Contingent Contact-Based Motion Planning

Előd Páll Arne Sieverling Oliver Brock

***Abstract*— A robot with contact sensing capability can reduce uncertainty relative to the environment by deliberately moving into contact and matching the resulting contact measurement to different possible states in the world. We present a manipulation planner that finds and sequences these actions by reasoning explicitly about the uncertainty over the robot's state. The planner incrementally constructs a policy that covers all possible contact states during a manipulation and finds contingencies for each of them. In contrast to conformant planners (without contingencies), the planned contingent policies are more robust. We demonstrate this in simulated and real-world manipulation experiments. In contrast to POMDP-based planners, we show that our planner can be directly applied to high-dimensional configuration spaces.**

I. Introduction

When robots move in the real world, their motion is unavoidably affected by uncertainty. This uncertainty may lead to a single action having possibly several different outcomes. To be prepared for this, a motion planner must in advance determine suitable reactions and capture these alternatives in branches of a motion plan. During the execution of such a plan, the planner can use sensor data to select among these branches the one matching the current situation. A plan able to address such eventualities is called a *contingency* plan; a planner producing it is called a *contingent* planner [1].

To find contingency plans, a planner can model the uncertainty of the robot and reason probabilistically about sensor events that might happen during execution. However, taking into account all possible eventualities is not possible. The high-dimensionality and the continuous state and action space in manipulation problems make global, complete solutions to contingent planning intractable. To be effective, a planner must make approximations.

We present the Contingent, Contact-Exploiting RRT (ConCERRT)—a planner that overcomes this complexity for contingencies based on contact sensing. The planner finds robust strategies in configuration space for problems such as the one shown in Fig. 1. These problems require the robot to adapt its motion based on sensor information obtained from contact sensors. The key to our planner is the assumption that contact sensing is uncertainty-free. This assumption allows to rule out a large part of the robot's state space, given a contact measurement. This simplifies planning and leads to a low number of contingencies.

Compared to related Partially Observable Markov Decision Process (POMDP) approaches, our planner does not require any a priori discretization of configuration or action space. Instead, it builds a task-specific discretization of the state space during planning, informed by the available actions, similar to sampling-based motion planners. In our experiments, we show that our planner consistently finds successful contingency plans in realistic applications. We show solutions for problems with high uncertainty where non-contingent planners fail. In real world experiments, we show that our assumptions about uncertainty hold in realistic scenarios.

All authors are with the Robotics and Biology Laboratory, Technische Universität Berlin, Germany. We gratefully acknowledge the funding provided by the European Commission (SOMA, H2020-ICT-645599).

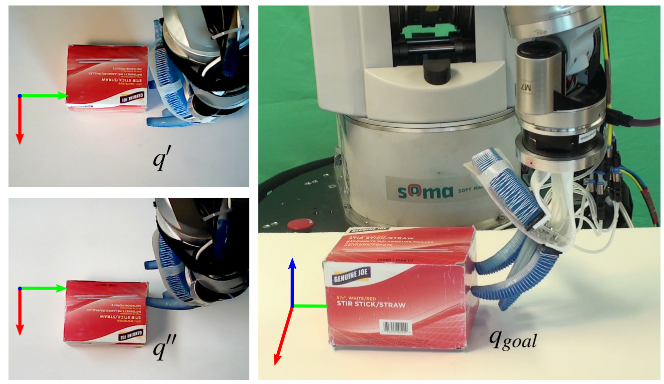


Fig. 1. A robot performs tactile localization of an object using contact sensors on two fingers of a soft hand. By moving into contact and measuring which finger makes contact first, the robot can estimate the position of the box relative to itself and then adapt its action to reach q_{goal}. Our planner finds such strategies using models of the world and the uncertainty.

II. Related Work

We will discuss methods on the spectrum between uncertainty-unaware planning and complete search in belief space, in increasing order of their treatment of uncertainty.

A. Sampling-based Motion Planning

Sampling-based motion planners, like RRT [2], generate a collision-free path from a description of the environment and robot kinematics. They efficiently search the high-dimensional configuration space because they do not require any a priori discretization but create it during planning, adapted to the problem. Sampling-based motion planners can plan in the space of configurations in contact [3, 4, 5], but they usually do not consider uncertainty which makes resulting plans brittle. Our planner is based on the RRT framework, but it reasons explicitly about uncertainty.

