Gaussian Belief Space Planning for Imprecise Articulated Robots

Alex Lee Sachin Patil John Schulman Zoe McCarthy Jur van den Berg Ken Goldberg Pieter Abbeel

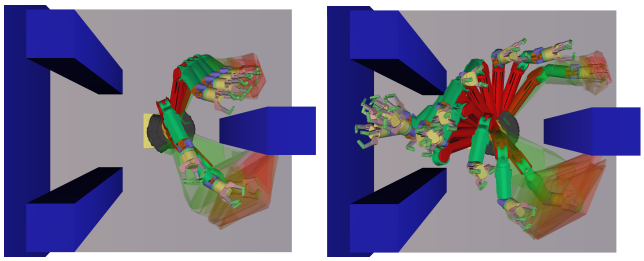
***Abstract*— For many emerging applications, actuators are being developed to avoid injuring humans and actuation and sensing precision is reduced to keep costs down. For planning and control of such robots, *belief space* and POMDP-based methods show great promise but have primarily been applied to cases where robots can be approximated as points or spheres. We extend this framework to non-spherical articulated robots operating under Gaussian models of motion and sensing uncertainty. For collision avoidance in Gaussian belief spaces, we propose the use of *sigma hulls*, which are convex hulls of the robot geometry transformed according to the sigma standard deviation boundary points generated by the Unscented Kalman filter (UKF). We analyze signed distances between sigma hulls and obstacles in the workspace to formulate collision avoidance constraints compatible with the Gilbert-Johnson-Keerthi (GKJ) and Expanding Polytope Algorithm (EPA) for collision avoidance within an optimization-based planning framework. We present results in simulation for planning motions for a 7-DOF articulated robot with imprecise actuation and equipped with inexpensive, inaccurate sensors. Our experiments suggest that this framework is computationally efficient and effective for reducing the probability of collision during execution by re-planning after every time step.**

I. Introduction

Our work is motivated by the desire to facilitate robust operation of cost-effective robots such as the Raven surgical robot [27], Baxter manufacturing robot [26], and low-cost manipulators [25]. These robots use cheaper actuation methods such as cable-driven mechanisms and serial elastic actuators that are less precise than stiff geared actuators. They also rely on cheaper, inaccurate encoders and other sensors such as accelerometers to sense the robot state. For such robots to robustly complete navigation and manipulation tasks under motion and sensing uncertainty, it is important for the robot to explicitly perform information gathering actions to minimize the effects of uncertainty.

The problem of maximizing information gain under uncertainty is often formalized as a Partially Observable Markov Decision Process (POMDP) [14] and is defined over the space of probability distributions of the state space, also referred to as the belief space. The objective is to compute an optimal action for each possible belief state such that the expected reward of completing the task is maximized. Finding globally optimal solutions to the POMDP problem is computationally intractable, even with finite planning horizons, discrete states, actions, and observations [6]. As a result, a large subset of prior work has focused on approximately solving the POMDP problem for Gaussian belief spaces using sampling-based motion planners [23], [5] or optimization-based methods [19], [10], [21], [33], [31].

Collision detection is an integral component of motion planning. Deterministic motion planning algorithms plan for robot motions in the configuration space by performing collision detection in the workspace since computing configuration space obstacles is not practical [18]. However, for planning in belief spaces, collision detection needs to be performed in a probabilistic sense by considering collisions with respect to all possible states that the robot could be in. Prior Gaussian belief space planning methods only consider point robots or spherical approximations of the robot geometry, in which case probabilistic collision detection can be performed efficiently in the workspace [5], [33], [31]. In the case of articulated robots that are not well approximated as spheres, one would have to apply these methods directly in the configuration space. An alternative approach would be to use Monte Carlo sampling [17], [9], which would be very computationally expensive for the belief space planning.



(a) Uncertainty unaware RRT plan (b) Gaussian belief space plan

Fig. 1. Simulated trajectory traces for a 7-DOF articulated robot with imprecise actuation and cheap, inaccurate sensors moving in a constrained environment with obstacles (blue). The robot localizes itself based solely on distance of the robot end-effector from a wall in the environment (far left), with the signal strength decaying quadratically with the distance. We visualize the uncertainty at the initial state and target in terms of the *sigma hulls* (rendered translucent). (a) A naïve trajectory computed using a rapidly exploring random trees planner (RRT) [18] produces a trajectory that avoids collisions with obstacles but is oblivious to the uncertainty in the robot state. (b) With Gaussian belief space planning, the robot executes a trajectory that leads it close to the wall for reliable localization before reaching the target with reduced uncertainty.

Our key insight for collision avoidance in Gaussian belief spaces is to use *sigma hulls*, which are convex hulls of the robot geometry transformed according to the sigma points on the boundary of the standard deviation contour of the Gaussian distribution as computed by the Unscented Kalman filter (UKF). We used the signed distances between the sigma hulls and workspace obstacles to formulate collision avoidance constraints within an optimization-based planning framework based on sequential convex optimization [3]. We rely on advances in collision detection [1] to formulate these constraints without explicitly computing the convex hulls. We show that the collision avoidance constraints can be locally approximated by convex constraints, and we derive how to compute this convex approximation analytically.

We adopt the approach of Platt et al. [21] to use trajectory optimization [2] for computing a locally optimal trajectory in belief space. The optimization formulation incorporates constraints on the state and control inputs and is well-suited for highly under-actuated belief space planning problems. In addition, we build on recent advances in trajectory optimization in the state space [28] to plan motions for articulated robots in Gaussian belief spaces. As is standard in nonlinear optimization for control, we follow the model predictive control (MPC) paradigm [7] of re-planning after every time step. This has been demonstrated to be a very effective way of performing feedback control to remain robust to large perturbations, provided one can re-plan sufficiently fast.

We present results in simulation for planning motions for a 7-DOF articulated robot with imprecise actuation and equipped with cheap, inaccurate sensors such as accelerometers and distance sensors. We exploit the symmetry in the covariance matrix to plan in a 35 dimensional belief space. In the scenarios considered, our implementation is able to compute an initial locally optimal belief space trajectory in constrained workspaces with obstacles in less than 3 seconds on a commodity processor, by solving a constrained nonlinear optimization problem. During execution, we re-plan after every time step to take the current belief state into account. Our experiments indicate that re-planning significantly reduces the probability of collision during execution from 83% for open-loop execution of the initial trajectory down to 6% for re-planning after every time step.

