Gaussian Belief Space Planning with Discontinuities in Sensing Domains

Sachin Patil, Yan Duan, John Schulman, Ken Goldberg, Pieter Abbeel

Abstract— **Discontinuities in sensing domains are common when planning for many robotic navigation and manipulation tasks. For cameras and 3D sensors, discontinuities may be inherent in sensor field of view or may change over time due to occlusions that are created by moving obstructions and movements of the sensor. The associated gaps in sensor information due to missing measurements pose a challenge for belief space and related optimization-based planning methods since there is no gradient information when the system state is outside the sensing domain. We address this in a belief space context by considering the signed distance to the sensing region. We smooth out sensing discontinuities by assuming that measurements can be obtained outside the sensing region with noise levels depending on a sigmoid function of the signed distance. We sequentially improve the continuous approximation by increasing the sigmoid slope over an outer loop to find plans that cope with sensor discontinuities. We also incorporate the information contained in not obtaining a measurement about the state during execution by appropriately truncating the Gaussian belief state. We present results in simulation for tasks with uncertainty involving navigation of mobile robots and reaching tasks with planar robot arms. Experiments suggest that the approach can be used to cope with discontinuities in sensing domains by effectively re-planning during execution.**

I. INTRODUCTION

Our work is motivated by the desire to facilitate reliable, autonomous robotic manipulation and navigation tasks. For autonomous operation, robots use sensors to obtain measurements about their state and the state of the environment they are operating in. Commonly used sensors such as cameras, laser range finders, and depth sensors have a limited range of sensing, outside of which no measurements or noisy, unusable measurements are obtained. Occlusions due to robot geometry or objects in the environment may also be responsible for missing measurements. These factors induce discontinuities in sensing domains that can introduce considerable uncertainty during autonomous task execution.

Consider, for instance, a robot, such as Willow Garage's PR2 robot, that is equipped with an on-board camera and depth sensor and is trying to autonomously perform a grasping and manipulation task. The limited field of view of these sensors and occlusions due to the robot arm obstructing the view while trying to grasp an object lead to missing measurements that can introduce uncertainty in task execution. A common workaround is to carefully engineer how the sensors are positioned and how the arms are moved such that the object is in the field of view of the sensors. Ideally, however, the robot itself should be capable of planning about its sensor
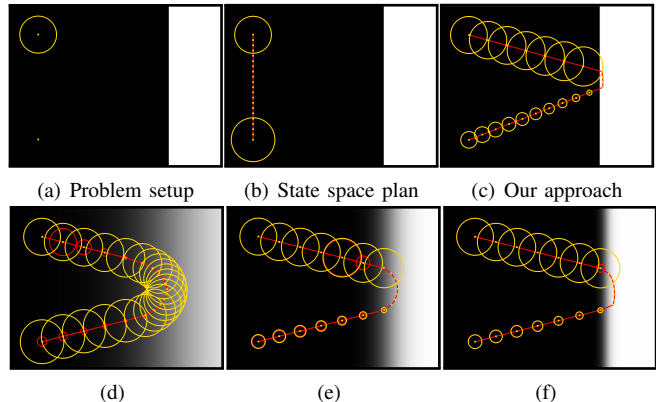
Fig. 1. **Belief space planning with sensing discontinuity**: Comparison of plans generated by state space and belief space methods for a point robot moving in a 2D light-dark domain adapted from Platt et al. [18] (Sec. VI-A). (a) The robot's start position (upper-left corner), initial uncertainty (ellipse), and desired target (bottom left corner) are shown in yellow. The robot only obtains a measurement of its position when it is in the sensing domain (the white rectangular region on the right hand side). (b) State-space planning [23] generates a locally-optimal trajectory straight toward the target, unaware of the sensing zone, so cannot reduce uncertainty (initial and final uncertainty shown). (c-f) Belief space planning using our approach computes a trajectory that first visits the sensing domain to reduce uncertainty before headings towards the target. To address the discontinuity in sensing domain, our algorithm solves a sequence of optimization problems that begin with (d) a very relaxed (continuous) sigmoid approximation to the step function, gradually reducing the relaxation (e,f) toward the actual (discontinuous) step function. The resulting plan shown in (f) is more robust than the plan in (b) or (d), as detailed in Sec. VI. The beliefs as computed using the continuous approximation (red) and the true beliefs (yellow), are shown.

and arm motions in a way that it can succeed even in the presence of uncertainty.

We address the problem of computing locally-optimal trajectories in belief space, given that measurements may not be obtained during execution due to factors such as limited field of view of sensors and occlusions. We adopt a trajectory optimization framework for belief space planning, which has shown promise in terms of scaling to higher dimensional belief spaces [3], [23], [15]. Incorporating sensing discontinuities in this optimization framework is challenging since when no measurements are obtained, there is no gradient information to guide the optimization to regions where valid measurements could be obtained.

To handle the discrete transition between a sensor providing a measurement versus not providing a measurement, we solve a sequence of optimization problems. Each problem is a continuous sigmoid approximation of the discontinuities in the sensing domain [6], and we sequentially improve the approximation till the true sensing discontinuity is represented within a user-defined tolerance (Fig. 1). This approach of continuously deforming an easier problem into the given hard

problem, while solving a family of deformed problems, is referred to as an homotopy approach in optimization [29].

Our proposed approach handles limited sensing regions and occlusions within the same framework by considering the signed distance to the valid sensing region. Valid measurements are obtained if the expected state of the robot or other objects is within the valid sensing region, thus corresponding to a negative signed distance, and obtaining no measurements corresponds to a positive signed distance. It is important to note that since measurements obtained during execution are not known a priori, we revise the plan according to the revised expected state of the system using the model predictive control (MPC) paradigm [21] of re-planning after every time step. During execution, there is information contained in not obtaining a measurement about the state, which is incorporated by appropriately truncating the Gaussian belief state with respect to the boundary of the valid sensing region [17].

We present results in simulation for tasks involving navigation of mobile robots in environments with limited sensing regions and for a planar articulated arm performing a reaching task under uncertainty. Our experiments suggest that this approach can be used to re-plan for model predictive control to cope with discontinuities in sensing domains.

II. RELATED WORK

Research efforts have focused on solving POMDPs involving discrete state, action, and/or observation spaces using approximate value iteration [19], [14] but these approaches do not currently scale to continuous state, action, and observation spaces that arise in robotic manipulation and navigation. Recently, extensions have been proposed to continuous spaces [1] but these methods may not be suitable for online planning. Kaelbling et al. [12] proposed an integrated hierarchical planning framework that combines discrete and continuous planning by relying upon a regression-based planner. Factors such as the limited field of view of sensors and occlusions are encoded in this framework using user-defined task primitives.

