversing the tree from a node in the goal region  $\mathcal{X}^{\text{goal}}$  to the root of the tree containing the start state  $\mathbf{x}^{\text{start}}$ . Multiple such plans are extracted from the tree to generate a set of candidate motion plans  $\Pi$ . It is important to note that the motion plan selected for execution is selected from this set  $\Pi$  and may not necessarily be globally optimal.

### B. Model Linearization

Executing a plan computed in the previous section in an open-loop manner is unlikely to reach the goal region in practice because of uncertainty due to unpredictable actuation, noisy sensor measurements, and uncertainty arising from the deformable object simulation itself. We create a feedback controller to mitigate these uncertainties.

In general, physically-based deformable object simulations can be highly nonlinear, and system identification for nonlinear control for such simulators can be difficult. However, since the deformable system is controlled to stay close to the computed plan, we approximate these nonlinear models by locally linearizing the model around the plan  $\pi$ . In essence, we consider uncertainty about the expected deformation rather than taking the prior approach of considering the deformation itself to be a type of uncertainty.

It is convenient to express the control problem in terms of the *deviation* from the plan. By defining the deviation in

the state as  $\bar{\mathbf{x}}_t = (\mathbf{x}_t - \mathbf{x}_t^*)$ , deviation in control input as  $\bar{\mathbf{u}}_t = (\mathbf{u}_t - \mathbf{u}_t^*)$ , and deviation in measurement as  $\bar{\mathbf{z}}_t = (\mathbf{z}_t - h(\mathbf{x}_t^*, \mathbf{0}))$ , the dynamics and observation models given by Equations (2) and (3) can be linearized as:

$$\begin{aligned} \bar{\mathbf{x}}_{t+1} &= A_t \bar{\mathbf{x}}_t + B_t \bar{\mathbf{u}}_t + \mathbf{m}_t, & \mathbf{m}_t &\sim \mathcal{N}(\mathbf{0}, M_t), \\ \bar{\mathbf{z}}_t &= H_t \bar{\mathbf{x}}_t + W_t \mathbf{n}_t, & \mathbf{n}_t &\sim \mathcal{N}(\mathbf{0}, N_t) \end{aligned} \quad (4)$$

where

$$\begin{aligned} A_t &= \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}_t^*, \mathbf{u}_t^*, \mathbf{0}), & B_t &= \frac{\partial f}{\partial \mathbf{u}}(\mathbf{x}_t^*, \mathbf{u}_t^*, \mathbf{0}), \\ H_t &= \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}_t^*, \mathbf{0}), & W_t &= \frac{\partial h}{\partial \mathbf{n}}(\mathbf{x}_t^*, \mathbf{0}) \end{aligned}$$

are the Jacobian matrices of  $f$  and  $h$  along a given plan  $\pi$ .

If the dynamics model  $f$  is known, then the Jacobian matrices can be computed analytically. In the case of physically-based simulators for deformable objects, it is typically difficult or impossible to compute these Jacobian matrices analytically because the simulations in general cannot be written using closed-form formulas and are not directly differentiable. Given a nominal plan  $\pi : (\dots, \mathbf{x}_{t-1}^*, \mathbf{u}_{t-1}^*, \mathbf{x}_t^*, \mathbf{u}_t^*, \mathbf{x}_{t+1}^*, \mathbf{u}_{t+1}^*, \dots)$ , we numerically estimate the Jacobian matrices  $A_t$  and  $B_t$  at each time-step along the plan.

The Jacobian matrices of the given observation model  $h$ ,  $H_t$  and  $W_t$ , are computed analytically. The stochastic noise term  $\mathbf{m}_t$  models the uncertainty due to simulation parameters  $\mathbf{s} \sim S$  and is assumed to be drawn from a zero-mean Gaussian distribution with variance  $M_t$ , which is also numerically estimated as described below.