These initial results are encouraging and we posit that advances in optimization techniques and computational hardware will eventually make our approach computationally efficient to enable robots to plan at a high frequency in belief spaces. This would be an important step towards enabling low-cost robots to safely and robustly complete navigation and manipulation tasks in unstructured environments while operating under considerable uncertainty.

II. Related Work

A considerable body of work exists for planning motions for articulated robots to accomplish navigation and manipulation tasks [18]. Recently, research efforts have focused on the planning problem under uncertainty, which in generality is formalized as a POMDP, computing solutions for which is computationally intractable [6]. For problems involving discrete state, action, and/or observation spaces, algorithms have been developed that use approximate value iteration [8], [22], [16], [29]. However, for problems more naturally defined over continuous state, action, and observation spaces such as those arising in robot manipulation and navigation, discretizing the problem and using the aforementioned approaches leads to an exponential growth in the number of states, subjecting these problems to the "curse of dimensionality". The methods of [30], [4], [12] handle continuous state and action spaces, but maintain a global (discrete) representation of the value function over the belief space, which limits their applicability to small to medium sized domains.

For problems in which it is reasonable to model beliefs as Gaussian distributions, beliefs can be parameterized in terms of the mean and the covariance of the Gaussian and sampling-based motion planning [23], [31], [5] or optimization-based methods [19], [10], [21], [33], [32] can be used for belief space planning. This concept can be extended to non-Gaussian beliefs [20] by using particle filters. These methods handle continuous spaces but only consider robots with point-like or spherical geometries to simplify probabilistic collision detection.

We adopt the approach of Platt et al. [21] to use trajectory optimization [2] in belief space assuming deterministic belief dynamics. We generalize this approach to plan motions for articulated robots that are not well approximated as points or spheres. The uncertainty during execution is mitigated by re-planning after each time step [7]. Our work extends the trajectory optimization approach of Schulman et al. [28] from state space to belief space. In doing so, we propose a novel formulation of collision avoidance constraints in belief space using sigma hulls.

III. Preliminaries and Objective

We consider an articulated robot with K links. We are given a description of the robot geometry and a geometric description of the obstacles $\mathcal{O}$ in the workspace. Let $\mathbf{x}$ denote the robot state, which includes the degrees of freedom (DOF) of the robot such as joint angles and position of the base. We refer to the position of a given point on the robot geometry in a world coordinate frame as a function of $\mathbf{x}$ as $\mathbf{p}(\mathbf{x})$ and the pose (both position and orientation) as $\boldsymbol{\psi}(\mathbf{x})$, which is evaluated in closed form using the robot kinematics model. We assume that the state space trajectory is discretized into time intervals of equal duration $\mathcal{T} = \{0, 1, \ldots, T\}$.

At each time step $t \in \mathcal{T}$, the stochastic kinematics of the robot state $\mathbf{x}_t$ evolves according to the given model:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{q}_t), \quad \mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, I_{\dim[\mathbf{q}]}), \tag{1}$$

where $\mathbf{u}_t \in F_{\mathcal{U}}$ is a control input drawn from the set of feasible control inputs $F_{\mathcal{U}}$ and $\mathbf{q}_t$ is the motion noise. Although the state is not observed directly, noisy observations $\mathbf{z}_t$ are obtained using sensors and are related to the state $\mathbf{x}_t$ according to the given stochastic model:

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t), \quad \mathbf{r}_t \sim \mathcal{N}(\mathbf{0}, I_{\dim[\mathbf{r}]}), \tag{2}$$

where $\mathbf{r}_t$ is the sensing noise. Without loss of generality, both $\mathbf{q}_t$ and $\mathbf{r}_t$ are drawn from independent Gaussian distributions with zero mean and unit variance and can be scaled appropriately to be state and control input dependent within the functions $\mathbf{f}$ and $\mathbf{h}$, respectively.

Given Gaussian models of motion and sensing uncertainty considered above, the probability distribution over the state or belief is parameterized as a Gaussian distribution. Specifically, the belief state $\mathbf{b}_t = \begin{bmatrix} \hat{\mathbf{x}}_t \\ \mathrm{vec}[\sqrt{\Sigma_t}] \end{bmatrix}$ is a vector comprising of the mean state $\hat{\mathbf{x}}_t$ and the columns of the square root $\sqrt{\Sigma_t}$ of the covariance Σ_t of a Gaussian distribution $N(\hat{\mathbf{x}}_t, \Sigma_t)$.

We assume that the initial belief $\mathbf{b}_0 = \begin{bmatrix} \hat{\mathbf{x}}_0 \\ \mathrm{vec}[\sqrt{\Sigma_0}] \end{bmatrix}$ is given. Given a current belief $\mathbf{b}_t$, a control input $\mathbf{u}_t$, and a

measurement $\mathbf{z}_{t+1}$, the belief state evolves using a Kalman filter [21], [32]. Prior work on Gaussian belief space planning has used the extended Kalman filter (EKF) [34] but we use an unscented Kalman filter (UKF) [13], which propagates the mean and covariance more accurately for highly nonlinear dynamics and observation models as compared to the EKF.

The UKF uses the unscented transform to compute a set of samples or *sigma points* that characterize the σ-standard deviation contours of the Gaussian distribution [13] for a given value of the parameter σ. Given a Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}, \Sigma)$, $\mathcal{X} = \texttt{SigmaPoints}(\hat{\mathbf{x}}, \sqrt{\Sigma}, \sigma)$ is a set of $(2\dim[\hat{\mathbf{x}}] + 1)$ sigma points given according to:

$$\mathcal{X} = [\hat{\mathbf{x}} \dots \hat{\mathbf{x}}] + \sigma[\mathbf{0} \ \sqrt{\Sigma} \ \ -\sqrt{\Sigma}], \tag{3}$$

where $[\hat{\mathbf{x}} \dots \hat{\mathbf{x}}]$ denote the repeated copies of the mean $\hat{\mathbf{x}}$ concatenated as columns in a matrix. Since the UKF requires computing the square root of the covariance, we directly use the square root of the covariance included in the belief state. We also exploit the symmetry of $\sqrt{\Sigma_t}$ in our implementation to eliminate the redundancy.