As a result, a large subset of prior work has focused on computing locally optimal solutions for problems involving robot navigation and manipulation using sampling-based motion planners or optimization-based methods. Sampling-based planners [20], [4], [9], [10] can directly incorporate parametric and nonparametric belief representations and discontinuities in sensing domains due to limited sensor field of view or occlusions but these methods do not currently scale to higher dimensional belief spaces. Optimization-based methods [8], [18], [27], [15] use a parametric Gaussian belief state representation and use continuous dynamics and measurement models that provide gradient information to the optimization. We consider an extension to optimization-based methods to handle sensing discontinuities and occlusions by iteratively solving a series of problems that better approximate the true problem.

Vitus and Tomlin [28] considered the problem of placing sensors in the environment to minimize the uncertainty in the state of the robot along a pre-planned trajectory using numerical optimization. We use a similar relaxation method as theirs to incorporate sensing discontinuities in a numerical optimization framework. In our work, we use a belief space planning formulation that could be used to plan locally-optimal trajectories for both the robot and deployed sensors. We also consider heterogeneous sensors and anisotropic sensing regions induced due to factors such as the limited field of view of sensors or occlusions.

Prior work in the domain of active perception and sensing [2] has considered the effect of discontinuities in sensing domains in decision making under uncertainty. We refer the reader to an extensive survey of developments in this field by Chen et al. [7]. Active perception addresses the question of where to position sensors and how to make decisions for next actions, in order to maximize information gain and minimize cost. Most approaches in this domain assume that the location of the sensors is known with certainty. The belief space formulation used in this work is capable of jointly planning motions for the robot and sensors under uncertainty.

The domain of active exploration considers motion planning for a mobile robot to minimize uncertainty in the location of sensed objects in the environment. Proposed approaches relax the stochastic planning problem to make it computationally tractable. For instance, a small set of actions and one-step lookahead (myopic) planning is considered [25], a reinforcement learning approach based on a discretized state and action space is evaluated [13], and active policy search for approximately solving the underlying POMDP has also been explored [16]. In contrast, our work uses a belief space planning formulation that optimizes over the entire belief state.

III. PRELIMINARIES AND OBJECTIVE

State space system: Let $\mathbf{x} = \left[\begin{smallmatrix} \mathbf{x}_R \\ \mathbf{x}_{\mathcal{O}} \end{smallmatrix} \right]$ be the system state that is comprised of the state $\mathbf{x}_R$ of the robot and the state $\mathbf{x}_{\mathcal{O}}$ of other relevant objects in the environment. Let $\mathbf{u} = \left[\begin{smallmatrix} \mathbf{u}_R \\ \mathbf{u}_{\mathcal{O}} \end{smallmatrix} \right]$ denote the combined control input applied to the system and $\mathbf{z} = \left[\begin{smallmatrix} \mathbf{z}_R \\ \mathbf{z}_{\mathcal{O}} \end{smallmatrix} \right]$ be the vector of measurements obtained about the system state using sensors.

We assume we are given a discrete-time stochastic dynamics model that describes how the system state evolves and a measurement model that relates the obtained measurements to the state, given here in state-transition notation:

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t, \mathbf{q}_t), \quad \mathbf{q}_t \sim \mathcal{N}(\mathbf{0}, I), \qquad (1)$$

$$\mathbf{z}_t = \mathbf{h}(\mathbf{x}_t, \mathbf{r}_t), \qquad \mathbf{r}_t \sim \mathcal{N}(\mathbf{0}, I), \qquad (2)$$

where we assume that the state space trajectory is discretized into ℓ time intervals of equal duration, $\mathbf{q}_t$ is the process noise and $\mathbf{r}_t$ is the measurement noise that is assumed to be drawn, without loss of generality, from a Gaussian distribution 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.

Due to discontinuities in the sensing domain, some of the dimensions of the measurement vector $\mathbf{z}_t$ at time step t cannot be measured using sensors. To quantify this phenomenon,

we define a binary vector $\boldsymbol{\delta}_t \in \mathbb{R}^{\dim[\mathbf{z}]}$, where the ith entry in the vector δ_t^i takes the value 1 if a measurement of the ith dimension of $\mathbf{z}_t$ is obtained at time step t and a value of 0 if no measurement is obtained due to discontinuities in the sensing domain. We formalize the relationship between the δ_t^i variables and valid sensing domains in which measurements can be obtained, in Sec. IV.

Belief system: We consider a Gaussian parameterization of the probability distribution over the state, also known as the belief. Specifically, the belief state $\mathbf{b}_t = \left[\begin{smallmatrix} \mathbf{x}_t \\ \text{vec}[\sqrt{\Sigma_t}] \end{smallmatrix} \right]$ is a vector comprised of the mean state $\mathbf{x}_t$ and the columns of the principal square root $\sqrt{\Sigma_t}$ of the covariance Σ_t of a Gaussian distribution $N(\mathbf{x}_t, \Sigma_t)$. We only include the lower (or equivalently, upper) triangular entries of $\sqrt{\Sigma_t}$ to eliminate redundancy in the belief state.

We assume that the initial belief $\mathbf{b}_0 = \left[\begin{smallmatrix} \mathbf{x}_0 \\ \text{vec}[\sqrt{\Sigma_0}] \end{smallmatrix} \right]$ 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 (such as an extended Kalman filter) and is a stochastic process [27]. However, inspired by Platt et al. [18], we make the assumption that the maximum likelihood observation is obtained at each time step. This eliminates the stochasticity from the belief dynamics to make it suitable for trajectory optimization using sequential quadratic programming (SQP) [3], [23]. The deterministic belief dynamics are given by:

$$\hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t) = \begin{bmatrix} \hat{\mathbf{x}}_{t+1} \\ \text{vec}[\sqrt{\Sigma_{t+1}^- - K_t H_t \Sigma_{t+1}^-}] \end{bmatrix}, \quad (3a)$$

$$\hat{\mathbf{x}}_{t+1} = \mathbf{f}(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t, \mathbf{0}), \quad \Sigma_{t+1}^- = A_t \sqrt{\Sigma_t}(A_t \sqrt{\Sigma_t})^T + Q_t Q_t^T, \quad (3b)$$

$$A_t = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t, \mathbf{0}), \qquad Q_t = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}(\hat{\mathbf{x}}_t, \hat{\mathbf{u}}_t, \mathbf{0}), \quad (3c)$$

$$H_t = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\hat{\mathbf{x}}_{t+1}, \mathbf{0}), \qquad R_t = \frac{\partial \mathbf{h}}{\partial \mathbf{r}}(\hat{\mathbf{x}}_{t+1}, \mathbf{0}), \quad (3d)$$

$$K_t = \Sigma_{t+1}^- H_t^T \Delta_{t+1} (\Delta_{t+1} H_t \Sigma_{t+1}^- H_t^T \Delta_{t+1} + R_t R_t^T)^{-1} \Delta_{t+1}. \quad (3e)$$

Here, $\Delta_{t+1} = \text{diag}[\boldsymbol{\delta}_{t+1}^T]$ is a diagonal matrix comprised of entries of the binary vector $\boldsymbol{\delta}_{t+1}$ corresponding to which dimensions of the maximum likelihood observation $\mathbf{z}_{t+1} = \mathbf{h}(\hat{\mathbf{x}}_{t+1}, \mathbf{0})$ can be measured. Note that the Kalman gain matrix in Eq. (3e) differs from the standard EKF formulation [26] and is a generalization of the formulation considered by Sinopoli et. al. [24], where a single binary value is used to indicate if the entire measurement vector $\mathbf{z}$ is obtained or not [1]. An alternative formulation using an extended information filter (EIF) [26] can also be used to incorporate entries of the binary vector in the update of the belief state [28].