B. Conformant Planning

A conformant plan is a fixed sequence of actions that is guaranteed to lead to a goal state, even under significant uncertainty. To find such plans, planners can operate in information space [2] or, as a special probabilistic case, belief space. A classic way to compute conformant plans is to compute all regions from which compliant actions lead to a goal (pre-images) and then chain them to a sequence of actions [6]. This framework was used to find manipulation strategies that bring objects with unknown position into a desired state, without any sensing [7, 8].

For high-dimensional problems, conformant plans can be found in a sampling-based framework by representing the uncertain state as a Gaussian distribution [9, 10, 11, 12] or as a set of particles [13, 14, 15, 16]. Highly related to our planner is the Contact-Exploiting RRT (CERRT) planner [17] which is a conformant, sampling-based planner that searches a combined space of configurations in contact and in free space. Our planner also searches this combined space using a particle-based state but instead of a single action sequence it generates a policy with multiple branches in reaction to contact events.

C. Contingent Planning

A contingency plan (also conditional plan or feedback plan) is a decision tree or graph that branches based on observations. It can be understood as a policy π that maps observations to actions. Contingent planners generally solve a broader class of problems than conformant planners, e.g. part orientation for arbitrary shapes and realistic friction models [18]. In this paper we present a contingent extension of the CERRT planner. One way to add contingencies to a plan is to reverse and retry motions that do not lead to the desired outcome [14]. However, in our problem most motions increase uncertainty and therefore are generally irreversible. Instead, our planner incrementally constructs the policy π by repeatedly invoking the conformant CERRT planner and reconnecting less likely outcomes to previously found solutions.

D. Optimal Planning under Uncertainty

A generic approach to compute globally optimal contingency plans is the POMDP framework. The solution to a POMDP is a policy π that maps from belief space (i.e. the space of probability distributions) to actions. For low-dimensional, discrete action and state spaces, point-based solvers [19] can approximate optimal solutions. POMDPs were used to solve simple 2D grasping problems [20] that use contact as feedback. However these approaches scale badly in high-dimensional spaces. While there exist solvers for continuous state [21] and action spaces [22], they can not easily be applied to the high-dimensional configuration space of a robot manipulator.

One way to use POMDP solvers for manipulation is to discretize the lower-dimensional manifold of configurations in contact [23, 24]. This allows to solve a tactile localization task, where a robot uses a sequence of contact motions to locate an object. However, the discretization limits the approach to problems with few possible contacts. Our approach does not rely on a predefined discretization but only searches the reachable contact manifold during planning. In our experimental results we show that we can generate similar policies on the tactile localization task for more complex contact manifolds.

III. Problem Description

Planning contingencies for contact-sensing robots requires to combine reasoning about uncertainty with a model for contact sensing. Before presenting our algorithm, we will describe the problem formally.

ConCERRT plans in the n-dimensional configuration space $\mathcal{C}$. $\mathcal{C}_{\text{valid}}$ is the *valid* configuration space composed of free space $\mathcal{C}_{\text{free}}$ and the configurations in contact at the boundary $\partial\mathcal{C}_{\text{free}}$. The robot can execute actions u which are either straight line joint space motions in free space, guarded motions (moving until the robot is in contact), or compliant slides along surfaces. All motions have an uncertain outcome and the robot can not fully perceive its configuration but must estimate it from noisy sensors. Additionally, the robot does not fully know its initial configuration. Therefore, instead of planning in configuration space, we plan in belief space $\mathcal{B}$, where each belief $b \in \mathcal{B}$ is a probability distribution over the configuration space. We model the initial state uncertainty as a Gaussian distribution $b_0 := \mathcal{N}(q_0, \sigma_0)$ and use a motion model with independent joint noise $\delta\hat{q} = \delta q + \mathcal{N}(0, \sqrt{|\delta q|}\sigma_\delta)$. We assume the robot has access to reliable contact sensors to observe the active contact(s) at a given configuration q. A key assumptions for efficient planning is that contact sensing is fully observable. i.e. we assume that no contact is ever detected wrongly. In this paper we use two different contact sensor models:

1) *Tactile sensor model* assumes binary contact sensing on different parts of the robot: $\mathcal{O}_{\text{tactile}}(q) = \{o_1, \ldots, o_k\}$, where each contact observation $o_i = c_i$ is a sensor patch indicating contact in the given configuration.

