

Interleaving Motion in Contact and in Free Space for Planning Under Uncertainty

Arne Sieverling

Clemens Eppner

Felix Wolff*

Oliver Brock

Abstract—In this paper we present a planner that interleaves free-space motion with motion in contact to reduce uncertainty. The planner finds such motions by growing a search tree in the combined space of collision-free and contact configurations. The planner reasons efficiently about the accumulated uncertainty by factoring the state in a belief over configuration and a fully observable contact state. We show the uncertainty-reducing capabilities of the planner on manipulation benchmark from the POMDP literature. The planner scales up to more complex problems like manipulation under uncertainty in seven-dimensional configuration space. We validate our planner in simulation and on a real robot.

I. INTRODUCTION

We propose a planner for robust motion under uncertainty that interleaves motion in free space with motion in contact. The need for free-space motion is obvious: motion in free space is efficient, easy to control, and generates low risk for damage to robot and environment. But motion in free space accumulates execution uncertainty, possibly leading to failure to achieve the desired goal. Since contact is a robust and effective way of reducing uncertainty [1, 2, 3], we propose to interleave both types of motions. As a result the planner must reason about the amount of accumulated uncertainty along the path to ensure robust goal attainment, while relying on free-space motion whenever possible.

The advantages of combining motion in contact and in free space are exploited extensively in robotics. For example, strategies that interleave motion in contact and in free space have been the key to success in the DARPA ARM challenge 2011, where robots touched the environment to localize the arm in the world (see Fig. 1). But so far there has been little research on how to plan such strategies from a description of the scene geometry.

We present a planner called *Contact-Exploiting RRT (CERRT)*, based on the rapidly exploring random tree (RRT) [4]. This planner finds robust manipulation strategies under uncertainty in robot position, actuation, and world model. The planner scales to high-dimensional configuration spaces. The resulting motions make and break contact with the environment, slide along surfaces, but also avoid collisions with links that have no contact sensing capability.

The main difficulty of planning under a partially observable robot state is the high dimensionality of the associated

All authors, except for Felix Wolff, are with the Robotics and Biology Laboratory, Technische Universität Berlin, Germany.

*Hasso-Plattner-Institut Potsdam, Germany.

We gratefully acknowledge the funding provided by the European Commission (EC, SOMA, H2020-ICT-645599). We thank Ludovic Righetti for access to Team USC’s material from the DARPA ARM challenge



Fig. 1. Contact can efficiently reduce the uncertainty about the robot’s state. *Left*: An example from a manipulation challenge, where the robot first touches the door to localize the handle before attempting a grasp [5]. *Right*: A similar strategy, generated by our planner CERRT. Shown in green are several executions of the motion under uncertainty, which all first contact the door and then slide down to the handle.

belief space. Our planner overcomes this problem by exploiting the insight that sensing contact is reliable and can be assumed to be fully observable. This factors the problem into a tractable reasoning over the robots position.

We evaluate the planner’s capability to reason efficiently about high uncertainty on a benchmark manipulation planning problem from the POMDP literature. We show how the planner generalizes to more complex problems by increasing the complexity and the dimensionality of the configuration space. We will validate our planning results with simulation and real world experiments for a manipulation task under significant uncertainty. We also provide quantitative results supporting our claim that contact and free-space motion must be interleaved.

II. RELATED WORK

Planning free-space motion and planning contact are two well-established research areas and we will briefly outline our planner’s connection to related work in both fields. Our planner balances free-space motion and contact by reasoning about uncertainty, for which we will review related work in the second half of this section.

A. Free-space motion

Sampling-based motion planners like the Probabilistic Roadmap (PRM) [6] or the RRT [4] search the collision-free configuration space efficiently and without any fixed discretization. These planners assume no uncertainty and explicitly avoid contact. In this paper we will modify RRTs to include contact and an explicit reasoning about uncertainty. In our planner, we exploit a handful of useful strategies from the motion planning literature: we utilize the Voronoi-bias [4] to quickly explore configuration and contact spaces, we use the goal-connect strategy [7] to balance exploration and exploitation [8], and we use a projection strategy similar

to task-constrained motion planning [9, 10] to implement sliding along surfaces.

B. Contact-space motion

Classic work in manipulation planning showed how a sequence of compliant motions can be robust to uncertainty. So called pre-images [1] characterize the regions from which compliant actions reach a desired goal state. Chaining them gives uncertainty-tolerant plans. In certain cases robust manipulation can be achieved without any sensors [2]. Sampling-based motion planning can explore the space of configurations in contact [11, 12] but does not reason about the uncertainty reducing capability. Our planner searches the space of all configurations in contact to exploit its uncertainty-reducing capability. Instead of backwards-chaining, it uses forward simulation to approximate pre-images.

C. Reasoning about uncertainty

To decide whether to exploit contact or to move in free space, our planner reasons explicitly about uncertainty. This distinguishes it from all previously mentioned methods. We will now review methods that reason about uncertainty explicitly.