Objective: Our goal is to plan trajectories in belief space that will minimize uncertainty during task execution. In general, objectives that are functions of means and covariances can be considered. We formulate this as a nonlinear optimization problem which minimizes a user-defined cost function that encodes minimization of uncertainty in the system state while satisfying task-specific constraints.

For notational convenience, we concatenate the belief states and control inputs for all time steps $0 \leq t \leq \ell$ to form

---

[1] A derivation, omitted due to space constraints, is available here: http://rll.berkeley.edu/beliefopt/

$\hat{\mathcal{B}} = [\hat{\mathbf{b}}_0 \dots \hat{\mathbf{b}}_\ell]$ and $\hat{\mathcal{U}} = [\hat{\mathbf{u}}_0 \dots \hat{\mathbf{u}}_{\ell-1}]$ that parameterize a nominal belief space trajectory such that $\hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t)$. The optimization problem is then formally stated as:

$$\min_{\hat{\mathcal{B}}, \hat{\mathcal{U}}} \quad \mathbf{C}(\hat{\mathcal{B}}, \hat{\mathcal{U}}) \quad (4a)$$

$$\text{s. t.}_{0 \leq t < \ell} \quad \hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t), \quad \hat{\mathbf{x}}_\ell \in \mathcal{X}_{\text{target}},$$
$$\hat{\mathbf{x}}_t \in \mathcal{X}_{\text{feasible}}, \quad \hat{\mathbf{u}}_t \in \mathcal{U}_{\text{feasible}} \quad (4b)$$

where $\mathbf{C}(\hat{\mathcal{B}}, \hat{\mathcal{U}})$ is a cost function encoding the objective, $\hat{\mathbf{x}}_\ell \in \mathcal{X}_{\text{target}}$ constrains the final estimate of the system state $\hat{\mathbf{x}}_\ell$ to lie in the desired target region $\mathcal{X}_{\text{target}}$, $\hat{\mathbf{u}}_t \in \mathcal{U}_{\text{feasible}}$ constrains the control input $\hat{\mathbf{u}}_t$ to lie in the set of feasible control inputs $\mathcal{U}_{\text{feasible}}$, and $\hat{\mathbf{x}}_t \in \mathcal{X}_{\text{feasible}}$ constrains the state $\hat{\mathbf{x}}_t$ to lie in the set of feasible states $\mathcal{X}_{\text{feasible}}$. In our experiments, we use a cost function of the form $\mathbf{C}(\hat{\mathcal{B}}, \hat{\mathcal{U}}) = \sum_{t=0}^{\ell} \text{tr}[M_t \hat{\Sigma}_t] + \sum_{t=0}^{\ell-1} \hat{\mathbf{u}}_t^T N_t \hat{\mathbf{u}}_t$ that encodes the objective of minimizing uncertainty (minimizing the trace of the covariance $\hat{\Sigma}_t$) while penalizing the control effort, but the cost function can also include problem-specific additional terms. The matrices $M_t \succeq 0$ and $N_t \succeq 0$ are positive semi-definite cost matrices that weigh the contributions of the two cost terms. The optimization is initialized with a belief trajectory $\bar{\mathcal{B}} = [\bar{\mathbf{b}}_0 \dots \bar{\mathbf{b}}_\ell]$ and $\bar{\mathcal{U}} = [\bar{\mathbf{u}}_0 \dots \bar{\mathbf{u}}_{\ell-1}]$.

The optimization formulation is general enough to include other constraints such as probabilistic collision avoidance with obstacles in belief space [15] but we do not consider obstacles in this work for the sake of clarity.

IV. Trajectory Optimization with Sensing Discontinuities

The general optimization problem given in Eq. (4) involves binary entries in the Kalman gain matrix in Eq. (3e). One alternative would be to include the binary entries in the Kalman gain matrix in Eq. (3e) in the optimization formulation. This leads to a binary nonlinear program, solving which is known to be at least NP-hard [5]. Directly considering the binary entries in a continuous nonlinear optimization formulation is challenging because when a measurement cannot be obtained, there is no local gradient information that would guide the optimization towards sensing regions where measurements could be obtained.

We use an approximation of the entries in $\boldsymbol{\delta}_t$ using sigmoid functions to map them to a value in the range $[0, 1]$, and this approximation is successively improved till the values of entries in $\boldsymbol{\delta}_t$ arrive within a user-defined threshold of the true binary values. We use sequential quadratic programming (SQP) [3], [23] to iteratively solve each nonlinear optimization problem with continuous approximations.

To consider a continuous approximation to the original problem, we first examine the relationship between the binary entries and whether valid measurements can be obtained. Let Π denote the valid sensing region associated with a given sensor. As shown in Fig. 2, the valid sensing region depends on factors such as the limited field of view of the sensor and presence of other objects within the field of view that might occlude portions of the sensing region. If the expected system state $\hat{\mathbf{x}}_t$ lies within the valid sensing region, a measurement
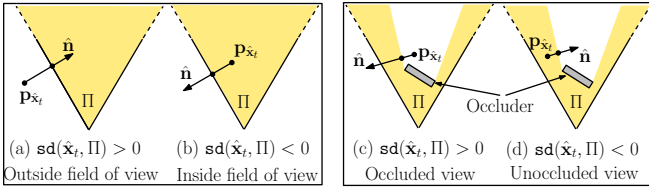
Fig. 2. **Relationship between measurements and signed distance to valid sensing region**: (a) The expected position $\mathbf{p}_{\hat{\mathbf{x}}_t}$ of the robot or object lies outside the field of view Π (shown in yellow) of a sensor, corresponding to a positive signed distance, where no measurements are obtained. The normal vector $\hat{\mathbf{n}}$ indicates the direction of closest approach. (b) A negative value of the signed distance corresponds to a valid measurement being obtained. (c,d) The field of view is partially blocked by an occluder object (shown in gray). Even though the expected position of the robot or object is within the triangular field of view, it is still outside (or inside) the valid sensing region corresponding to a positive (or negative) signed distance.

is obtained, and no measurement is obtained if it does not lie in the valid sensing region.