**Numerical estimation of matrix  $A_t$ :** In the absence of any deviation from the nominal control input  $\mathbf{u}_t^*$ , the Jacobian matrix  $A_t \in \mathbb{R}^{p \times p}$  describes how the deviation in state evolves from  $\bar{\mathbf{x}}_t$  to  $\bar{\mathbf{x}}_{t+1}$  when the corresponding nominal control input  $\mathbf{u}_t^*$  is applied to the perturbed state  $\mathbf{x}_t = (\mathbf{x}_t^* + \bar{\mathbf{x}}_t)$ . This is given by the relation  $\bar{\mathbf{x}}_{t+1} = A_t \bar{\mathbf{x}}_t$ . By performing  $K = p$  independent simulation runs from states  $(\mathbf{x}_t^* + \bar{\mathbf{x}}_t^k), k \in \{1, 2, \dots, K\}$  to states  $(\mathbf{x}_{t+1}^* + \bar{\mathbf{x}}_{t+1}^k)$ , we can appropriately assemble the deviations in the consecutive states into matrices  $\bar{X}_t$  and  $\bar{X}_{t+1}$  and estimate the matrix  $A_t$  numerically as  $A_t = \bar{X}_{t+1} \bar{X}_t^{-1}$ .

It is not possible to perturb the nominal state  $\mathbf{x}_t^*$  to  $(\mathbf{x}_t^* + \bar{\mathbf{x}}_t)$  within a physically-based deformable simulator. For instance, in the case of a needle procedure, this would correspond to the displacement and re-orientation of the tip, which is not possible without affecting the entire needle trajectory and adjusting needle/tissue interaction parameters. So instead we perform  $K$  independent simulation runs from the previous state  $\mathbf{x}_{t-1}^*$ , where each run  $k \in \{1, 2, \dots, K\}$  involves the application of control input  $\mathbf{u}_{t-1}^*$  perturbed by noise  $\mathbf{w}_k$ , where we assume that  $\mathbf{w}_k$  is drawn from a user-specified zero-mean Gaussian with variance  $W$ . This generates a set of states  $(\mathbf{x}_t^* + \bar{\mathbf{x}}_t^k)$ ,  $k \in \{1, 2, \dots, K\}$  within the simulator, as shown in Fig. 3(b). We then perform  $K$  independent simulation runs from the set of states  $(\mathbf{x}_t^* + \bar{\mathbf{x}}_t^k)$  by applying the nominal control input  $\mathbf{u}_t^*$  to the robot to generate a set of states  $(\mathbf{x}_{t+1}^* + \bar{\mathbf{x}}_{t+1}^k)$ . We then assemble the deviations from the nominal state  $\bar{\mathbf{x}}_{t+1}^k$  into a matrix  $\bar{X}_{t+1}$ . The matrix  $A_t$  can then be estimated as described above. In practice, we found that performing  $K (> p)$  simulation runs and solving the resultant over-determined least-squares problem  $A_t \bar{X}_t = \bar{X}_{t+1}$  by taking the Moore-Penrose pseudo-inverse of matrix  $\bar{X}_t$  yielded better results at the expense of computational overhead. We used  $K = 2p$  simulation runs for all our experiments.

**Numerical estimation of matrix  $B_t$ :** In the absence of any deviation from the nominal state  $\mathbf{x}_t^*$ , the matrix  $B_t \in \mathbb{R}^{p \times q}$  describes the relationship between deviations in control input  $\bar{\mathbf{u}}_t$  and the deviation in the subsequent state  $\bar{\mathbf{x}}_{t+1}$ . This is given by the relation  $\bar{\mathbf{x}}_{t+1} = B_t \bar{\mathbf{u}}_t$ . We perform  $K = q$  independent simulation runs from the nominal state  $\mathbf{x}_t^*$  by perturbing individual elements of the control input  $\mathbf{u}_t^*$  by  $\hat{\mathbf{e}}_k \cdot \epsilon$ ,  $k \in \{1, 2, \dots, K\}$ , where  $\hat{\mathbf{e}}_k$  is a unit vector given by  $(\dots, 0, 0, 1_k, 0, 0, \dots)$  and  $\epsilon$  is a user-defined perturbation constant. This generates a set of states  $(\mathbf{x}_{t+1}^* + \bar{\mathbf{x}}_{t+1}^k)$ , as shown in Fig. 3(c). The deviations  $\bar{\mathbf{x}}_{t+1}^k$  scaled by the perturbation constant  $\epsilon$  comprise the individual columns of the matrix  $B_t$ .