2) *Force sensor model* assumes the robot can detect the contact normal: $\mathcal{O}_{\text{force}}(q) = \{o_1, \ldots, o_k\}$. Each observation $o_i = (c_i, n_i)$ is a pair of sensor patch and wall normal n_i. A belief b is valid, if it lies mostly in the valid configuration space, i.e.:

$$\int_{q \in \mathcal{C}_{\text{valid}}} b(q)dq > 1 - \varepsilon$$

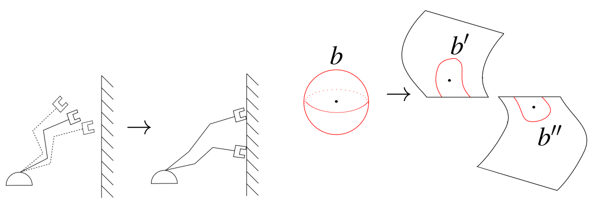
$\mathcal{B}_{\text{valid}}$ is the space of all valid beliefs. The planning problem is now the following: given a start and goal belief $b_0, b_g \in \mathcal{B}_{\text{valid}}$, find a policy $\pi : \mathcal{B}_{\text{valid}} \to \mathcal{U}$ that brings the robot to the goal belief state with high probability. In this paper we only care about finding feasible policies and do not consider optimality. This problem is a belief space planning problem [10], however, the possibility for the robot to make contact with the environment makes the state non-Gaussian or even multi-modal.

IV. ConCERRT

In this section, we present the Contingent Contact-Exploiting RRT planner (ConCERRT), a sampling-based motion planner that finds contingency plans based on contact sensing. Before describing the algorithm, we will first explain two insights that are crucial for understanding our planner.

A. *Belief State Partitioning*

The first insight is that a robot can effectively reduce uncertainty by moving into contact and observing the resulting contact measurement to rule out parts of its state space. For example, a robot (Fig. 2) with uncertain state moves towards a wall until it touches with the left or the right finger. By visualizing this in belief space, the reduction of uncertainty becomes obvious: the contact event partitions the belief in two halves, each with less uncertainty than the original belief.



(a) A 3-DOF robot with initial position uncertainty moves from free space into contact with a two-fingered hand.

(b) Contact partitions the belief b into b' and b'' configurations in contact with the left and right fingers respectively.

Fig. 2. Belief state partitioning with binary contact sensing

These belief space partitions can also arise out of contact direction sensing which we show in Fig. 3. Here the robot (in this case a 2D point robot) can sense the normal at the point of contact. The robot now moves towards an edge and matches the sensor reading to the wall normals. Just like before, the result of this action is the partition of a large belief into two smaller belief states, which is an efficient reduction of uncertainty.

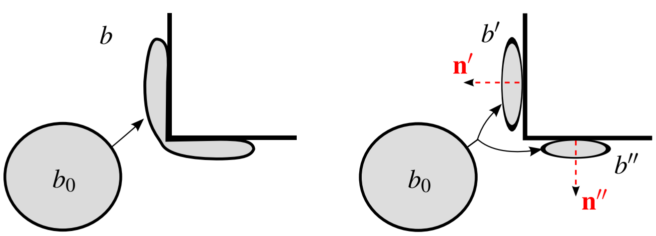


Fig. 3. *Left*: a robot with state uncertainty moves from state b_0 towards a corner in the world, which projects the uncertainty on the surfaces. *Right*: measuring the contact normal ($\mathbf{n}'$ or $\mathbf{n}''$) allows the robot to partition its belief state b into lower uncertainty states b' and b''.

This is the first insight exploited by the ConCERRT planner: some actions reduce uncertainty by partitioning the belief space. The ConCERRT planner exploits these actions and assembles them into robust policies.

B. *Incremental Policy Construction*

Using belief-space partitioning actions in a planner is not straightforward as every added partition adds at least two new belief states to the policy. Both states must be eventually connected to the goal. However, we will show now that this effort can be limited in practice, as a planner can reuse sub-solutions to speed up planning.