This relationship is formalized using the notion of signed distance $\mathtt{sd}(\hat{\mathbf{x}}_t, \Pi)$ in the workspace between the expected state $\hat{\mathbf{x}}_t$ of the system and the valid sensing region Π, as shown in Fig. 2. The signed distance here corresponds to the minimum translation distance required to bring the expected system state $\hat{\mathbf{x}}_t$ inside the valid sensing region Π from the outside (Fig. 2(a,c)) or push it outside the valid sensing region from the inside (Fig. 2(b,d)).

Each binary entry $\delta_t^i = \mathbb{1}(\mathtt{sd}(\hat{\mathbf{x}}_t, \Pi) < 0), i = \{1, \ldots, \dim[\mathbf{z}]\}$ then takes the value 1 when a measurement is obtained, i.e., when the signed distance to the valid sensing region is negative, and a value of 0 when the signed distance is positive, corresponding to when no measurement is obtained (Fig. 3). Inspired by [6], [28], we approximate the indicator function with a sigmoid function as:

$$\delta_t^i = \mathbb{1}(\mathtt{sd}(\hat{\mathbf{x}}_t, \Pi) < 0)$$
$$\approx \delta_t^i(\alpha) = 1 - \frac{1}{1 + \exp(-\alpha \cdot \mathtt{sd}(\hat{\mathbf{x}}_t, \Pi))}, \quad (5)$$

where α is a parameter that governs the degree of approximation of the indicator function $\mathbb{1}(\mathtt{sd}(\hat{\mathbf{x}}_t, \Pi) < 0)$, especially near the origin (Fig. 3). Increasing the value of α is equivalent to scaling the measurement noise levels outside the sensing region as a function of the signed distance. Note that the valid sensing region corresponding to δ_t^i is determined by the sensor that is responsible for measuring the ith dimension of the measurement vector $\mathbf{z}_t$. Using Eq. (5), we replace all occurrences of $\boldsymbol{\delta}_t$ in Eq. (3e) with $\boldsymbol{\delta}_t(\alpha)$.
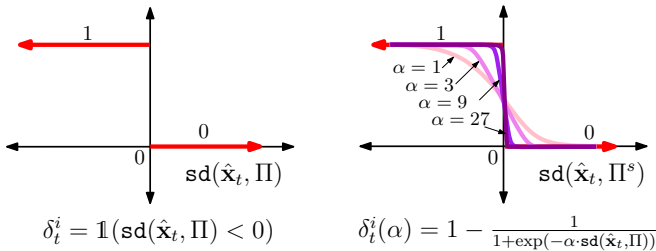

Fig. 3. Continuous approximation of $\boldsymbol{\delta}_t^i$ using sigmoid functions.

Alg. 1 outlines the steps involved in SQP-based optimization [3], [23] for belief space planning with sensing discontinuities. In particular, we sequentially solve a series

of continuous approximations to the true problem, obtained by increasing the value of the parameter α, starting from a user-defined constant α_{init}, by a constant factor k (lines [2-10] in Alg. 1). We used $k = 3$ for all our experiments. We update the trajectory initialization $[\bar{\mathcal{B}}, \bar{\mathcal{U}}]$ after each α update to seed the optimization. The algorithm converges when the entries of $\boldsymbol{\delta}_{0:\ell}$ are within a user-specified tolerance ϵ of the true binary values computed using Eq. (5).

V. EXECUTION

It is important to note that we only compute the signed distance to the valid sensing region with respect to the expected state $\hat{\mathbf{x}}_t$ for deciding if a measurement will be obtained or not. We compute this expected state based on the maximum likelihood observation assumption [18]. However, the expected state may not remain the same during execution after updating the belief state based on actual measurements. To account for the deviation in the expected state, we follow the model predictive control (MPC) paradigm [21] of re-planning after every time step. This has been demonstrated to be an effective way of performing feedback control to remain robust to large perturbations, provided one can re-plan sufficiently fast.

Alg. 1 outlines the steps involved in executing the trajectory by re-planning after every time step following a model predictive control paradigm. At each time step τ, we solve a belief space planning problem, as described in Sec. IV (lines [2-10]). We then select the first control input $\bar{\mathbf{u}}_\tau$ from the sequence of control inputs $\bar{\mathcal{U}}_{\tau:\ell-1}$ computed by the optimization (line 11) and apply the actuation commands to the system. We then obtain a measurement $\mathbf{z}_{\tau+1}$ and use it to update the belief state $\bar{\mathbf{b}}_{\tau+1} = \begin{bmatrix} \bar{\mathbf{x}}_{\tau+1} \\ \text{vec}[\sqrt{\Sigma_{\tau+1}}] \end{bmatrix}$ using the extended Kalman filter (EKF) as (line 12):

$$\bar{\mathbf{x}}_{\tau+1} = \mathbf{f}(\bar{\mathbf{x}}_\tau, \bar{\mathbf{u}}_\tau, \mathbf{0}) + K_t(\mathbf{z}_{\tau+1} - \mathbf{h}(\mathbf{f}(\bar{\mathbf{x}}_\tau, \bar{\mathbf{u}}_\tau, \mathbf{0}), \mathbf{0}), \quad (6a)$$

$$\sqrt{\Sigma_{\tau+1}} = \sqrt{\Sigma_{t+1}^- - K_t H_t \Sigma_{t+1}^-}, \quad (6b)$$

where Σ_{t+1}^-, K_t, and H_t are computed using Eq. (3). The Kalman gain matrix K_t from Eq. (3e) contains binary entries corresponding to if a measurement is obtained or not and it automatically disregards subsets of the obtained measurement $\mathbf{z}_{\tau+1}$ that do not constitute a valid measurement in the Kalman update (Eq. (6a)).

In the event that a measurement should have been obtained based on the estimated system state $\bar{\mathbf{x}}_{\tau+1}$ but no measurement is obtained during execution, the system can infer that the real system state is not present within the valid sensing region. We incorporate this information during execution by appropriately truncating the Gaussian belief state against the boundary of the valid sensing region.