**Numerical estimation of matrix  $M_t$ :** The stochastic noise term  $\mathbf{m}_t$  models uncertainty in the simulation parameters  $\mathbf{s} \sim S$ . It is reasonable to assume that the uncertainty arising from a large number of sources can be modeled as a Gaussian distribution. Here,  $\mathbf{m}_t$  is assumed to be drawn from a zero-mean Gaussian distribution with variance  $M_t$ . Since the uncertainty distributions modeling the noise in control inputs and variance in simulation parameters  $S$  are supplied by the user, the variance  $M_t$  can be estimated a priori by performing simulations by perturbing the simulation parameters  $\mathbf{s}$  independently and estimating the parameters of the Gaussian distribution that models the resulting uncertainty.

### C. LQG Control

Given linear(ized) dynamics and observation models and a quadratic cost function, the optimal approach for executing a plan is to use a linear-quadratic regulator (LQR) feedback controller in combination with a Kalman filter for state estimation. This ensemble is called linear-quadratic Gaussian (LQG) control [27] and is provably optimal for state estimation and control for linear systems.

We use an extended Kalman filter (EKF) [26] for optimal state estimation during actual execution of the plan. The Kalman filter keeps track of the estimate  $\tilde{\mathbf{x}}_t$  and variance of the deviation in true state  $\bar{\mathbf{x}}_t$  during control. It continually performs two steps; a process update to propagate the applied control input, and a measurement update to incorporate the obtained measurement.

We compensate for uncertainty during plan execution by using an LQR feedback controller that aims to keep the true state close to the corresponding nominal state in the plan. The LQR formulation seeks the optimal control inputs by minimizing a quadratic cost function that seeks to simultaneously minimize deviations from the plan and deviations from the control input. Solving the cost function gives the control policy  $\bar{\mathbf{u}}_t = L_t \tilde{\mathbf{x}}_t$ , for feedback matrices  $L_t$  that are pre-computed using a standard recursive procedure. We refer the reader to [27, 31] for additional details.

### D. Selecting a High Quality Plan

We use an extension of the LQG-MP framework [31] to select a plan with the highest estimated probability of successfully avoiding obstacles and reaching the goal region. Given the LQG controller for a plan  $\pi$ , we compute the a priori distributions of the deviation in the true state  $\bar{\mathbf{x}}_t$  and the estimated deviation  $\tilde{\mathbf{x}}_t$ . These distributions are used to approximately compute the probability of collisions with obstacles during actual plan execution. We also estimate the probability of reaching the goal region by sampling the a priori distribution at the final time-step and determining how many of those samples lie within the goal region. It is important to take into account the deformed configurations of the goal and obstacles in the environment at each time-step along the plan. The cumulative probability of success is then computed as the product of the two estimated probabilities. We refer the reader to Van den Berg et al. [31] for additional details.

## V. EXAMPLE APPLICATIONS AND RESULTS

We present results for two scenarios with different planning objectives: (1) guiding a nonholonomic mobile robot (such as a bevel-tip steerable needle) through a deformable environment with obstacles to reach a desired goal, and (2) externally manipulating a planar deformable object to reposition internal points to desired coordinates. We implemented the planner in C++ and tested it on a 3.33 Ghz Intel® i7™ PC.

### A. Nonholonomic Robot

We apply our method to a simple car model without reverse [18] that traverses a planar deformable environment and pushes the surface forward as it advances. This model is equivalent to a simplified model of the bevel-tip steerable needle moving through a planar slice of deformable tissue [20]. These needles bend when inserted into soft tissue and can be steered within tissue by re-orienting the bevel tip. Inserting needles into tissues also causes the surrounding tissue to deform, thereby introducing a challenging planning problem.