We refer the reader to [13] for the details of the UKF, but on a high level the propagation is as follows, given $\hat{\mathbf{x}}_t$ and Σ_t that characterize the current belief:

$$\mathcal{X}_t = \texttt{SigmaPoints}(\hat{\mathbf{x}}_t, \sqrt{\Sigma_t}, \sigma), \tag{4}$$
$$\mathcal{Q}_t = \texttt{SigmaPoints}(\mathbf{0}, I_{\dim[\mathbf{q}]}, \sigma), \tag{5}$$
$$\mathcal{X}^-_{t+1} = \mathbf{F}(\mathcal{X}_t, \mathbf{u}_t, \mathbf{0}) \cup \mathbf{F}(\hat{\mathbf{x}}_t, \mathbf{u}_t, \mathcal{Q}_t), \tag{6}$$
$$\mathcal{R}_{t+1} = \texttt{SigmaPoints}(\mathbf{0}, I_{\dim[\mathbf{r}]}, \sigma), \tag{7}$$
$$\mathcal{Z}_{t+1} = \mathbf{H}(\mathcal{X}^-_{t+1}, \mathbf{0}) \cup \mathbf{H}(\mathrm{E}[\mathcal{X}^-_{t+1}], \mathcal{R}_{t+1}), \tag{8}$$

where $\mathrm{E}[\cdot]$ refers to the sample mean of a set of samples, and $\mathbf{F}(\cdot)$ and $\mathbf{H}(\cdot)$ refer to the sets of results of the function $\mathbf{f}$ and $\mathbf{h}$ applied to each element of the operand, respectively. The new belief is then computed according to:

$$\hat{\mathbf{x}}_{t+1} = \mathrm{E}[\mathcal{X}^-_{t+1}] + K_t(\mathbf{z}_{t+1} - \mathrm{E}[\mathcal{Z}_{t+1}]), \tag{9}$$
$$\sqrt{\Sigma_{t+1}} = \sqrt{\mathrm{Var}[\mathcal{X}^-_{t+1}] - K_t \, \mathrm{Var}[\mathcal{Z}_{t+1}] K_t^T}, \tag{10}$$

where $K_t = \mathrm{Cov}[\mathcal{X}^-_{t+1}, \mathcal{Z}_{t+1}] \, \mathrm{Var}[\mathcal{Z}_{t+1}]^{-1}$, and $\mathrm{Var}[\cdot]$ and $\mathrm{Cov}[\cdot, \cdot]$ refer to the sample variance and sample covariance respectively of the respective sets of samples. The mean is computed using $s = |\mathcal{X}^-_{t+1}| = 2(\dim[\mathbf{x}] + \dim[\mathbf{q}]) + 1$ points.

The stochastic belief space dynamics is then given by:

$$\mathbf{b}_{t+1} = \mathbf{g}(\mathbf{b}_t, \mathbf{u}_t) + W(\mathbf{b}_t, \mathbf{u}_t)\mathbf{w}_t, \quad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, I), \tag{11}$$
$$\mathbf{g}(\mathbf{b}_t, \mathbf{u}_t) = \begin{bmatrix} \mathrm{E}[\mathcal{X}^-_{t+1}] \\ \mathrm{vec}[\sqrt{\mathrm{Var}[\mathcal{X}^-_{t+1}] - K_t \, \mathrm{Var}[\mathcal{Z}_{t+1}] K_t^T}] \end{bmatrix}, \tag{12}$$
$$W(\mathbf{b}_t, \mathbf{u}_t) = \begin{bmatrix} \sqrt{K_t \, \mathrm{Var}[\mathcal{Z}_{t+1}] K_t^T} \\ 0 \end{bmatrix}. \tag{13}$$

Inspired by Platt et al. [21], we make the assumption that the maximum likelihood observation is obtained at each time step, i.e., $\mathbf{z}_{t+1} = \mathrm{E}[\mathcal{Z}_{t+1}]$, which eliminates the stochasticity from the belief dynamics. This assumption of determinism is just used for efficiently planning locally optimal trajectories in belief space. We refer to the nominal state obtained under the maximum likelihood observation assumption as $\hat{\mathbf{b}}$ and the nominal control input as $\hat{\mathbf{u}}$. The deterministic dynamics of the nominal belief $\hat{\mathbf{b}}_t$ evolves as:

$$\hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t), \tag{14}$$

where the function $\mathbf{g}$ is as defined in Eq. (12). During execution, we compute the actual belief state $\mathbf{b}_{t+1}$ using the observation $\mathbf{z}_{t+1}$ and re-plan a locally optimal nominal belief space trajectory starting from $\mathbf{b}_{t+1}$, essentially invoking model predictive feedback control (MPC) [7] in belief space.

Objective: To facilitate safe execution of motion plans in Gaussian belief space, our objective is to compute optimal trajectories in belief space that are at least σ-standard deviations safe, where σ is a user-specified parameter. [1] At each step, we formulate the planning problem as a nonlinear optimization problem which minimizes a user-defined cost function encoding the task objective while satisfying all task constraints, as described below.