Truncating the belief state (line 13): We use the method of Patil et al. [17] to locally convexify the region of the workspace not spanned by the valid sensing region, i.e., represent the complement of the valid sensing region as a conjunction of linear inequality constraints as $\bigcap_{i=0}^{k} \mathbf{a}_i^T \mathbf{x}_{\tau+1} \leq b_i$. For each constraint, we apply an affine transformation $x_{\tau+1}^i = \mathbf{a}_i^T \mathbf{x}_{\tau+1}$ to transform the

Algorithm 1 Belief space planning with sensing discontinuities (MPC paradigm)

Inputs:
    $\bar{\mathcal{B}}_{0:\ell} = [\bar{\mathbf{b}}_0 \ldots \bar{\mathbf{b}}_\ell]$, $\bar{\mathcal{U}}_{0:\ell-1} = [\bar{\mathbf{u}}_0 \ldots \bar{\mathbf{u}}_{\ell-1}]$: Belief space trajectory initialization
    ℓ: Number of time intervals
    Cost and constraint definitions (Eq. (4))

Parameters:
    α: Approximation parameter for relaxing discrete sensing constraints
    k: Coefficient to control rate of increase of α
    τ: Execution time step $(0 \leq \tau \leq \ell)$
    ϵ: Convergence tolerance parameter

Variables:
    $\hat{\mathcal{B}}_{0:\ell} = [\hat{\mathbf{b}}_0 \ldots \hat{\mathbf{b}}_\ell]$, $\hat{\mathcal{U}}_{0:\ell-1} = [\hat{\mathbf{u}}_0 \ldots \hat{\mathbf{u}}_{\ell-1}]$: Optimization variables
    $\boldsymbol{\delta}_{0:\ell}$: Binary vector to track value of continuous approximation for convergence criterion

1: **for** $\tau = 0, \ldots, \ell - 1$ **do**   ▷ Re-planning loop following the MPC paradigm
2:     $\alpha \leftarrow \alpha_{\text{init}}$
3:     **while** $\boldsymbol{\delta}_{\tau:\ell}$ not within ϵ tolerance of true binary values $\{0, 1\}$ **do**
4:         Reset trust region size and penalty coefficient   ▷ [23]
5:         $[\hat{\mathcal{B}}_{\tau:\ell}, \hat{\mathcal{U}}_{\tau:\ell-1}] \leftarrow$ SQP-based optimization of approximation given $[\bar{\mathcal{B}}_{\tau:\ell}, \bar{\mathcal{U}}_{\tau:\ell-1}]$ ▷ [23]
6:         $\alpha \leftarrow k * \alpha$ ▷ α-update to increase noise outside sensing region
7:         $\hat{\mathbf{b}}_{t+1} = \mathbf{g}(\hat{\mathbf{b}}_t, \hat{\mathbf{u}}_t) \ \forall \ t = \tau, \ldots, \ell - 1$ ▷ Integrate belief trajectory after α-update
8:         Update $\boldsymbol{\delta}_{\tau:\ell} \leftarrow \boldsymbol{\delta}_{\tau:\ell}(\alpha)$ ▷ Eq. (5)
9:         $[\bar{\mathcal{B}}_{\tau:\ell}, \bar{\mathcal{U}}_{\tau:\ell-1}] \leftarrow [\hat{\mathcal{B}}_{\tau:\ell}, \hat{\mathcal{U}}_{\tau:\ell-1}]$ ▷ Update trajectory initialization
10:    **end while**
11:    Execute $\bar{\mathbf{u}}_\tau$
12:    Obtain measurement and update $\bar{\mathbf{b}}_{\tau+1}$ using EKF   ▷ Eq. (6a)
13:    Truncate $\bar{\mathbf{b}}_{\tau+1}$ w.r.t sensing region boundary
14:    Update sensing regions for all sensors
15:    $\bar{\mathbf{b}}_{t+1} = \mathbf{g}(\bar{\mathbf{b}}_t, \bar{\mathbf{u}}_t) \ \forall \ t = \tau+1, \ldots, \ell-1$ ▷ Integrate belief trajectory after Kalman update
                                          ▷ using previously optimized control inputs $\bar{\mathcal{U}}_{\tau+1:\ell-1}$
16: **end for**

belief state $\mathcal{N}(\hat{\mathbf{x}}_{\tau+1}, \Sigma_{\tau+1})$ to a 1D Gaussian $x_{\tau+1}^i \sim \mathcal{N}(\mathbf{a}_i^T \bar{\mathbf{x}}_{\tau+1}, \mathbf{a}_i^T \Sigma_{\tau+1} \mathbf{a}_i)$ along an axis normal to the constraint. The problem now reduces to truncating the 1D Gaussian distribution at an upper bound given by $x_{\tau+1}^i = b_i$ to give $\tilde{x}_{\tau+1}^i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ [11]. We estimate the truncated belief state based on the truncated 1D Gaussian distribution $\tilde{x}_{\tau+1}^i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ [17]. We truncate the belief state $\bar{\mathbf{b}}_{\tau+1}$ with respect to each of the k constraints in a sequential manner and then accumulate the effect of truncation over all the constraints.

Updating the sensing regions (line 14): We update the valid sensing region for the sensors after execution at each time step. This step is necessary for sensors with a limited field of view, e.g., cameras or depth sensors, where the field of view might be blocked due to an occluder object that might move during execution. We perform a convex decomposition of the valid sensing region and check for occlusions. For instance, for a camera-like sensor, we project beams outward from the camera and truncate these beams with respect to occluder geometry. The convex decomposition facilitates signed distance computation and determination of constraints along the sensing region boundary for truncating the belief. We note that other decomposition methods such as voxel grids or geometric methods can also be used.

VI. EXPERIMENTS

A. Navigation: Mobile robots

Point robot: We consider a point robot navigating in a light-dark environment with a sensing discontinuity, adapted from Platt et al. [18] (Fig. 1). The robot is able to obtain reliable measurements about its state in the light region of the environment, but gets no measurements if it is in the dark region. We assume the following stochastic dynamics model $\mathbf{f}$ and measurement model $\mathbf{h}$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{u} \cdot \mathrm{dt} + S_{\mathbf{x}} \cdot \mathbf{q}_t \quad (7a)$$

$$\mathbf{z}_t = \mathbf{x}_t + S_{\mathbf{z}} \cdot \mathbf{r}_t, \quad (7b)$$

where the state $\mathbf{x} = [x, y]^T \in \mathbb{R}^2$ consists of the position of the robot, control input $\mathbf{u} = [v_x, v_y]^T \in \mathbb{R}^2$ consists of the robot's velocity, $\mathrm{dt} = 1$ is the duration of a time step, and $S_{\mathbf{x}} = 0.1 \cdot I$ and $S_{\mathbf{z}} = 0.01 \cdot I$ are constant matrices that scale the process noise $\mathbf{q}_t$ and measurement noise $\mathbf{r}_t$, respectively. The initial mean position is $\mathbf{x}_0 = [0, 4]^T$ and initial covariance is $\Sigma_0 = 0.5 \cdot I$, and the target position is at $[0, 0]^T$. The boundary of the light region is located at $x = 5$.