Planning with uncertain actions: Markov Decision Processes (MDP) model actions with uncertain outcome. This framework allows robots to reason about the collision probability of actions and balance short and safe paths [13]. Very related to our method are particle-RRTs [14, 15] which represent the outcome of actions as a set of particles, just like our planner. Such a representation is suited to reason about the uncertainty-reducing capability of contact because the belief over configurations in contact is non-gaussian [15]. There are three important differences of particle-RRTs to our work: 1) The particle-RRT assumes perfect knowledge about robot state which CERRT does not. This allows our planner to solve a broader class of problems. 2) Our method explicitly seeks contact to reduce uncertainty, while the particle-RRT just achieves contact randomly. We believe this is the reason for our planner’s efficiency. 3) CERRT generates only one sequence of free-space and contact-motions while the particle-RRT has actions with multiple outcomes. This makes the particle-RRT’s behaviour more robust to failure. We will look into this behaviour to extend the CERRT in future work.

Planning with Uncertain Actions and Observations:

Once uncertainty exists in action outcome and the robot can not fully observe its own state, the planning problem is a Partially-Observable MDP (POMDP). The solution to a POMDP is a global sensing-action strategy that balances uncertainty reduction optimally with goal achievement. Unfortunately POMDPs of realistic sizes are intractable to solve optimally and hard to approximate due to the combinatorial explosion of belief space. To tackle the high complexity, further assumptions must be made. Assuming Gaussian state uncertainty is effective [16, 17, 18, 19, 20, 21] but does not adequately represent the belief state of configurations in contact. Sampling-based solvers [22, 23, 24] can approximate

POMDP solutions in reasonable time but are limited to low-dimensional problems, often with discrete states and actions.

Touch-based localization of the robot relative to a known environment can be casted as an optimization of a submodular metric [25, 26], which is efficiently solved by a greedy algorithm. However, submodularity does not hold if motions in free-space increase uncertainty.

Our method tackles the high complexity by planning with a belief over the robot configuration but a fully observable contact-state. This moves our problem in the domain of Mixed-observability MDPs [27] which are easier to solve.

POMDPs for manipulation: POMDP solvers were applied to low-dimensional versions of manipulation tasks such as in-hand manipulation to localize an object [28] or pre-grasp manipulation [29, 30]. The latter application is relevant to our method and we will show the uncertainty-reducing capabilities of our planner on the same problem in Section IV, but with two important differences: 1) unlike the POMDP-approaches [29, 30] our method does not assume any a priori discretization of state or action space. It can be directly applied just using the geometric model of world and robot as input. 2) we do not assume uncertain contact sensors while in the POMDP scenario sensors can return false measurements, which makes up a large part of the complexity. We think that our noise-free assumption is justified for undirected, binary contact-sensing.

III. CONTACT-EXPLOITING RRT (CERRT)

We now present our motion planner, which finds strategies that move both in free space as well as exploit contact to reduce uncertainty. To capture information about both uncertain configuration and contact, we will first define the search space of the planner.

CERRT plans with a combined state of belief over configuration and fully-observable contact $x = (\mathcal{Q}, \mathcal{C})$. We represent the belief over the configuration with a set of particles $\mathcal{Q} = \{q_1, \dots, q_N\}$, where each element q is an n -dimensional robot configuration (see Fig. 2). We will denote the sample mean and variance of \mathcal{Q} with μ_x and Σ_x . Each belief state is also associated with a fully observable contact $\mathcal{C} = \{c_1, \dots, c_m\}$. Each contact c is a pair of surfaces in contact $(s_{\text{robot}}, s_{\text{world}})$. s_{robot} is a surface on the robot that has contact sensing capabilities and s_{world} a surface of the environment.

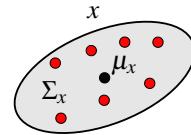


Fig. 2. The planner’s belief state x . The belief is represented by particles q_i (red). The black dot depicts sample mean μ_x and the gray ellipse depicts sample covariance Σ_x .

The planner finds strategies that combine free-space and contact motion. In CERRT we assume that free-space motion always increases uncertainty and model this with a noisy motion model $\delta(q)$ (which we will discuss in Sec. III-C). Because free space motions increase uncertainty, the

planner must sequence them with contact motions that reduce uncertainty. Fig. 3 shows an example of a decision the planner must take. The robot can not directly enter the narrow passage but must first contact the wall to reduce uncertainty.

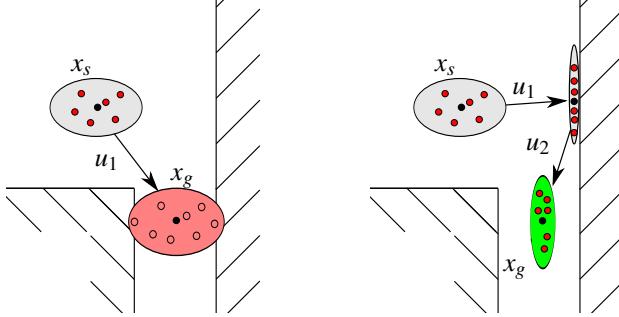


Fig. 3. *Left:* To enter the narrow passage, the robot cannot directly take action u_1 because the resulting uncertainty would lead to collision. *Right:* By sequencing a contact move u_1 and a free space move u_2 , the robot reduces position uncertainty sufficiently to enter the narrow passage. CERRT finds such sequences of contact and free-space motions.