IV. Trajectory Optimization in Belief Space

We formulate belief space planning as an optimization problem. For notational convenience, we concatenate the belief states and control inputs for all time steps $t \in \mathcal{T}$ to form $\hat{\mathcal{B}} = [\hat{\mathbf{b}}_0 \dots \hat{\mathbf{b}}_T]^T$ and $\hat{\mathcal{U}} = [\hat{\mathbf{u}}_0 \dots \hat{\mathbf{u}}_{T-1}]^T$ that parameterize a nominal belief space trajectory such that $\hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t) \ \forall \ t \in \{0, \dots, T-1\}$. The optimization problem is then formally stated as:

$$\begin{aligned} \min_{\hat{\mathcal{B}}, \hat{\mathcal{U}}} \quad & \mathbf{C}(\hat{\mathcal{B}}, \hat{\mathcal{U}}) \\ \text{s. t.}_{\forall t \in \mathcal{T}} \quad & \boldsymbol{\psi}(\hat{\mathbf{x}}_T) = \boldsymbol{\psi}_{\text{target}}, \\ & \hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t), \\ & \boldsymbol{\Phi}(\hat{\mathcal{B}}, \hat{\mathcal{U}}, \sigma) \geq \mathbf{0}, \\ & \hat{\mathbf{u}}_t \in F_{\mathcal{U}}, \end{aligned} \tag{15}$$

where $\mathbf{C}(\hat{\mathcal{B}}, \hat{\mathcal{U}})$ is a cost function encoding the task objective, $\boldsymbol{\psi}(\hat{\mathbf{x}}_T) = \boldsymbol{\psi}_{\text{target}}$ constrains the robot end effector pose $\boldsymbol{\psi}(\hat{\mathbf{x}}_T)$ at the final time step T to be the desired end effector pose $\boldsymbol{\psi}_{\text{target}}$, and $\hat{\mathbf{u}}_t \in F_{\mathcal{U}}$ constrains the control input $\hat{\mathbf{u}}_t$ to lie in the set of feasible control inputs $F_{\mathcal{U}}$. $\boldsymbol{\Phi}(\hat{\mathcal{B}}, \hat{\mathcal{U}}, \sigma) \geq \mathbf{0}$ enforces that the trajectory is σ-standard deviations safe for probabilistic collision avoidance in belief space. The optimization is initialized with a belief trajectory $\bar{\mathcal{B}} = [\bar{\mathbf{b}}_0 \dots \bar{\mathbf{b}}_T]^T$ and $\bar{\mathcal{U}} = [\bar{\mathbf{u}}_0 \dots \bar{\mathbf{u}}_{T-1}]^T$.

The general form of the optimization problem given in Eq. (15) is challenging because of the highly nonlinear objective and constraints. It is thus difficult to find globally optimal solutions to this problem. We compute locally optimal solutions using sequential convex programming [2], by repeatedly constructing a locally convex approximation to the original problem around the initialization trajectory $(\bar{\mathcal{B}}, \bar{\mathcal{U}})$. We refer the reader to Betts [2] for a detailed exposition of trajectory optimization and Schulman et al. [28] for application of these methods to robot motion planning in state space.

[1] For a multivariate Gaussian distribution of dimension $\dim[\mathbf{x}]$, the probability that a sample is within σ standard deviations is given by $\Gamma(\dim[\mathbf{x}]/2, \sigma/2)$, where Γ is the regularized Gamma function [24].

Costs and Constraints: We describe the costs and constraints in detail below:

1) $\mathbf{C}(\hat{\mathcal{B}},\hat{\mathcal{U}})$: We consider the planning objective of accomplishing the desired task while minimizing uncertainty and control effort. The cost function we use is of the form:

$$\mathbf{C}(\hat{\mathcal{B}},\hat{\mathcal{U}}) = \sum_{t=0}^{T} \mathrm{tr}[M_t \hat{\Sigma}_t] + \sum_{t=0}^{T-1} \hat{\mathbf{u}}_t^T N_t \hat{\mathbf{u}}_t, \qquad (16)$$

where M_t and N_t are positive definite cost matrices $\forall\ t \in \mathcal{T}$ that weigh the contributions of the two cost terms. The term $\mathrm{tr}[M_t\hat{\Sigma}_t]$ penalizes the uncertainty by considering the departure from zero variance and the term $\hat{\mathbf{u}}_t^T N_t \hat{\mathbf{u}}_t$ is a quadratic cost encoding the total control effort.

2) $\hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t)\ \forall\ t \in \{0,\dots,T-1\}$: This enforces the constraint on the nominal belief dynamics given by Eq. (14). The locally convex approximation of the nonlinear equality constraint is obtained by linearizing around the initialization trajectory $(\bar{\mathcal{B}},\bar{\mathcal{U}})$ as:

$$\mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t) \approx \mathbf{g}(\bar{\mathbf{b}}_t, \bar{\mathbf{u}}_t) + G_t(\hat{\mathbf{b}}_t - \bar{\mathbf{b}}_t) + H_t(\hat{\mathbf{u}}_t - \bar{\mathbf{u}}_t) \quad (17)$$

$$G_t = \frac{\partial \mathbf{g}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t, \bar{\mathbf{u}}_t), \quad H_t = \frac{\partial \mathbf{g}}{\partial \hat{\mathbf{u}}}(\bar{\mathbf{b}}_t, \bar{\mathbf{u}}_t), \qquad (18)$$

where G_t and H_t are the Jacobians of the nominal belief dynamics function evaluated (numerically) at $\bar{\mathbf{b}}_t, \bar{\mathbf{u}}_t$.

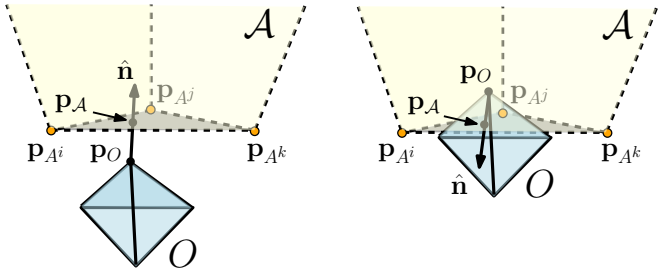
We also add inequality constraints on the belief states $\hat{\mathbf{b}}_t$ and control inputs $\hat{\mathbf{u}}_t$ of the form $\|\hat{\mathbf{b}}_t - \bar{\mathbf{b}}_t\|_2 < \epsilon_{\mathcal{B}}$ and $\|\hat{\mathbf{u}}_t - \bar{\mathbf{u}}_t\|_2 < \epsilon_{\mathcal{U}}$ to ensure that the optimization progresses only within bounds of the region where the locally convex approximation holds.