Fig. 1 shows that a state space planner [23] computes a straight line trajectory to the target, resulting in considerable final uncertainty. Belief space planning with a single continuous approximation ($\alpha = 1$) (Fig. 1(d)) plans a trajectory
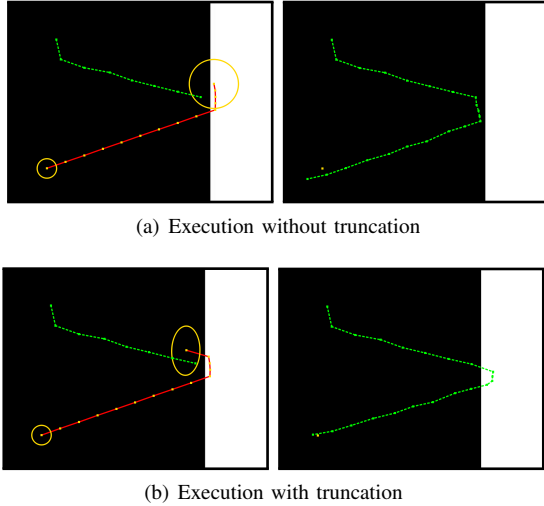
(a) Execution without truncation

(b) Execution with truncation

Fig. 4. **Effect of belief state truncation during execution**: The execution trace so far (dashed green line), current plan (solid red line), and current beliefs (yellow ellipses) are shown. The desired target is at the bottom left corner (yellow dot). (a) Without truncating the Gaussian belief state, the robot thinks it is in the light region where it can get measurements but it is actually still in the dark region. As a result, the robot starts heading back, resulting in a larger targeting error at the final time step. This results from the fact that the plan is computed starting from the expected state and not the true state, which is unknown during execution. (b) With truncation, at the same time step as (a), the belief is updated by truncating against boundary of light region and the robot estimates that it has not yet arrived in the light region. At the next step, the robot then performs a different action and moves into the light region for reducing uncertainty. Over 100 simulated executions, the robot reaches the light region in 64% of the executions without truncation and 100% of the executions with truncation. The state estimation during execution is more accurate with truncation, resulting in plans with reduced final targeting error.

towards the light region but the robot fails to reach it, while our approach results in the robot reaching the light region before heading to the target with reduced uncertainty (Fig. 1(c)). Fig. 4 shows the effects of truncation on the Gaussian belief state during execution. Our experiments indicate that considering truncation reduces the final uncertainty at the
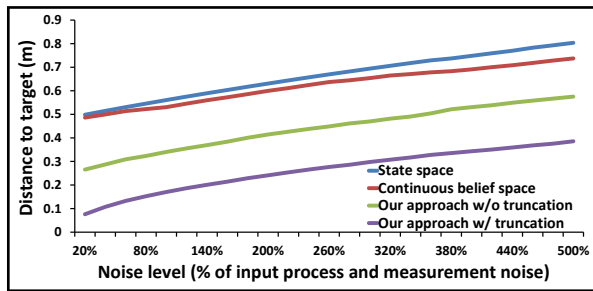

Fig. 5. Comparison of the targeting error using simulated executions with artificially added noise and re-planning after every time step for four approaches: (i) state space planning, (ii) belief space planning with a single continuous approximation ($\alpha = 1$), (iii) our approach without truncation, and (iv) our approach with truncation of the belief state. We varied the scaling of the process and measurement noise variances considered in Eq. (7) from 20% to 500% and considered the average distance to the target (in m) over 100 simulated executions for each noise level. Our approach with truncation of the Gaussian belief state results in the lowest error because it is able to consistently lead the robot to the light region to reduce the uncertainty. The rate of increase in the targeting error as the noise level increases is similar for all approaches since the initial state uncertainty is much larger than the considered process and measurement noise.
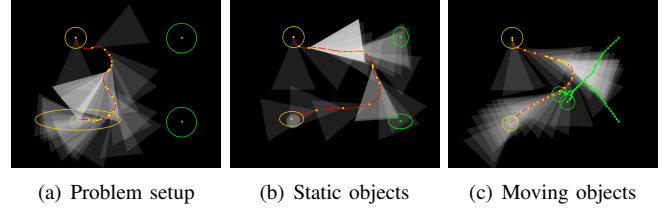

(a) Problem setup   (b) Static objects   (c) Moving objects

Fig. 6. **Composite belief space planning for robots and objects in the environment**: We consider a nonholonomic car-like robot with an on-board stereo camera sensor that is oriented in the direction in which the car is heading. The field of view of the camera is visualized in white. The robot localizes itself by detecting objects in the environment, whose initial positions are very uncertain (position and uncertainty shown in green). This is an example of belief space planning applied to active exploration. (a) A state space planner plans a trajectory that results in considerable uncertainty at the target. (b) When the objects are static, our approach finds a trajectory such that the car orients itself to place the objects in its field of view and then reaches the target, thereby reducing uncertainty in both the robot state and state of the objects. (c) The objects are now allowed to move. Our approach finds a trajectory for the robot and the objects such that the visibility of the objects is maximized in the field of view of the camera. The final uncertainty in car state and object positions reduces further. It is interesting to note that the final positions of the objects are close to each other since the uncertainty in the robot state and object states is correlated (just as in the simultaneous localization and mapping problem [26]).

target. Fig. 5 shows the results of evaluation of different approaches in terms of the final error at the target, using simulated executions with artificially added noise and re-planning after every time step. Our approach with truncated belief states outperforms other approaches that do not account for sensing discontinuities or do not truncate the belief state during execution.

Nonholonomic car-like robot: We consider a nonholonomic car-like robot navigating in an environment. The car is equipped with an on-board stereo camera sensor that is oriented in the direction θ in which the car is heading and the car is able to localize itself by detecting objects in the environment, whose initial positions are uncertain (Fig. 6(a)).

(i) **Static objects**: This is an example of belief space planning applied to the active exploration setting. We assume the following stochastic dynamics model $\mathbf{f}$ and measurement model $\mathbf{h}$:

$$\mathbf{x}_{t+1} = [x_t + v_t \, \mathrm{dt} \cos(\theta_t), \; y_t + v_t \, \mathrm{dt} \sin(\theta_t),$$
$$\theta_t + v_t \, \mathrm{dt} \tan(\phi_t)/d, \; x_t^1, \; y_t^1, \; x_t^2, \; y_t^2]^T + S_{\mathbf{x}} \cdot \mathbf{q}_t \quad (8a)$$
$$\mathbf{z}_t = [x_t^1 - x_t, \; y_t^1 - y_t, x_t^2 - x_t, \; y_t^2 - y_t]^T + S_{\mathbf{z}} \cdot \mathbf{r}_t, \quad (8b)$$

where the system state $\mathbf{x} = \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_{\mathcal{O}} \end{bmatrix} \in \mathbb{R}^7$ comprises of the state $\mathbf{x}_R = [x, y, \theta]^T \in \mathbb{R}^3$ of the robot containing its position $[x, y]^T$ and its orientation θ, and the state of the objects $\mathbf{x}_{\mathcal{O}} = [x^1, y^1, x^2, y^2]^T \in \mathbb{R}^4$ consists of the positions of the two objects. The control input $\mathbf{u} = [\phi, v]^T \in \mathbb{R}^2$ applied to the system consists of the robot's steering wheel angle ϕ and speed v. Here, dt is the time step duration, d is the length of the car-like robot, and $S_{\mathbf{x}}$ and $S_{\mathbf{z}}$ are constant matrices that scale the process noise $\mathbf{q}_t$ and measurement noise $\mathbf{r}_t$, respectively. Fig. 6(b) shows the planned trajectory for the car, in which the car orients itself in a such a way that it places the objects in the camera field of view, to reduce the overall uncertainty in the state of both robot and objects.