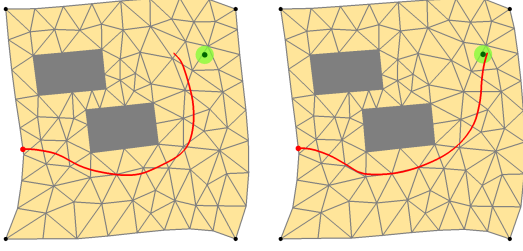


Fig. 4. Actuation noise, in the absence of feedback, causes the nonholonomic robot to veer away from the goal region in the deformable environment (left). Our feedback controller compensates for this noise to guide the robot to the goal in the deformable environment (right).

We model the deformable environment as a unit square that is fixed at the four corners, as shown in Fig. 1(a). The state of the robot  $\mathbf{x}_t = (x_t, y_t, \theta_t)$  is a vector consisting of its position  $(x_t, y_t)$  and its orientation  $\theta_t$  at time  $t$ . We do not include any points from the deformable environment in the definition of the state  $\mathbf{x}_t$ . The control input  $\mathbf{u}_t = (v_t, \phi_t)$  consists of the speed of the robot  $v_t \in [0, v_{\max}]$  and the steering angle  $\phi_t \in [-\phi_{\max}, \phi_{\max}]$ . For a bevel-tip steerable needle, the continuous steering angle can be transformed into duty-cycling parameters for actuation at the needle base [20]. The state of the robot then evolves as:

$$\mathbf{x}_{t+1} = \begin{bmatrix} x_t + \tau v_t \cos \theta_t \\ y_t + \tau v_t \sin \theta_t \\ \theta + v_t \tan(\phi_t) / (r_{\min} \tan(\phi_{\max})) \end{bmatrix}$$

where  $\tau$  is the duration of a time-step and  $r_{\min}$  is the minimum turning radius of the nonholonomic robot. In our experiments, we used  $v_{\max} = 0.5$ ,  $\phi_{\max} = \pi/3$ ,  $\tau = 0.1$ , and  $r_{\min} = 0.25$  where all quantities are expressed in appropriate units.

Inspired by needle steering, the motion of the robot exerts an interaction force of magnitude  $f^{\text{int}} = 1$  in the direction of its movement, thus deforming the environment. We model the environment as an isotropic, linearly elastic material and use a linear FEM simulator to compute the deformations in the environment as a result of the applied forces.

We also assume that the robot only receives feedback on its position and not its orientation, i.e.  $h(\mathbf{x}_t) = \begin{bmatrix} x \\ y \end{bmatrix}$ . This is a reasonable assumption since current medical imaging technologies such as ultrasound do not allow for measuring the full state of the needle tip (as the imaging resolution is often too low to infer its orientation). The noise in the sensor measurement is modeled as  $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, N_t)$ , where the variance in sensing noise  $N_t$  is known.

Given the initial state  $\mathbf{x}^{\text{start}}$  of the robot and the goal region  $\mathcal{X}^{\text{goal}}$ , the planning objective for the robot is to move to the goal region without colliding with any obstacles (Fig. 1(a)). We used the RRT algorithm as described in Sec. IV-A to generate a set of 100 candidate motion plans, which took 81 seconds. Fig. 1(b) shows one such plan. We executed the remainder of our method to select a high quality plan, as shown in Fig. 1(d). This process took 57 seconds. The output of our method is the selected motion plan and the corresponding LQG controller.

To demonstrate the effectiveness of the controller computed

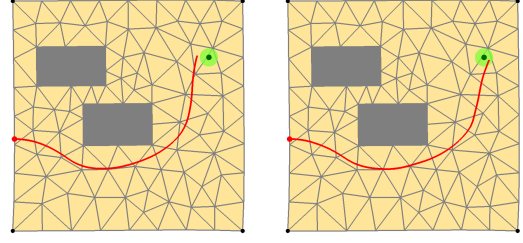


Fig. 5. A much stiffer than expected material composing the deformable environment causes the robot to miss the goal region in the absence of feedback (left). Our controller guides the robot back to the goal region even with high variance in material parameters (right).

by our method, we show actual plan executions under actuation noise (Fig. 4) and varying material properties (Fig. 5). The controller computed by our method is successfully able to correct for uncertainty in actuation and deformation modeling.