To find such strategies, we grow a tree in the combined space of contact state and belief over configuration. The key to the planners efficiency is a tailored exploration strategy of this space. To adjust the search behaviour of the planner, we introduce a parameter $\gamma \in [0, 1]$ that describes the rate with which the planner attempts free-space or contact moves (see Fig. 4). If $\gamma = 0$, the planner only explores free space, and behaves like an RRT-Connect with goal bias. If $\gamma = 1$, the planner’s only objective is to reduce uncertainty. Values between 0 and 1 balance both objectives.

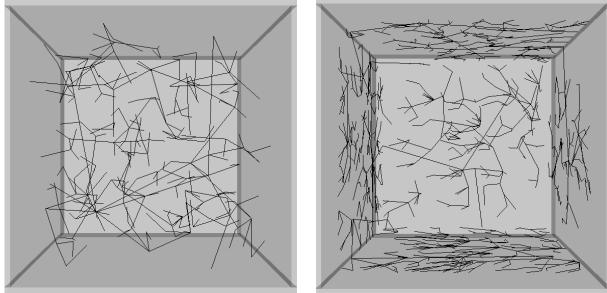


Fig. 4. The search behaviour of CERRT is governed by a free-space/contact exploration bias γ . Shown are two search trees of the CERRT planner exploring the inside of a cube for different values of γ . *Left:* For $\gamma = 0$, the behaviour matches that of a standard RRT. *Right:* For $\gamma = 1$, the planner searches the space of configuration in contact with the walls of the cube. The CERRT planner interleaves both behaviours.

The Contact-Exploiting RRT (CERRT) is closely related to the kinodynamic RRT [4] and its structure given in Algorithm 1 is identical to the RRT. However, CERRT differs substantially in the implementation of the subroutines which we will explain in detail in the rest of this section, following the order of the pseudocode in Algorithm 1.

A. Node selection: NEAREST_NEIGHBOUR

Like the RRT, our planner selects the next node to extend x_{near} with minimal distance to a randomly sampled configura-

Algorithm 1 CERRT

```

Input:  $x_{\text{start}}, x_{\text{goal}}, \epsilon_{\text{goal}}, \gamma$ 
Output:  $G = (V, E)$ 
1:  $V \leftarrow \{x_{\text{start}}\}$  init tree with start state
2:  $E \leftarrow \emptyset$ 
3: while true do search until goal reached
4:    $q_{\text{rand}} \leftarrow \text{RANDOM\_CONFIG}()$ 
5:    $x_{\text{near}} \leftarrow \text{NEAREST\_NEIGHBOUR}(q_{\text{rand}}, T, \gamma)$  Sec. III-A
6:    $u \leftarrow \text{SELECT\_INPUT}(q_{\text{rand}}, x_{\text{near}}, \gamma)$  Sec. III-B
7:    $x_{\text{new}} \leftarrow \text{NEW\_STATE}(x_{\text{near}}, u, q_{\text{rand}})$  Sec. III-C
8:   if IS_VALID( $x_{\text{new}}$ ) then Sec. III-D
9:      $V \leftarrow V \cup \{x_{\text{new}}\}$ 
10:     $E \leftarrow E \cup \{(x_{\text{near}}, x_{\text{new}})\}$ 
11:     $x_{\text{connect}} \leftarrow \text{NEW\_STATE}(x_{\text{new}}, \text{connect}, \mu_{x_{\text{goal}}})$ 
12:    if  $\|x_{\text{connect}} - x_{\text{goal}}\| < \epsilon_{\text{goal}}$  then
13:      return  $G$ 
14:    end if
15:   end if
16: end while

```

tion q_{rand} . Because the node is a belief state we need to define a suitable metric for states x . The choice of metric strongly influences the planning performance [31]. For CERRT, we use a metric that takes the parameter γ into account and can balance the search towards free space or contact motion.

For $\gamma = 0$ we want the tree to expand into free-space quickly, just like the RRT. We achieve this by choosing the node x_n whose mean is closest to q_{rand} . To do so we compute the Euclidean distance $d_\mu(x_n) := \|\mu_{x_n} - q_{\text{rand}}\|$.

For $\gamma = 1$ we want to reduce uncertainty by exploring contact space. We achieve this by picking a node with low uncertainty. More specifically, we compute a norm of Σ_{p_n} , which is the covariance matrix of the robot’s end-effector position p_n at configuration q_n . We then compute the trace norm, leading to: $d_\Sigma(x_n) := \sqrt{\text{tr}(\Sigma_{p_n})}$. We use this norm mainly because it does not become 0 if the distribution loses support in one dimension (which happens in contact), and also because it is inexpensive to compute.