3) $\hat{\mathbf{u}}_t \in F_{\mathcal{U}}\ \forall\ t \in \{0,\dots,T-1\}$: This constrains the control input to lie within the set of feasible control inputs $F_{\mathcal{U}}$. For instance, control input limits might correspond to the maximum speed with which the base or links might move. We formulate the feasibility constraint as box inequality constraints $\mathbf{u}_{\min} \leq \hat{\mathbf{u}}_t \leq \mathbf{u}_{\max}$ at each time step t corresponding to the minimum $\mathbf{u}_{\min}$ and maximum $\mathbf{u}_{\max}$ control input bounds, respectively. The inequality constraints are treated as (hard) constraints in the optimization [28].

V. Collision Avoidance Constraints

We formulate collision avoidance constraints $\mathbf{\Phi}(\hat{\mathcal{B}},\hat{\mathcal{U}},\sigma) \geq \mathbf{0}$ within the optimization framework to ensure that the optimized trajectory is σ-standard deviations safe. To accomplish this, we consider the sigma hull, which is the convex hull of the robot geometry transformed according to the UKF sigma points $\mathcal{X}$ given by Eq. (6) lying on the σ-standard deviation contour of the covariance.

We use the signed distance [11] between the sigma hull and objects in the workspace $\mathcal{O}$, as shown in Fig. 2. Informally, the signed distance corresponds to the minimum translation distance required to either put two geometric shapes in contact or separate them if they are overlapping. The distance between two shapes can be calculated by the Gilbert-Johnson-Keerthi (GJK) algorithm and the penetration depth is calculated by the Expanding Polytope Algorithm (EPA) [11]. One useful feature of these two algorithms is that they represent any given convex shape A by its support mapping, i.e., a function that maps a vector $\mathbf{v}$ to the point in A that is furthest in direction $\mathbf{v}$:



(a) $\mathtt{sd}(\mathcal{A}, O) > 0$ (b) $\mathtt{sd}(\mathcal{A}, O) < 0$

Fig. 2. Signed distance between a 3D pyramidal object O in the workspace and a 3D convex hull $\mathcal{A}$ obtained by transforming the robot link geometry using the UKF sigma points. $\mathbf{p}_{\mathcal{A}}$ and $\mathbf{p}_O$ are the points of closest approach or penetration. Note that the convex hull of the geometry corresponding to the sigma points is never explicitly constructed. Instead, we rely on the concept of support mapping to formulate collision avoidance constraints. (a) The signed distance is the positive minimum distance between $\mathcal{A}$ and O and corresponds to the length of the smallest translation that puts the shapes into contact. (b) The signed distance is the negative of the penetration depth and corresponds to the minimum translation length that separates $\mathcal{A}$ and O.

$$\mathtt{support}_A(\mathbf{v}) = \arg\max_{\mathbf{x} \in A} \mathbf{v} \cdot \mathbf{x}. \qquad (19)$$

We consider the sigma hull of each individual link in the robot by transforming the geometry of each link according to the sigma points. [2] For a K link articulated robot and $|\mathcal{O}|$ objects in the workspace, we formulate the collision avoidance constraints for each of the $|\mathcal{O}|K$ link-object pairs at each time step. We ignore self-collisions between robot links or collisions between pairs of objects in the workspace and assume that the objects $\mathcal{O}$ in the workspace are precisely known. We also assume that all the objects $\mathcal{O}$ are convex. In case of non-convex obstacles, we use standard convex decomposition techniques to decompose non-convex geometries into convex geometries.

For a given link A, let $\mathcal{A} = \mathtt{convhull}\{A^0,\dots,A^s\}$ be the sigma hull obtained by transforming the link geometry according to the s UKF sigma points (Eq. (3)). It turns out that we do not have to explicitly compute a geometrical representation of the convex hull $\mathcal{A}$ to perform the signed distance computation, since the signed distance can be calculated using the support mapping function (Eq. (19)) as:

$$\mathtt{support}_{\mathcal{A}}(\mathbf{v}) = \mathtt{support}_{A^i}(\mathbf{v}), \qquad (20)$$

where

$$i = \arg\max_{i \in \{0,\dots,s\}} \mathtt{support}_{A^i}(\mathbf{v}) \cdot \mathbf{v}. \qquad (21)$$

We use the above definition to compute the signed distance of $\mathcal{A}$ with respect to objects $\mathcal{O}$ in the workspace by defining the support mapping function for the transformed link geometries $\{A^0,\dots,A^s\}$. We use efficient implementations of

[2]For an articulated robot, we obtain the transformed geometries of all the links in a single pass by starting from the last link that contains the end effector. Since the UKF sigma points corresponding to the last link contain all the DOF of the kinematic chain, we perform all the necessary forward kinematics evaluations only once and cache the sigma shapes for all the remaining links as the computation is being performed for the last link.

the GJK and EPA algorithms available in the Bullet collision checking library [1] for this purpose.

Fig. 2 shows the signed distance between an object $O \in \mathcal{O}$ and the convex hull $\mathcal{A}$. Two objects are out of collision if the signed distance $\mathtt{sd}(\mathcal{A}, O)$ is positive. We enforce the following collision avoidance constraint at each time step:

$$\mathtt{sd}(\mathcal{A}, O) \geq d_{\text{safe}} \quad \forall\ O \in \mathcal{O}, \tag{22}$$

where a user-specified safety margin $d_{\text{safe}} > 0$ is used to ensure that the robot stays at least d_{safe} distance from other objects to avoid being in contact (corresponding to $d_{\text{safe}} = 0$).