To evaluate our method, we simulate 1000 executions of the planner solution to evaluate the percentage of successful plan executions. Execution of a plan is considered successful if the goal region is reached and obstacles are avoided. It is important to note that each simulated plan execution assumes no knowledge of the internal deformable model being used. We considered the following scenarios to model uncertainty in actuation, sensing, and deformation modeling:

- 1) *Material properties (low variance)*: We model the environment to have Young's modulus  $E \sim \mathcal{N}(50.0, 5.0)$  and Poisson's ratio  $\nu \sim \mathcal{N}(0.4, 1e^{-04})$ .
- 2) *Material properties (high variance)*: We model the environment to have Young's modulus  $E \sim \mathcal{N}(50.0, 75.0)$  and Poisson's ratio  $\nu \sim \mathcal{N}(0.4, 4e^{-04})$ .
- 3) *Actuation noise*: The control inputs  $\begin{bmatrix} v_t \\ \phi_t \end{bmatrix}$  are perturbed by noise drawn from a zero-mean Gaussian distribution  $\sim \mathcal{N}(\mathbf{0}, \begin{bmatrix} 0.005 & 0.0 \\ 0.0 & 0.025 \end{bmatrix})$ .
- 4) *Number of mesh elements*: The number of elements in the discretized mesh of the environment is uniformly distributed between 50 and 1000 elements, where the expected mesh contains 175 elements.
- 5) *Interaction force*: We model the interaction force  $f^{\text{int}}$  to be uniformly distributed between 0 (no deformations) and 2 units.

We compared the results of our method to two existing approaches. First, we consider *RRT with deformations* in which we use a RRT plan as computed in section IV-A but do not explicitly consider uncertainty (i.e. no feedback control). Second, we consider *LQG-MP only* in which a plan is selected by LQG-MP [31] from a set of candidate plans computed in a static environment, which treats deformation as a source of uncertainty. For each approach, we simulate plan execution 1000 times using the scenarios above.

As shown in Fig. 6, our method consistently yields a significantly higher probability of success compared to existing approaches that plan in deterministic deformable environments without uncertainty or consider deformation as a type of uncertain perturbation. The poor rate of success of the RRT with deformations plan is due to the lack of any feedback control to compensate for uncertainty due to actuation noise

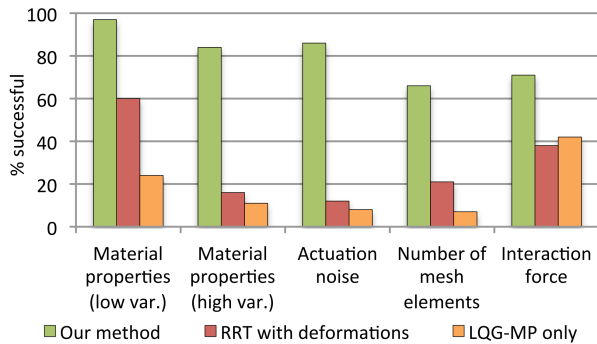


Fig. 6. Percentage of successful plan executions. The plan chosen by our method consistently outperforms plans that do not simultaneously account for both uncertainty and deformations.

and deformation modeling errors. On the other hand, the LQG-MP plan has a poor rate of success because it fails to account for large time-dependent motions of the goal and obstacles due to deformations.

We also evaluated the impact of the LQG-MP plan selection step of our method. Averaging across all the scenarios considered above, our method without the LQG-MP step performed 34% better than RRT with deformations and 17% worse than our complete method. The feedback controller significantly improves the rate of success by mitigating uncertainty encountered during execution of the plan. The use of the LQG-MP step further improves the rate of success at the expense of additional computation.