For $0 < \gamma < 1$, we balance the two aforementioned metrics with a convex combination:

$$x_{\text{near}} = \underset{x_n}{\operatorname{argmin}} (\gamma d_\Sigma(x_n) + (1 - \gamma) d_\mu(x_n))$$

Both distance terms are normalized to the interval $[0, 1]$ by dividing them by the maximum observed value over all samples.

B. Action selection: SELECT_INPUT

After choosing a node for extension the planner needs to pick the next action. CERRT must have enough options to move in free space, along contact surfaces, or to switch from free space to contact and vice versa. We implement these options with three different action types. We will briefly introduce them here and give their implementation details later in Sec. III-C.

connect: this action attempts a straight line connection in configuration space to the sample q_{rand} . *connect* explores the free space and usually increases position uncertainty (Fig. 5(a)).

guarded: this action moves in the direction of q_{rand} until it establishes contact with the environment. *guarded* is required to switch from free space to contact and always reduces uncertainty in one dimension. (Fig. 5(b)).

slide: this action slides along a surface until the contact state changes, either by moving into another contact (Fig. 6(a)) or by leaving the sliding surface (Fig. 6(b)). *slide* explores the space of all contacts, always keeps uncertainty low in one dimension, and can reduce uncertainty in a second dimension.

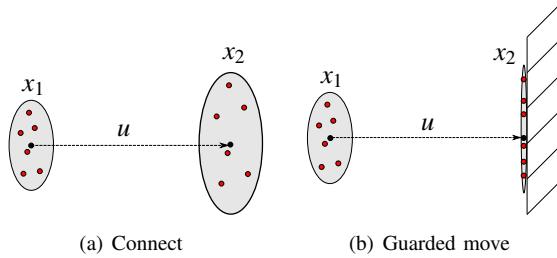


Fig. 5. A free-space move and a move into contact. x_1 and x_2 are the initial and final particle distributions before and after applying action u .

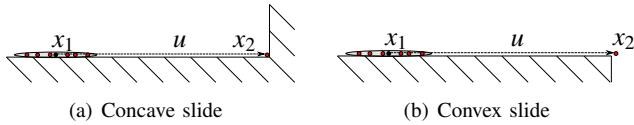


Fig. 6. Two sliding actions. (a) The slide moves the distribution along the surface until it achieves contact with another surface. (b) The slides moves until it loses contact with the surface. Both slides keep uncertainty low in one dimension and reduce it in another dimension.

Our planner selects one of the three actions probabilistically, biased by γ in the following way: if x_{near} is not in contact, it performs a connect move or a guarded move. If x_{near} is in contact, it slides or leaves the contact with a connect move. We choose actions based on these distributions:

$$\begin{aligned} p(\text{connect} | \mathcal{C}_x = \emptyset) &= 1 - \gamma \\ p(\text{guarded} | \mathcal{C}_x = \emptyset) &= \gamma \\ p(\text{connect} | \mathcal{C}_x \neq \emptyset) &= 1 - \gamma \\ p(\text{slide} | \mathcal{C}_x \neq \emptyset) &= \gamma \end{aligned}$$

We chose these distributions so that the planner is an RRT-Connect for $\gamma = 0$.

C. Forward simulation: NEW_STATE

For each of these actions, the planner must be able to reason about the change of uncertainty. We approximate this with a simulation of N noisy actions. The input to the simulation is a motion model $\delta_\alpha(\dot{q})$ with parameter vector α . Examples for the motion model δ are the classical angular and translational motion error for mobile robots or independent error for all joints of the robot.

To extend a node x_1 , CERRT samples a particle from \mathcal{Q}_{x_1} and also samples a vector α of parameters of the motion model δ . The extension step then is an invocation of the local planner that executes action u with the motion error δ_α , which we will describe in detail in the next Section.

The target of the local planner is q_{rand} with the initial error of the particle added. The extension step is repeated for all particles so that the outcome of the simulation is a new set of particles \mathcal{Q}_{x_2} which is added to the new state x_2 .

Algorithm 2 NEW_STATE

Input: x_1, u, q_{rand}

Output: x_2

```

1: for  $i \in N_{\text{particle}}$  do
2:    $q_{\text{near}} \leftarrow \text{SAMPLE}(\mathcal{Q}_{x_1})$  sample particle from node
3:    $\alpha \leftarrow \text{SAMPLE}(\mathcal{N}(0, \sigma_\delta))$  sample motion error
4:    $q_{\text{target}} \leftarrow q_{\text{rand}} + (q_{\text{near}} - \mu_{x_1})$  add the initial error
5:    $q_{\text{sample}} \leftarrow \text{LOCAL\_PLANNER}(a, q_{\text{near}}, q_{\text{target}}, \delta_\alpha)$  simulate
       action with one of the local planners (Sec. III-C)
6:    $\mathcal{Q}_{x_2} \leftarrow \mathcal{Q}_{x_2} \cup \{q_{\text{sample}}\}$ 
7: end for
8: return  $x_2$ 