(ii) **Moving objects**: This is an example of composite belief space planning for both the robot and moving objects (or

sensors) in the environment. The object motion is assumed to be holonomic. We assume the following stochastic dynamics and measurement models:

$$\mathbf{x}_{t+1} = [x_t + v_t\,dt\cos(\theta_t),\ y_t + v_t\,dt\sin(\theta_t),\ \theta_t + v_t\,dt\tan(\phi_t)/d,$$

$$x_t^1 + v_x^1\,dt,\ y_t^1 + v_y^1\,dt,\ x_t^2 + v_x^2\,dt,\ y_t^2 + v_y^2\,dt]^T + S_{\mathbf{x}}\cdot\mathbf{q}_t \quad (9a)$$

$$\mathbf{z}_t = [x_t^1 - x_t,\ y_t^1 - y_t, x_t^2 - x_t,\ y_t^2 - y_t]^T + S_{\mathbf{z}}\cdot\mathbf{r}_t, \quad (9b)$$

where the composite control input is now given by $\mathbf{u} = [\phi, v, v_x^1, v_y^1, v_x^2, v_y^2]^T \in \mathbb{R}^6$, where $[v_x^k, v_y^k]$, $k = 1, 2$, is the speed of the kth object. Fig. 6(c) shows the planned trajectory for the car and the objects, such that the visibility of the objects in the camera field of view is always maximized. Since the uncertainty in the robot state and object states is correlated [26], the objects move close to each other to minimize the overall uncertainty in the system state.

B. Reaching Task: Planar three-link manipulator

We consider the case of a planar three-link manipulator with a stereo camera mounted at its base. The objective is to try to grasp a static object whose position is very uncertain, on account of the robot arm occluding the object in the camera field of view. We consider the following two cases:

(a) Problem setup (b) State space plan (c) Execution (intermediate) (d) Execution (final time step)
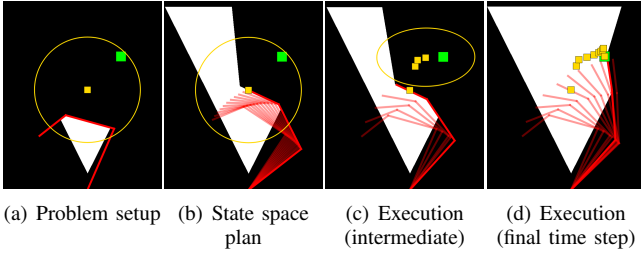
Fig. 7. **Reaching task under uncertainty (static camera):** We consider a three-link manipulator trying to grasp an object whose position is very uncertain, on account of the robot arm occluding the object in the camera field of view. The camera is looking towards top of figure and is fixed. The visible region in white is the part of the camera view frustum that is visible and the rest is occluded by the robot arm. The estimated position of the object is shown in yellow with very large uncertainty. The actual position of object is shown in green. (b) A state space planner computes a trajectory to estimated position of object but fails because the object is not present there. (c) An intermediate time step is shown during execution with re-planning. The robot first reaches the initial estimated position of the object, discovers nothing there but truncates the belief of the object position with respect to the sensing region. Traces of the updated estimated object position and truncated belief is shown in yellow. (d) The robot discovers the object during execution and is able to successfully grasp it.

(i) **Static camera**: The camera is assumed to be static and oriented as shown in Fig. 7(a). We assume the following stochastic dynamics and measurement models:

$$\mathbf{x}_{t+1} = [\theta_t^1 + \omega_t^1\,dt, \theta_t^2 + \omega_t^2\,dt, \theta_t^3 + \omega_t^3\,dt, x_t, y_t]^T + S_{\mathbf{x}}\cdot\mathbf{q}_t \quad (10a)$$

$$\mathbf{z}_t = [\theta_t^1, \theta_t^2, \theta_t^3, x_t - x^c,\ y_t - y^c]^T + S_{\mathbf{z}}\cdot\mathbf{r}_t, \quad (10b)$$

where the state $\mathbf{x} = [\theta^1, \theta^2, \theta^3, x, y]^T \in \mathbb{R}^5$ consists of the joint angles of the manipulator $[\theta^1, \theta^2, \theta^3]^T$ and the position $[x, y]^T$ of the object it is trying to grasp, the control input $\mathbf{u} = [\omega^1, \omega^2, \omega^3]^T \in \mathbb{R}^3$ consists of the angular speeds at each of the joints of the arm, and $[x^c, y^c]^T$ is the position of the camera base. Fig. 7 shows the traces of how the estimated object position and belief state are updated during execution using truncation to account for the fact that no measurements are obtained when a measurement should have been obtained

based on the estimated position of the object. The valid sensing region is also updated after each time step. The manipulator is able to successfully grasp the object during execution by accounting for the limited field of view of the camera and occlusions due to the arm.

(a) State space plan (b) Execution (intermediate) (c) Execution (final time step)
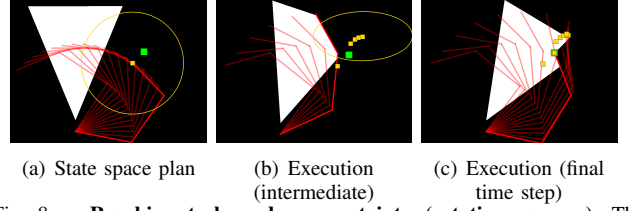
Fig. 8. **Reaching task under uncertainty (rotating camera):** The camera is now allowed to rotate about an axis (coming out of the page). The estimated position of the object is shown in yellow with very large uncertainty. The actual position of object is shown in green. (a) A state space planner computes a trajectory to estimated position of object but fails because the object is not present there. (b) An intermediate time step is shown during execution with re-planning. The robot first reaches the initial estimated position of the object, discovers nothing there but truncates the belief of the object position with respect to the sensing region corresponding to the new orientation of the camera. Traces of the updated estimated object position and truncated belief is shown in yellow. (c) The robot discovers the object during execution by rotating the camera at the base and is able to successfully grasp it.