For the purposes of optimization, the nonlinear constraint in Eq. (22) is approximated by linearized inequality constraints obtained by linearizing around the initialization trajectory $(\bar{\mathcal{B}}, \bar{\mathcal{U}})$. The signed distance $\mathtt{sd}(\mathcal{A}, O)$ is a function of the belief state $\hat{\mathbf{b}}_t$ and is denoted as $\mathtt{sd}_{\mathcal{A}O}(\hat{\mathbf{b}}_t)$. We denote the closest points on $\mathcal{A}$ and O that realize the signed distance as $\mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t)$ and $\mathbf{p}_O$, respectively, and the contact normal as the unit vector $\hat{\mathbf{n}}(\hat{\mathbf{b}}_t)$. We approximate the signed distance between $\mathcal{A}$ and O at a belief state $\hat{\mathbf{b}}_t$ in the neighborhood of the current belief $\bar{\mathbf{b}}_t$ as:

$$\mathtt{sd}_{\mathcal{A}O}(\hat{\mathbf{b}}_t) \approx \hat{\mathbf{n}}(\bar{\mathbf{b}}_t) \cdot (\mathbf{p}_O - \mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t)), \tag{23}$$

$$\hat{\mathbf{n}}(\bar{\mathbf{b}}_t) = \frac{(\mathbf{p}_O - \mathbf{p}_{\mathcal{A}}(\bar{\mathbf{b}}_t))}{\|\mathbf{p}_O - \mathbf{p}_{\mathcal{A}}(\bar{\mathbf{b}}_t)\|_2}, \tag{24}$$

where we assume that the contact normal does not change in the local neighborhood of $\bar{\mathbf{b}}_t$. We now linearize the expression for signed distance $\mathtt{sd}_{\mathcal{A}O}(\hat{\mathbf{b}}_t)$ given by Eq. 23 at the current belief $\bar{\mathbf{b}}_t$ as:

$$\mathtt{sd}_{\mathcal{A}O}(\hat{\mathbf{b}}_t) \approx \mathtt{sd}_{\mathcal{A}O}(\bar{\mathbf{b}}_t) + S_t(\hat{\mathbf{b}}_t - \bar{\mathbf{b}}_t), \tag{25}$$

$$S_t = \frac{\partial \mathtt{sd}_{\mathcal{A}O}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t) \approx -\hat{\mathbf{n}}(\bar{\mathbf{b}}_t)^T \frac{\partial \mathbf{p}_{\mathcal{A}}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t). \tag{26}$$

The closest point $\mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t)$ lies at a vertex, edge, or face of the convex hull $\mathcal{A}$, the vertices of which come from the transformed link geometries $\{A^0, \ldots, A^s\}$. Since collision checking is expensive even while planning in the state space [28], we compute the derivative S_t analytically instead of relying on expensive numerical derivatives as follows:

We consider the most general case where the closest point $\mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t)$ lies on a face of the convex hull (as shown in Fig. 2). With small notational modifications, our analysis holds for general simplices but we consider faces to be triangles since triangles meshes are the most popular representations of geometry in collision libraries [1]. Without loss of generality, we assume that face is spanned by three vertices $\mathbf{p}_{A^i}$, $\mathbf{p}_{A^j}$, and $\mathbf{p}_{A^k}$ and that the three vertices come from three distinct instances of geometry A^i, A^j, and A^k corresponding to the i, j, and k^{th} UKF sigma points, respectively. This information is made available to us from the collision library when the signed distance computation is performed. We represent $\mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t)$ as a barycentric combination of the vertices as:

$$\mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t) = \sum_{l \in \{i,j,k\}} \alpha_l \mathbf{p}_{A^l}, \sum_{l \in \{i,j,k\}} \alpha_l = 1, \tag{27}$$

where $\alpha_l, l \in \{i, j, k\}$ are the barycentric coordinates of the point $\mathbf{p}_{\mathcal{A}}(\hat{\mathbf{b}}_t)$. The derivative term from Eq. (26) can now be written as:

$$\frac{\partial \mathbf{p}_{\mathcal{A}}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t) = \sum_{l \in \{i,j,k\}} \alpha_l \frac{\partial \mathbf{p}_{A^l}}{\partial \hat{\mathbf{b}}}(\bar{\mathbf{b}}_t). \tag{28}$$

Each point $\mathbf{p}_{A^l}$ is a function of the l^{th} sigma point given by $\mathbf{p}_{A^l}(\mathcal{X}^l)$. Each $\mathcal{X}^l$ is defined as a linear combination of $\hat{\mathbf{x}}_t$ and a corresponding column of $\sqrt{\hat{\Sigma}_t}$ as given in Eq. (3), which constitute the belief state $\hat{\mathbf{b}}_t$. The derivatives in Eq. (28) can now be evaluated analytically in terms of the position Jacobian matrices $J_{\mathbf{p}_{A^l}}$ that relate the point $\mathbf{p}_{A^l}$ to the current state $\bar{\mathbf{x}}_t$.

Using Eqs. (22), (25), (26), and (28), we include the collision avoidance constraint in the optimization formulation (Eq. (15)) in terms of the belief state $\hat{\mathbf{b}}_t$ as:

$$\mathtt{sd}_{\mathcal{A}O}(\bar{\mathbf{b}}_t) + S_t(\hat{\mathbf{b}}_t - \bar{\mathbf{b}}_t) - d_{\text{safe}} \geq 0,\ \forall\ O \in \mathcal{O}. \tag{29}$$

VI. Results

We evaluate our approach in simulation for planning motions for a 7-DOF articulated robot with imprecise actuation and equipped with cheap, inaccurate sensors. The state $\mathbf{x} = (\theta_1, \ldots, \theta_7)$ of the robot is a 7-D vector consisting of the joint angles and the control input $\mathbf{u} = (\omega_1, \ldots, \omega_7)$ is a 7-D vector consisting of the angular speeds at each of the joints. The motion noise $\mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, I_{\dim[\mathbf{q}]})$ is scaled by a constant matrix Q. This results in the following stochastic model:

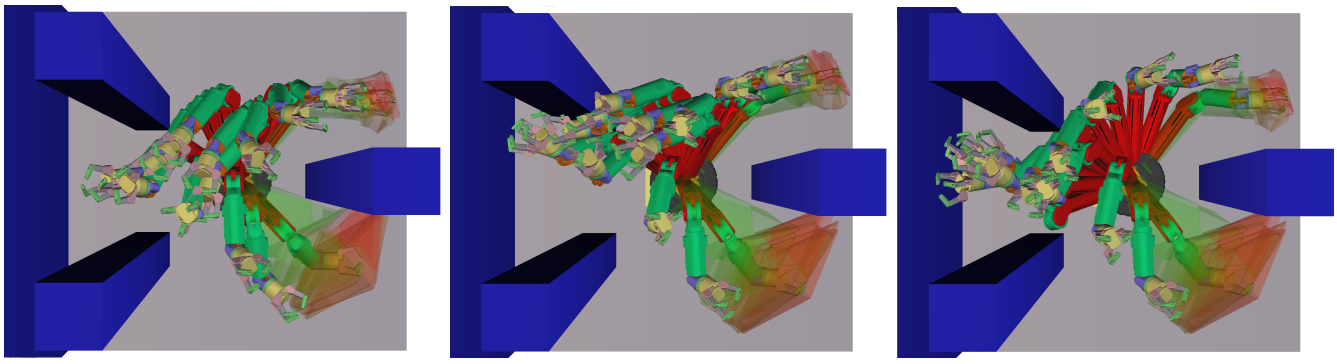
$$\mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{q}_t) = \begin{bmatrix} \theta_1 + \tau\omega_1 \\ \vdots \\ \theta_7 + \tau\omega_7 \end{bmatrix} + Q\mathbf{q}_t, \tag{30}$$

where τ is the duration of the time step.

We consider a scenario where the robot is equipped with an inaccurate distance sensor mounted at the end-effector. The signal strength measured by the sensor decays quadratically with the distance to a given object. This is similar in spirit to the light-dark example considered by Platt et al. [21] where the robot obtains reliable sensing information in a certain region of the environment but the sensor measurements get noisier as the robot moves away from the region. In this scenario, the sensor measures the signal strength in terms of the distance to a wall, which is parallel to the y-z plane and has a known location $x = x_{\text{wall}}$ (Fig. 3(a)). Let $\mathbf{p}_e(\mathbf{x}) = [x_e, y_e, z_e]^T$ be the position of the robot end-effector in the world coordinate frame. The robot also measures the angle at the base using an inaccurate encoder. The sensing noise $\mathbf{r}_t \sim \mathcal{N}(\mathbf{0}, I_{\dim[\mathbf{r}]})$ is scaled by a constant matrix R. The stochastic sensing measurements are then related to the state $\mathbf{x}_t$ as:

$$\mathbf{h}(\mathbf{x}_t, \mathbf{r}_t) = \begin{bmatrix} 1/((x_e - x_{\text{wall}})^2 + 1) \\ \theta_1 \end{bmatrix} + R\mathbf{r}_t. \tag{31}$$

Evaluation: A naïve collision free trajectory computed using an uncertainty unaware planner like rapidly exploring random trees (RRT) [18] avoids obstacles (Fig. 1(a)) but is oblivious to the uncertainty in the robot state and might



(a) Initial optimal trajectory (b) Open-loop execution (c) Execution with re-planning (MPC)

Fig. 3. Simulated trajectories computed using our Gaussian belief space planning framework for a 7-DOF articulated robot. (a) A locally optimal trajectory in belief space that follows nominal belief dynamics computed using our framework. Notice how the robot moves closer to the wall through a narrow passage to reliably localize itself before moving to the desired target. (b) Open-loop execution of the nominal belief space trajectory leads to collisions with objects in the workspace because the optimal trajectory is computed assuming deterministic belief dynamics (Eq. (14)). The end-effector is also farther away from the desired target. (c) We initialize the optimization with the previously computed solution and the current belief state to re-plan after every time step. The resulting trajectory obtained using this strategy is able to successfully avoid collisions during execution. Also note how the trajectory moves the robot end-effector closer to the wall for an extended period of time for reliable localization before safely leading the robot back to the target.

accumulate considerable uncertainty during execution. We show that belief space planning improves the input trajectory to compute a locally optimal trajectory that is able to safely guide the robot towards the wall for reliable localization before heading back to the target (Fig. 3(a)).

Open-loop execution of the locally optimal belief space trajectory leads to collisions with objects in the workspace because the optimal trajectory is computed assuming deterministic belief dynamics (Eq. (14)).) The end-effector is also farther away from the desired target because of the uncertainty accumulated during execution (Fig. 3(b)).

We follow the MPC paradigm of re-planning after every time-step to correct perturbations as they occur during execution. Given a locally optimal trajectory $(\hat{\mathcal{B}}_{t:T}, \hat{\mathcal{U}}_{t:T})$ at time step t, we execute the first control input $\mathbf{u}_t$, obtain the observation $\mathbf{z}_{t+1}$, compute the new belief state $\mathbf{b}_{t+1}$ according to Eqs. (9) and (10). We initialize the optimization with the previous trajectory and the new belief $\mathbf{b}_{t+1}$. Fig. 3(c) shows the resulting trajectory obtained using re-planning, which is able to successfully avoid collisions during execution. Note how the trajectory moves the robot end-effector closer to the wall for an extended period of time for reliable localization before safely leading the robot back to the target.

Our unoptimized C++ implementation running on a single 3.2 GHz Intel i7 processor core took 3 seconds to plan a locally optimal initial trajectory in a 35 dimensional belief space. During execution, the optimization converges faster because it is initialized with a locally optimal trajectory. Our example took 20 seconds to simulate the execution of a trajectory with 30 time steps.

We evaluate our approach by analyzing the effect of open-loop execution and re-planning on the probability of collision of the motion plan, executed a 100 times with simulated Gaussian noise. Open-loop execution results in a 83% probability of collision, which is expected since the initial locally optimal trajectory does not take the stochastic belief dynamics into account. In contrast, re-planning significantly reduces the probability of collision to 6%. Our experiments suggest that re-planning in belief space can significantly increase the probability of successful execution.

VII. Conclusion and Future Work

We presented a novel formulation for collision avoidance for planning in Gaussian belief spaces for articulated robots that are not well approximated as points or spheres. The key insight is to use sigma hulls, which are convex hulls of the robot geometry transformed according to the sigma points generated by the Unscented Kalman filter (UKF). We use efficient formulations of the support mapping of convex shapes to formulate collision avoidance constraints in terms of the signed distance between the sigma hulls and objects in the workspace within an optimization-based planning framework. We have obtained promising results for planning motions for a 7-DOF articulated robot with imprecise actuation and inaccurate sensing. Our experiments suggest that re-planning in belief space can considerably reduce the probability of collision during execution.