```

Local planners: Each of the three action types invokes a different local planner. We implement them in the following way:

connect: A connect move is identical to the RRT version. A connect-particle moves on a straight line in configuration space towards the sample q_{rand} , checking for collisions with a fixed resolution of ϵ . If the particle reaches the sample or moves into contact the motion ends.

guarded: A guarded motion is a connect move in the direction of q_{rand} . A guarded move always ends in contact so it might end before q_{rand} or move beyond q_{rand} .

slide: Sliding motions start with particles in contact and move them along the surface, always maintaining contact. We implement sliding motions as task-space force-feedback controllers with constant orientation. To simulate sliding actions we first choose a random sliding surface (because the node might be in contact with two surfaces at the same time) and then project the end-effector position of the robot in configuration q_{target} onto the sliding surface. The algorithm then alternates between 1) taking a step towards the projected goal, 2) applying the motion error for this step 3) projecting the configuration back on the surface (see Algorithm 3). In this way, the effect of the joint-space motion error can be projected onto the lower-dimensional manifold of configurations in contact with the environment. The slide ends if the robot reaches the projected goal, if there is another contact, or if the robot loses contact with the sliding surface (see Fig. 6). For all projections we use a damped pseudo-inverse. If the robot is close to a singularity at any step ($\sqrt{\det(JJ^T)} < 0.001$ [32]) the slide method returns failure.

D. Node validation: IS_VALID

All nodes in CERRT must have a uniquely defined contact state. To ensure this, we only add those simulation outcomes to the tree that fulfill two requirements:

- 1) All $q \in \mathcal{Q}_{x_2}$ must either end up in free space or in contact with the same pair of surfaces.
- 2) If \mathcal{Q}_{x_2} contains configurations in contact, the contact must occur with a link that has a contact-sensor.

The first condition is crucial for our planner's performance because it ensures that the robots contact state is

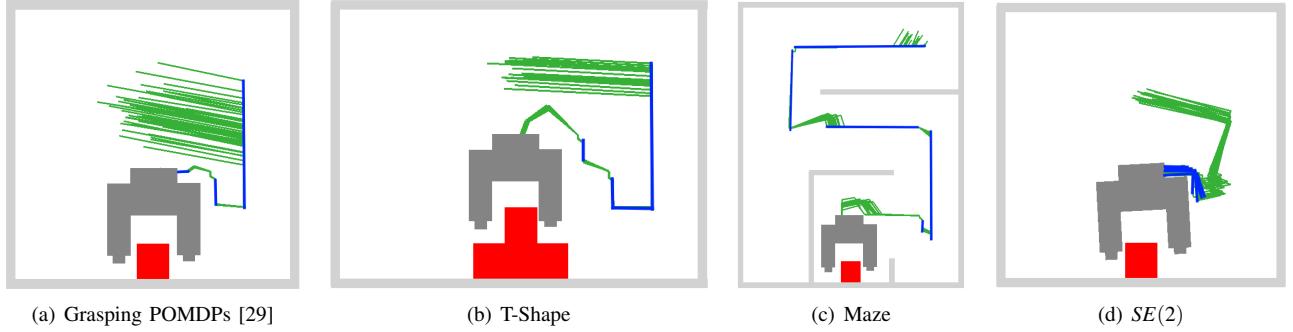


Fig. 7. Solutions of the CERRT planner for a grasping scenario. The gripper shows the final configuration of the path. The lines show 20 sampled trajectories, free-space motions are shown in green and slides in blue. The beginning of the paths is always in free space and the end is before grasping. CERRT outperforms POMDP planners on the benchmark (a) and scales to more complex problems.

Algorithm 3 SLIDE

Input: $q_{\text{near}}, q_{\text{sample}}$
Output: q_{real}

- 1: $(p_{\text{sample}}, R_{\text{sample}}) \leftarrow T_{EE}(q_{\text{sample}})$
- 2: $(p_{\text{surf}}, n_{\text{surf}}) \leftarrow \text{RANDCONTACT}(q_{\text{near}})$ sample random contact point and surface normal of q_{near}
- 3: $p'_{\text{sample}} \leftarrow p_{\text{sample}} - ((p_{\text{sample}} - p_{\text{surf}}) \cdot n_{\text{surf}}) n_{\text{surf}}$ project p_{sample} on surface
- 4: $\xi \leftarrow T_{\text{near}} - T_{\text{sample}}$
- 5: **while** $\|T_{EE}(q_{\text{robot}}) - T_{\text{sample}}\| > 0$ **do**
- 6: $\Delta q \leftarrow J^{\dagger}(q_{\text{robot}}) \xi$ move along surface towards sample
- 7: $q_{\text{robot}} \leftarrow q_{\text{robot}} + \varepsilon \cdot \Delta \hat{q}$ the particles most likely position
- 8: $q_{\text{real}} \leftarrow q_{\text{robot}} + \delta(\varepsilon \cdot \Delta \hat{q})$ the particles actual position
- 9: **while** $q_{\text{real}} \in \mathcal{C}_{\text{free}}$ **do**
- 10: $\Delta q_n \leftarrow -J^{\dagger}(q_{\text{new}}) n_{\text{surf}}$ move towards surface
- 11: $q_{\text{real}} \leftarrow q_{\text{real}} + \varepsilon \cdot \Delta \hat{q}_n$
- 12: **end while**
- 13: **end while**