(ii) **Moving camera**: The camera can now rotate about its axis (Fig. 7(a)). We assume the following stochastic dynamics and measurement models:

$$\mathbf{x}_{t+1} = [\theta_t^1 + \omega_t^1\,dt, \theta_t^2 + \omega_t^2\,dt, \theta_t^3 + \omega_t^3\,dt,$$

$$\theta_t^{\text{camera}} + \omega_t^{\text{camera}}\,dt, x_t, y_t]^T + S_{\mathbf{x}}\cdot\mathbf{q}_t \quad (11a)$$

$$\mathbf{z}_t = [\theta_t^1, \theta_t^2, \theta_t^3, \theta_t^{\text{camera}}, x_t - x^c,\ y_t - y^c]^T + S_{\mathbf{z}}\cdot\mathbf{r}_t, \quad (11b)$$

where the system state $\mathbf{x}$ now includes the camera orientation θ^{camera} and the control input includes the angular speed of rotation at the camera base ω^{camera}. Fig. 8(b) shows the traces of how the estimated object position and belief state are updated during execution using truncation against the valid sensing region corresponding to the camera orientation. The manipulator is able to successfully grasp the object during execution by simultaneously planning for both the manipulator and the camera.

C. Performance Evaluation

We analyzed the performance of our approach on each of the scenarios described above. The computation time depends on the number of optimization variables, which depends on the dimension of the belief state, control input, and the number of time steps used to discretize the trajectory. The computation time also depends on how many calls to the SQP solver (line 23 in Alg. 1) are made depending on the complexity of the problem. For instance, the planning problem for the nonholonomic car-like robot is difficult to solve because of the nonholonomic constraints on the car motion and the measurement model used. Note that the total execution time for execution is substantially faster than the number of time steps multiplied by the initial optimization time, since the optimization problem at successive time steps gets progressively easier to solve and a lot of effort is spent in computing the first locally-optimal solution.

Scenario	Belief state dim	Control input dim	Num time steps	Avg. time initial optimization (secs)	Avg time total exececution (secs)
point	5	2	20	1.6	9.8
car (static)	35	2	30	66.9	105.9
car (moving)	35	6	30	39.1	206
arm (static)	20	3	12	3.9	9.5
arm (moving)	27	4	12	15.3	30.9

Fig. 9. **Performance of our approach**: The average time for initial optimization indicates the time it takes to compute a locally-optimal solution for the first iteration ($\tau = 0$). The total execution time includes computation time required for re-planning after every time step. Performance evaluated using a C++ implementation running on a 3.2 GHz Intel PC.

VII. Conclusion and Future Work

In this work, we have addressed the problem of incorporating sensing discontinuities due to factors such as limited field of view of sensors and occlusions, in an optimization-based framework for belief space planning. Our key contributions include (i) solving a sequence of optimization problems that approximate sensing discontinuities using sigmoid functions in a SQP-based optimization framework, (ii) using the notion of signed distance between the estimated state of the system and the valid sensing region to ascertain if a measurement may be obtained, and (iii) using truncated Gaussians to update the belief state during execution to incorporate information contained in not obtaining a measurement. Our experiments suggest that this approach can be used to compute locally-optimal trajectories in belief space and effectively cope with discontinuities in sensing domains.

Our work opens up several avenues for future research. We plan to incorporate probabilistic collision avoidance constraints in our work [15]. Our experiments indicate that Gaussian belief space planning can provide compelling solutions to planning problems arising in robotic navigation and manipulation tasks. In situations where multi-modal beliefs are expected to appear, we plan to extend our approach to incorporate other general belief state representations [22]. We also want to apply this approach to real-world domains such as autonomous grasping and manipulation tasks with imprecise, articulated robots and medical needle steering.

References

[1] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, "Monte Carlo value iteration for continuous-state POMDPs," *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, pp. 175–191, 2011.

[2] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[3] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. Society for Industrial & Applied Mathematics, 2010, vol. 19.

[4] 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.

[5] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 97–106, 2012.

[6] C. Chen and O. L. Mangasarian, "Smoothing methods for convex inequalities and linear complementarity problems," *Mathematical Programming*, vol. 71, no. 1, pp. 51–69, 1995.

[7] S. Chen, Y. Li, and N. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research (IJRR)*, vol. 30, no. 11, pp. 1343–1377, 2011.

[8] T. Erez and W. D. Smart, "A scalable method for solving high-dimensional continuous POMDPs using local approximation," in *Uncertainty in Artificial Intelligence*, 2010, pp. 160–167.

[9] K. Hauser, "Randomized belief-space replanning in partially observable continuous spaces," *Proc. Workshop Algorithmic Foundations of Robotics (WAFR)*, pp. 193–209, 2011.

[10] G. Hollinger and G. Sukhatme, "Stochastic motion planning for robotic information gathering," in *Robotics: Science and Systems (RSS)*, 2013.

[11] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Multivariate Distributions: Models and Applications*. New York: John Wiley & Sons, 2002.

[12] L. P. Kaelbling and T. Lozano-Perez, "Integrated task and motion planning in belief space," *Int. Journal of Robotics Research*, 2013.

[13] T. Kollar and N. Roy, "Trajectory optimization using reinforcement learning for map exploration," *Int. Journal of Robotics Research*, vol. 27, no. 2, pp. 175–196, 2008.

[14] 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*, 2008.

[15] A. Lee, S. Patil, J. Schulman, Z. McCarthy, J. van den Berg, K. Goldberg, and P. Abbeel, "Gaussian belief space planning for imprecise articulated robots," in *IROS (to appear)*, 2013.

[16] R. Martinez-Cantin, N. de Freitas, E. Brochu, J. Castellanos, and A. Doucet, "A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot," *Autonomous Robots*, vol. 27, no. 2, pp. 93–103, 2009.

[17] S. Patil, J. van den Berg, and R. Alterovitz, "Estimating probability of collision for safe planning under Gaussian motion and sensing uncertainty," in *ICRA*, 2012, pp. 3238–3244.

[18] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems (RSS)*, 2010.

[19] 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.

[20] 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.

[21] J. Rawlings, "Tutorial overview of model predictive control," *IEEE Control Systems Magazine*, vol. 20, no. 3, pp. 38–52, 2000.

[22] N. Roy, G. Gordon, and S. Thrun, "Finding approximate POMDP solutions through belief compression," *Journal of Artificial Intelligence (JAIR)*, vol. 23, pp. 1–40, 2005.

[23] J. Schulman, J. Ho, A. Lee, H. Bradlow, I. Awwal, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *RSS*, 2013.

[24] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *Trans. on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[25] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Robotics: Science and Systems (RSS)*, 2005.

[26] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[27] 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.

[28] M. Vitus and C. Tomlin, "Sensor placement for improved robotic navigation," in *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[29] L. T. Watson and R. T. Haftka, "Modern homotopy methods in optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 74, no. 3, pp. 289–305, 1989.