Our work opens up several avenues for future research. Currently, the computed trajectory is σ-standard deviations safe and the parameter σ is specified by the user. We plan to incorporate σ in the optimization formulation to search over σ values for an optimal solution that is both safe and minimizes the cost function encoding the task objective. We also assume a Gaussian parameterization of the belief state, which might not be applicable for some applications, for instance ones where multi-modal beliefs are expected to appear. We are currently working on extending this framework to handle self-collisions between the robot links and collisions with other uncertain objects in the workspace.

We envision that our framework would provide efficient solutions for a variety of planning problems under uncertainty that can be modeled as POMDPs. For instance, co-located range sensors on articulated robots such as cameras on surgical robots such as the Raven [27] introduce

complications in planning because the actions of the robot have to be coordinated with sensing to maximize information gain [15]. This would be an important step towards enabling low-cost robots to safely and robustly complete navigation and manipulation tasks in unstructured environments while operating under considerable uncertainty.

REFERENCES

[1] Advanced Micro Devices Inc., "Bullet Collision Detection and Physics Library," Available: http://bulletphysics.org, 2013.

[2] J. T. Betts, *Practical Methods for Optimal Control and Estimation using Nonlinear Programming*. Society for Industrial & Applied Mathematics, 2010, vol. 19.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[4] A. Brooks, A. Makarenko, S. Williams, and H. Durrant-Whyte, "Parametric POMDPs for Planning in Continuous State Spaces," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 887–897, 2006.

[5] A. Bry and N. Roy, "Rapidly-exploring Random Belief Trees for Motion Planning Under Uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 723–730.

[6] J. T. C. Papadimitriou, "The Complexity of Markov Decision Processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, 1987.

[7] E. F. Camacho and C. Bordons, *Model Predictive Control*. London, UK: Springer Verlag, 2004.

[8] A. D. Christiansen and K. Goldberg, "Comparing Two Algorithms for Automatic Planning by Robots in Stochastic Environments," *Robotica*, vol. 13, no. 6, pp. 565–574, 1995.

[9] N. E. du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Trans. Robotics*, vol. 27, pp. 809–815, 2011.

[10] T. Erez and W. D. Smart, "A Scalable Method for Solving High-Dimensional Continuous POMDPs Using Local Approximation," in *Conf. on Uncertainty in Artificial Intelligence*, 2010, pp. 160–167.

[11] C. Ericson, *Real-Time Collision Detection*. Morgan Kaufmann, 2004.

[12] K. Hauser, "Randomized Belief-Space Replanning in Partially Observable Continuous Spaces," *Algorithmic Foundations of Robotics IX*, pp. 193–209, 2011.

[13] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[14] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains," *Artificial Intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.

[15] L. P. Kaelbling and T. Lozano-Perez, "Integrated Task and Motion Planning in Belief Space," *Submitted. Draft at http://lis.csail.mit.edu/pubs/tlp/IJRRBelFinal.pdf*, 2012.

[16] H. Kurniawati, D. Hsu, and W. S. Lee, "SARSOP: Efficient Point-based POMDP Planning by Approximating Optimally Reachable Belief Spaces," in *Robotics: Science and Systems (RSS)*, 2008.

[17] A. Lambert, D. Gruyer, and G. St Pierre, "A Fast Monte Carlo Algorithm for Collision Probability Estimation," in *Intl. Conf. on Control, Automation, Robotics and Vision*, 2008, pp. 406–411.

[18] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, Available at http://planning.cs.uiuc.edu.

[19] S. A. Miller, Z. A. Harris, and E. K. P. Chong, "Coordinated Guidance of Autonomous UAVs via Nominal Belief-State Optimization," in *American Control Conference (ACC)*, 2009, pp. 2811–2818.

[20] R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake, "Efficient Planning in Non-Gaussian Belief Spaces and its Application to Robot Grasping," in *Int. Symp. on Robotics Research (ISRR)*, 2011.

[21] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief Space Planning assuming Maximum Likelihood Observations," in *Robotics: Science and Systems (RSS)*, 2010.

[22] J. Porta, N. Vlassis, M. Spaan, and P. Poupart, "Point-based Value Iteration for Continuous POMDPs," *Journal of Machine Learning Research*, vol. 7, pp. 2329–2367, 2006.

[23] S. Prentice and N. Roy, "The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance," *Int. Journal of Robotics Research*, vol. 28, no. 11–12, pp. 1448–1465, 2009.

[24] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[25] M. Quigley, R. Brewer, S. P. Soundararaj, V. Pradeep, Q. Le, and A. Y. Ng, "Low-cost Accelerometers for Robotic Manipulator Perception," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 6168–6174.

[26] Rethink Robotics, "Baxter manufacturing robot," Available: http://www.rethinkrobotics.com/index.php/products/baxter/, 2012.

[27] J. Rosen, M. Lum, M. Sinanan, and B. Hannaford, "Raven: Developing a Surgical Robot from a Concept to a Transatlantic Teleoperation Experiment," in *Surgical Robotics: System Applications and Visions*, J. Rosen, B. Hannaford, and R. M. Satava, Eds. Springer, 2011, ch. 8, pp. 159–197.

[28] J. Schulman, A. Lee, H. Bradlow, I. Awwal, and P. Abbeel, "Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization," *Submitted. Draft at https://sites.google.com/site/rss2013trajopt/*, 2013.

[29] D. Silver and J. Veness, "Monte-Carlo Planning in Large POMDPs," in *Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 2164–2172.

[30] S. Thrun, "Monte Carlo POMDPs," *Advances in Neural Information Processing Systems*, vol. 12, pp. 1064–1070, 2000.

[31] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized Path Planning for Robots with Motion Uncertainty and Imperfect State Information," *Int. Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[32] J. van den Berg, S. Patil, and R. Alterovitz, "Motion Planning under Uncertainty using Iterative Local Optimization in Belief Space," *Int. Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.

[33] M. P. Vitus and C. J. Tomlin, "Closed-Loop Belief Space Planning for Linear, Gaussian Systems," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2011, pp. 2152–2159.

[34] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Univ. North Carolina at Chapel Hill, Tech. Rep. TR 95-041, 2006.