always fully observable. It prevents all actions that end in separate, undistinguishable contacts. Other Particle-RRT planners [14, 15] do not restrict these actions because they introduce a clustering method and insert multiple nodes for different action outcomes. The second condition allows to treat measurable contact separate from undesired non-observable contact.

After inserting a valid node, the planner attempts to reach the goal state from the newly inserted node, also using forward simulation. If the resulting distribution is close to the desired goal distribution, the planner returns success. Otherwise it moves to the next iteration and picks another sample.

E. Policy generation

Given a sequence of actions and nodes from start to goal $(u_1, x_1), \dots, (u_n, x_n)$, we need to generate a policy that can be executed on a robot. This policy is a sequence that alternates between controllers and contact-based jump conditions. We instantiate one controller followed by one jump condition for each tuple of action and node (u_t, x_t) . The type of controller depends on u_t : from *connect* and *guarded* we generate a joint-space velocity controller and from *slide* we generate a compliant operational-space controller. The type of jump condition depends on the contact state \mathcal{C}_{x_t} : If there is contact,

the control switch is based on the magnitude of the measured force while for non-contact states, it is based on the covered distance $\|\mu_{x_t} - \mu_{x_{t-1}}\|$. We execute all controllers with low gains to safely make and break contact. This leads to weak tracking performance on the real robot but, as the policy is inherently robust, does not critically affect the outcome.

IV. EXPERIMENTS

In this section we will first show policies generated by CERRT for problems from the POMDP literature but also for a high-dimensional manipulation problem. Second, we will analyze the effect of the planner’s parameters quantitatively.

Our planner is implemented in the Robotics Library (RL)¹ using the Bullet physics library² for collision detection. We executed all experiments on an office PC with a 3.3 GHz Intel Core i5 CPU running the Linux operating system. In all experiments we use a constant number of particles $N = 20$ and a goal bias in the sampler of 10%. We always initialize the start belief state x_{start} by sampling N particles from the distribution $\mathcal{N}(q_{\text{start}}, \sigma_{\text{start}})$. All experiments use an independent linear motion error for all joints of $\delta_i(\dot{q}) = \mathcal{N}(0, \sigma_{\delta} \dot{q}_i)$.

A. Performance on manipulation problems

2D grasping: This problem models a gripper picking up a square block at unknown location and is inspired by the POMDP literature [29, 30]. The gripper has contact sensors at each jaw and can translate in two dimensions. Because of a large initial uncertainty the gripper must contact the object or the walls first and then, after uncertainty is sufficiently reduced, attempt the grasp from the top.

Fig. 7(a) shows one of the solution paths CERRT found on the simple grasping scenario. All policies first establish contact with wall or object and then slide along the ground until contact with the object is perceived. The planning time for this problem averaged over ten runs is 6.8 s (± 5.1 s). A POMDP version of the problem with discrete state and actions required an average planning time of 8 s [24] and 160 s with continuous state and discrete actions [30]. Our approach easily scales to more complex scenarios. Fig. 7(b)

¹roboticslibrary.org

²bulletphysics.org

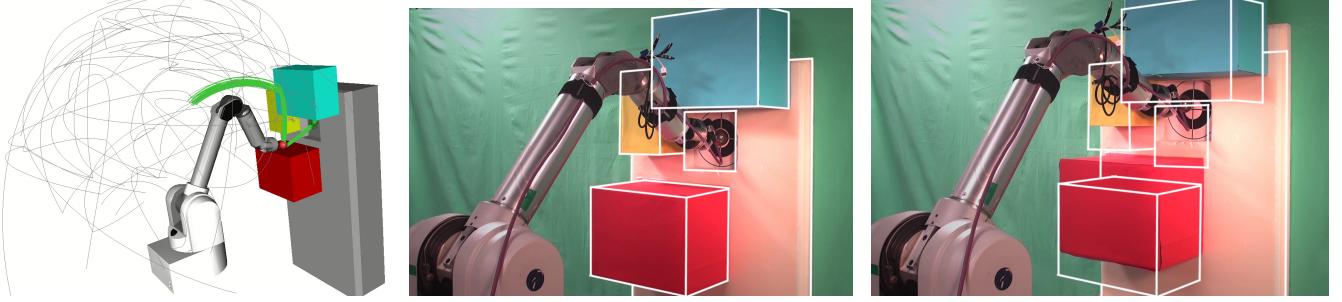


Fig. 8. The manipulator must touch the target in the square opening of the wall. *Left:* The planner output. Gray lines show all explored motions. The green line is the found path. Our planner finds a strategy that moves to the cyan box, slides down until it loses contact, does a guarded move to the top of the red box, and moves to the target. *Center:* The outcome of executing the strategy on the real robot without uncertainty. The robot reaches the goal precisely. *Right:* We now raise the obstacles by 7cm (the white overlay shows the wall position from (b)) and execute the policy from (b) again. The robot uses the contact to reduce uncertainty and reaches the target with an error of 2cm.