### B. Internal Point Manipulation

We also apply our method to external manipulation of planar deformable objects to reposition internal points to desired coordinates. This is an important step in many manufacturing processes such as creating seamless garments and mating flexible parts [33, 8]. It could also be useful in aligning clinical targets in soft tissue for precision radiation cancer treatment [7] or to expose targets for access via minimally invasive needle insertion procedures [19, 30]. Automatic repositioning of internal targets via external forces could result in greater accuracy and faster recovery times.

We model the deformable object as shown in Fig. 7(a). The state  $\mathbf{x} = \begin{bmatrix} \mathbf{p}^{\text{internal}} \\ \mathbf{p}^{\text{boundary}} \end{bmatrix}$  includes the position of the internal target point  $\mathbf{p}^{\text{internal}}$  and position of points located on the environment boundary  $\mathbf{P}^{\text{boundary}}$  (included for sensing and collision detection). The control input  $\mathbf{u}_t = \mathbf{f}^{\text{ext}}$  includes external forces being applied by the three manipulators (Fig. 7(a)). We use a geometrically nonlinear FEM simulator to accurately simulate the large deformations of the object.

We assume that only the positions of points on the boundary can be sensed using imaging modalities, i.e.  $h(\mathbf{x}_t) = \begin{bmatrix} \mathbf{P}^{\text{boundary}} \end{bmatrix}$ . The position of the internal point is estimated using the Kalman filter during plan execution. The noise in the sensor measurements is modeled as  $\mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, N_t)$ , where the variance in sensing noise  $N_t$  is known.

Given the initial state  $\mathbf{x}^{\text{start}}$  and the goal region  $\mathcal{X}^{\text{goal}}$ , the planning objective is to move the internal point to the goal

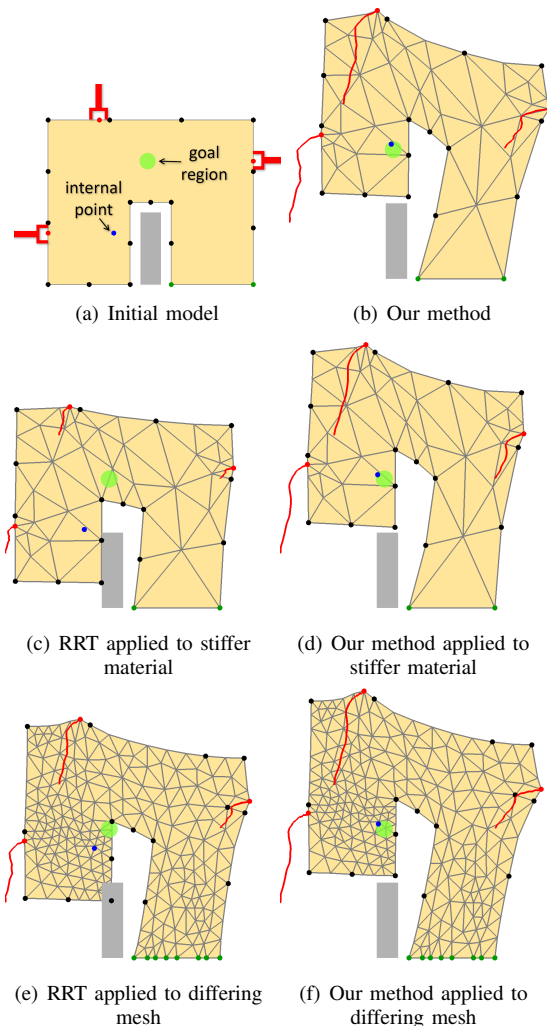


Fig. 7. External manipulation of deformable tissue to reposition an internal target point. (a) Three external manipulators (red dots) manipulate the tissue by applying external forces. Only the position of points on the boundary (black dots) can be sensed. The objective is to reposition the target point (blue dot) to the goal region (green) while avoiding the static obstacle (gray). The bottom right of the tissue is fixed. (b) Expected outcome of the plan computed by our method. The trace of the manipulator positions over time is shown in red. (c) Variation in material parameters causes collisions in the absence of feedback. (d) Our controller successfully guides the internal point to the goal region while avoiding collisions with the obstacle. (e) Failure of the planning objective when the number of mesh elements is increased. (f) Modeling errors due to the mesh discretization are mitigated by the controller.

region such that the deformable object does not collide with any obstacles. We used the RRT algorithm to generate a set of 100 candidate motion plans, which took 438 seconds. We executed the remainder of our method to select a high quality plan, as shown in Fig. 7(b). This process took 386 seconds.