shows the result for a multi-step piece ($8.2\text{s} \pm 6.9\text{s}$), Fig. 7(c) a version where the gripper must first navigate through a simple maze ($23.4\text{s} \pm 19.3\text{s}$), Fig. 7(d) a 3D version of the problem with translation and rotation of the gripper.

7D robot arm motion: CERRT is efficient enough to be directly applied to the seven-dimensional configuration space. We place a 7-DOF Barrett WAM robot in front of the wall depicted in Fig. 8, similar to the scenario from Phillips-Grafflin et al. [15]. The robot model has an initial uncertainty and a motion uncertainty of $\sigma_{\text{start}} = \sigma_\delta = 0.02$. Motion-dependent position error occurs in the real Barrett WAM robot due to stretch of the cables that move the joints. The robot uses a wrist-mounted ATI Gamma force-torque sensor to perceive contact with the end-effector but cannot perceive contact with any other part. We now raise the obstacles by 7cm (the white overlay shows the wall position from (b)) and execute the policy from (b) again. The robot uses the contact to reduce uncertainty and reaches the target with an error of 2cm. The outcome of the planner can be seen in Fig. 8. From ten attempts, the planner solved this problem six times within 180s. The six successful searches required an average time of $23.8\text{s} \pm 29.3\text{s}$. To validate the robustness of the plan, we introduce an unexpected disturbance. We raise the wall including all obstacles by 7cm and execute the motion on the robot. The contact with the cyan and red boxes reduces uncertainty and the robot reaches the target with an error of 2cm which is an effective reduction of 5cm.

B. Quantitative analysis of planner parameters

We will now present the results of quantitative experiments that suggest sensible values for the two parameters of the planner: the free-space/contact-space exploration bias γ , and the number of particles N .

The influence of γ : We executed the planner on two different scenarios: 1) a 2D scenario with narrow passages 2) the 7D manipulation problem from Fig. 8. In our analysis we varied γ and the standard deviation of the motion uncertainty σ_δ . We set $\sigma_{\text{start}} = 0$. In both scenarios, we ran the planner ten times each for 66 different combinations of γ and σ_δ . We show the average planning time for each combination in Fig. 9. The results show a strong influence of γ on the planning time, depending on the uncertainty.

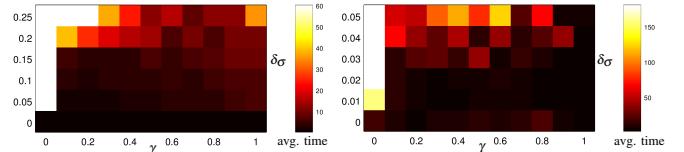


Fig. 9. Average planning time for different combinations of γ and σ_δ . *Left:* For the 2D scenario from Fig. 7(b) the optimal value of γ depends on the uncertainty. *Right:* for the 7D manipulation scenario from Fig. 8, the planner performs best for high values of γ , which lead to a contact-seeking behaviour.

The border case $\gamma = 0$ corresponds to pure free space search or pure contact motion. For the 2D scenario this is only reliable for problems without uncertainty. The case $\gamma = 1$ corresponds to a pure contact-space exploration. This strategy succeeds in both scenarios because they can be solved by a sequence of sliding motions. For values between 0 and 1 in the 2D scenario the planner always solves the problem. In 2D, free-space exploration is effective as long as uncertainties are low. A value of $\gamma = 0.3$ has the best performance. For high uncertainties more contact must be made and a value of $\gamma = 0.7$ performs best. In the 7D scenario from Fig. 8, the planner starts failing for uncertainties higher than 0.02 (we stop the search after 180s) and free-space exploration is far less effective. We achieved the best results with $\gamma = 0.95$. Our results show that for best planning performance, γ should be tuned to the problem at hand, as some problem require more free-space search and some require more contact.

The number of particles: The second important parameter is the number of particles to consider for planning. A too low number of particles will approximate the belief insufficiently which can lead to a policy with unexpected collisions. The number of particles influences the planning time at least linearly and should be kept low. To find a reasonable number, we ran the manipulator experiment (Fig. 8) 21 times varying the numbers of particles. We execute the resulting plans in a dynamic simulation implemented in the RoboticsLab³ framework and executed each plan 10 times with different motion error Fig. 10 shows the results of these experiments.