Figures 7(c) and 7(d) show actual execution of the selected plan without and with feedback when there is variation in material parameters. Figures 7(e) and 7(f) show actual execution of the selected plan without and with feedback when simulation errors are introduced due to a differing discretization of the environment geometry. For both these experiments, we use the same plan and corresponding controller as computed by our method. In both cases, our method successfully achieves the task subject to uncertainty in deformation modeling.

## VI. CONCLUSION AND FUTURE WORK

We have introduced a new, unified framework for motion planning under uncertainty in highly deformable environments that maximizes the probability of success by accounting for uncertainty in deformation models, noisy sensing, and unpredictable actuation. Unlike prior planners that assume deterministic deformations or treat deformations as a disturbance, our method explicitly considers uncertainty in large, time-dependent deformations. Although the method requires a simulator of the deformable environment, we place no significant restrictions on the simulator used. We have shown that our approach can generate high quality plans for guiding bevel-tip steerable needles through highly deformable tissue slices and for repositioning targets in the interior of a deformable object.

In future work, we plan to investigate improvements to each component in Fig. 2. Replacing the standard LQR control framework with approaches such as iterative LQR [29] or differential dynamic programming (DDP) [14] may improve controller performance. Similarly, replacing the standard Kalman filter with variants such as the unscented Kalman filter or a particle filter [26] may improve the quality of state estimation during plan execution. We will also investigate parallelizing the model linearization, which involves multiple, independent simulation runs, in order to reduce computation times. Finally, we plan to extend our method to 3D deformable environments [11, 6] and physically validate our results by conducting experiments on bevel-tip steerable needle guidance and internal target manipulation in tissue phantoms.

## ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation (NSF) under grant #IIS-0905344 and by the National Institutes of Health (NIH) under grant #R21EB011628. The authors thank Pieter Abbeel, Ken Goldberg, and M. Cenk Çavuşoğlu for their valuable input on this problem.

## REFERENCES

- [1] R. Alterovitz, M. Branicky, and K. Goldberg, "Motion planning under uncertainty for image-guided medical needle steering," *Int. J. Robotics Research*, vol. 27, no. 11–12, pp. 1361–1374, Nov. 2008.
- [2] R. Alterovitz, K. Goldberg, and A. M. Okamura, "Planning for steerable bevel-tip needle insertion through 2D soft tissue with obstacles," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Apr. 2005, pp. 1652–1657.
- [3] R. Alterovitz, T. Siméon, and K. Goldberg, "The Stochastic Motion Roadmap: A sampling framework for planning with Markov motion uncertainty," in *Proc. Robotics: Science and Systems III*, June 2007.
- [4] O. B. Bayazit, J.-M. Lien, and N. M. Amato, "Probabilistic roadmap motion planning for deformable objects," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2002, pp. 2126–2133.
- [5] B. Burns and O. Brock, "Sampling-based motion planning with sensing uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2007, pp. 3313–3318.
- [6] N. Chentanez, R. Alterovitz, D. Ritchie, J. Cho, K. K. Hauser, K. Goldberg, J. R. Shewchuk, and J. F. O'Brien, "Interactive simulation of surgical needle insertion and steering," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, pp. 88:1–88:10, Aug. 2009.
- [7] Cyberknife, "Cyberknife," Available: <http://www.cyberknife.com>, 2010.
- [8] J. Das and N. Sarkar, "Planning and control of an internal point of a deformable object," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2010, pp. 2877–2882.
- [9] V. Duindam, R. Alterovitz, S. Sastry, and K. Goldberg, "Screw-based motion planning for bevel-tip flexible needles in 3D environments with obstacles," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2008, pp. 2483–2488.
- [10] B. Frank, M. Becker, C. Stachniss, W. Burgard, and M. Teschner, "Efficient path planning for mobile robots in environments with deformable objects," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2008, pp. 3737–3742.
- [11] R. Gayle, P. Segars, M. C. Lin, and D. Manocha, "Path planning for deformable robots in complex environments," in *Proc. Robotics: Science and Systems (RSS)*, 2005, pp. 225–232.
- [12] L. Guibas, D. Hsu, H. Kurniawati, and E. Rehman, "Bounded uncertainty roadmaps for path planning," in *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, 2008.
- [13] K. Hauser, R. Alterovitz, N. Chentanez, A. Okamura, and K. Goldberg, "Feedback control for steering needles through 3D deformable tissue using helical paths," in *Proc. Robotics: Science and Systems (RSS)*, 2009.
- [14] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*. American Elsevier Publishing Company, 1970.
- [15] G. Kewlani, G. Ishigami, and K. Iagnemma, "Stochastic mobility-based path planning in uncertain environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 1183–1189.
- [16] H. Kurniawati, D. Hsu, and W. Lee, "SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," in *Proc. Robotics: Science and Systems (RSS)*, 2008.
- [17] F. Lamiroux and L. E. Kavraki, "Planning paths for elastic objects under manipulation constraints," in *Int. Journal of Robotics Research*, vol. 20, no. 3, 2001, pp. 188–208.
- [18] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, Available at <http://planning.cs.uiuc.edu>.
- [19] V. G. Mallapragada, N. Sarkar, and T. K. Podder, "Robot assisted realtime tumor manipulation for breast biopsy," in *IEEE Trans. on Robotics*, vol. 25, no. 2, 2009, pp. 316–324.
- [20] D. Minhas, J. A. Engh, M. M. Fenske, and C. Riviere, "Modeling of needle steering via duty-cycled spinning," in *Proc. Int. Conf. IEEE Eng. in Medicine and Biology Society*, 2007, pp. 2756–2759.
- [21] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson, "Physically based deformable models in computer graphics," in *Computer Graphics Forum*, vol. 25, no. 4, 2006, pp. 809–836.
- [22] S. Patil and R. Alterovitz, "Interactive motion planning for steerable needles in 3D environments with obstacles," in *Int. Conf. Biomedical Robotics and Biomechanics (BioRob)*, 2010, pp. 893–899.
- [23] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robotics Research*, vol. 28, no. 11–12, pp. 1448–1465, 2009.
- [24] S. Rodriguez, J.-M. Lien, and N. M. Amato, "Planning motion in completely deformable environments," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2006, pp. 2466–2471.
- [25] N. Roy, W. Burgard, D. Fox, and S. Thrun, "Coastal navigation - mobile robot navigation with uncertainty in dynamic environments," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 1999, pp. 35–40.
- [26] D. Simon, *Optimal State Estimation: Kalman, H-infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.
- [27] R. F. Stengel, *Optimal Control and Estimation*. Dover Publications, 1994.
- [28] R. Tedrake, "LQR-trees: Feedback motion planning on sparse randomized trees," in *Proc. Robotics: Science and Systems (RSS)*, 2009.
- [29] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proc. American Control Conference*, 2005, pp. 300–306.
- [30] M. Torabi, K. Hauser, R. Alterovitz, V. Duindam, and K. Goldberg, "Guiding medical needles using single-point tissue manipulation," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2009, pp. 2705–2710.
- [31] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," in *Proc. Robotics: Science and Systems*, June 2010.
- [32] J. van den Berg, S. Patil, R. Alterovitz, P. Abbeel, and K. Goldberg, "LQG-based planning, sensing, and control of steerable needles," in *Algorithmic Foundations of Robotics IX (Proc. WAFR 2010)*, vol. 68. Springer, 2010, pp. 373–389.
- [33] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji, "Robust manipulation of deformable objects by a simple pid feedback," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2001, pp. 85–90.