³simlab.co.kr

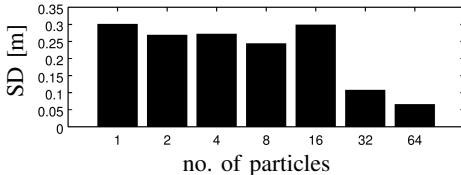


Fig. 10. The standard deviation of the final position error drops significantly with 32 particles.

While the average error of the robot's final position is about constant for different runs the standard deviation of the error drops at 32 particles. This suggests that the generated plans are not reliable below 16 particles. A similar number of particles was reported in [15].

V. CONCLUSION

We presented a planner to generate robust manipulation strategies under significant uncertainty in robot state, action, and world model. The planner achieves computational efficiency and robustness of resulting plans by interleaving motion in free space with motion in contact. The key to the planner's efficiency is a search strategy tailored to the combined contact space and free space as well as the assumption of a fully observable contact state. The experiments showed that the same planner can solve challenging benchmarks from the POMDP literature in continuous state and action spaces but also scales to realistic manipulation planning problems in configuration space. We believe there is room for runtime improvement in our planner and most extensions from the motion planning literature such as bidirectional search [7], guided sampling, and balancing of exploration and exploitation [8] will be just as useful for planning interleaved free-space and contact motion.

REFERENCES

- [1] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *The International Journal of Robotics Research*, vol. 3, no. 1, pp. 3–24, 1984.
- [2] M. A. Erdmann and M. T. Mason, "An exploration of sensorless manipulation," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 369–379, 1988.
- [3] C. Eppner, R. Deimel, J. Álvarez-Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1021–1038, 2015.
- [4] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Department of Computer Science, Iowa State University, Tech. Rep., 1998.
- [5] L. Righetti, M. Kalakrishnan, P. Pastor, J. Binney, J. Kelly, R. C. Voorhies, G. S. Sukhatme, and S. Schaal, "An autonomous manipulation system based on force control and optimization," *Autonomous Robots*, vol. 36, no. 1-2, pp. 11–30, 2014.
- [6] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [7] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2000, pp. 995–1001.
- [8] M. Rickert, A. Sieverling, and O. Brock, "Balancing exploration and exploitation in sampling-based motion planning," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1305–1317, 2014.
- [9] M. Stilman, "Task constrained motion planning in robot joint space," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2007, pp. 3074–3081.
- [10] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, 2011.
- [11] X. Ji and J. Xiao, "Planning motions compliant to complex contact states," *The International Journal of Robotics Research*, vol. 20, no. 6, pp. 446–465, 2001.
- [12] T. Siméon, J.-P. Laumond, J. Cortés, and A. Sahbani, "Manipulation planning with probabilistic roadmaps," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 729–746, 2004.
- [13] R. Alterovitz, T. Siméon, and K. Y. Goldberg, "The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty," in *Robotics: Science and systems*, vol. 3, 2007, pp. 233–241.
- [14] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 1617–1624.
- [15] C. Phillips-Grafflin and D. Berenson, "Planning and resilient execution of policies for manipulation in contact with actuation uncertainty," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.
- [16] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance," in *Robotics Research*. Springer, 2010, pp. 293–305.
- [17] R. Platt Jr, R. Tedrake, L. Kaelbling, and T. Lozano-Pérez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems*, 2010.
- [18] R. Platt, L. Kaelbling, T. Lozano-Pérez, and R. Tedrake, "Simultaneous localization and grasping as a belief space control problem," in *International Symposium on Robotics Research*, vol. 2, 2011.
- [19] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [20] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 723–730.
- [21] A.-A. Agha-Mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 268–304, 2014.
- [22] S. Thrun, "Monte carlo POMDPs," in *NIPS*, vol. 12, 1999, pp. 1064–1070.
- [23] J. M. Porta, N. Vlassis, M. T. Spaan, and P. Poupart, "Point-based value iteration for continuous POMDPs," *Journal of Machine Learning Research*, vol. 7, pp. 2329–2367, 2006.
- [24] 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.
- [25] S. Javdani, M. Klingensmith, J. A. Bagnell, N. S. Pollard, and S. S. Srinivasa, "Efficient touch based localization through submodularity," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 1828–1835.
- [26] N. A. Vien and M. Toussaint, "Touch based POMDP manipulation via sequential submodular optimization," in *International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 407–413.
- [27] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Planning under uncertainty for robotic tasks with mixed observability," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1053–1068, 2010.
- [28] M. C. Koval, N. S. Pollard, and S. S. Srinivasa, "Pre-and post-contact policy decomposition for planar contact manipulation under uncertainty," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 244–264, 2016.
- [29] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Grasping POMDPs," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 4685–4692.
- [30] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, "Monte carlo value iteration for continuous-state POMDPs," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*. Springer, 2010, pp. 175–191.
- [31] Z. Littlefield, D. Klimenko, H. Kurniawati, and K. Bekris, "The importance of a suitable distance function in belief-space planning," in *International Symposium on Robotics Research*, 2015.
- [32] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.