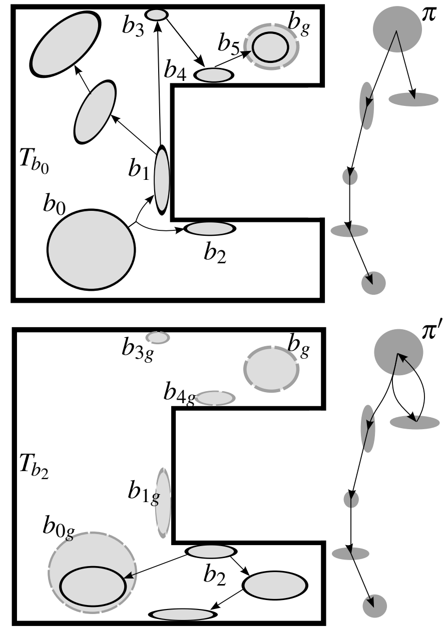


Fig. 4. Two iterations of the ConCERRT planner: *Top left*: the initial search tree T_{b_0} connecting start and goal beliefs b_0 and b_g. *Top right*: the resulting policy π, which consists of one path from start to goal but also contains one unconnected partition b_2. *Bottom left*: the next iteration of the algorithm. Starting from b_2, the algorithm builds a new search tree T_{b_2} that can connect to any of the beliefs in π. The planner finds a path that can reconnect by moving back to b_{0g}. *Bottom right*: This path is added to the final policy π'.

The working principle of ConCERRT is shown in Fig. 4. ConCERRT maintains two separate lists of belief states:

- $\mathcal{B}_{\text{open}}$ contains all belief states that are yet to be connected to the goal. It is initialized with the initial belief of the robot and increases with every belief space partition. If $\mathcal{B}_{\text{open}}$ is empty, the planner returns success.
- $\mathcal{B}_{\text{connected}}$ contains all beliefs that are already connected to the goal. Initially, it only contains the goal belief b_g, however over time the planner expands the policy and adds elements to $\mathcal{B}_{\text{connected}}$.

ConCERRT now runs for every state in $\mathcal{B}_{\text{open}}$ a separate tree search, attempting to connect to any state in $\mathcal{B}_{\text{connected}}$. If it can connect any node from $\mathcal{B}_{\text{open}}$ to any node from $\mathcal{B}_{\text{connected}}$, it adds the resulting action sequence to the policy and it also adds all nodes visited by that path to $\mathcal{B}_{\text{connected}}$. If this sequence results in any new partitions, it adds to $\mathcal{B}_{\text{open}}$ and creates a new tree for each of them.

This parallel search using a whole forest of trees might seem like an overhead. However, the effort is limited in practice, which can be explained by the algorithm moving from exploration to exploitation [25]. Initially the algorithm must explore most of the space as $\mathcal{B}_{\text{connected}}$ contains only one element. But whenever adding a path to the policy, the algorithm also adds states to $\mathcal{B}_{\text{connected}}$. At some point nodes in $\mathcal{B}_{\text{connected}}$ will cover most of the state space. All these beliefs are opportunities for exploitation which decrease the complexity of later iterations.

C. Algorithm Outline

Based on the two previous insight we can now give the full description of the ConCERRT planner (Alg. 1). To plan efficiently with the non-Gaussian belief states, we represent the belief with a set of particles $b = \mathcal{Q} = \{q_1, \ldots, q_N\}$ where each particle q is a configuration. We denote the sample mean and covariance of the configurations in a belief b with μ_b and Σ_b respectively.

ConCERRT initially samples a fixed number of particles from the initial belief b_0 and adds them as root to the initial search tree. Then, in every iteration, ConCERRT cycles through all elements of $\mathcal{B}_{\text{open}}$ and expands the respective tree. The expansion works similar to an RRT planner. It samples a random configuration, finds a nearest neighbour in the current search tree, chooses an action, simulates the effects of that action and adds the resulting state to the tree and tries to connect the new state to the goal(s).

Algorithm 1 ConCERRT

Input: b_0, b_g
Output: π
$\mathcal{B}_{\text{open}} \leftarrow b_0$
$\mathcal{B}_{\text{connected}} \leftarrow b_g$
$T_{b_0} \leftarrow b_0$
$\pi \leftarrow \emptyset$
while $P(\pi) < 1$ **do**
  for all $b \in \mathcal{B}_{\text{open}}$ **do**
    $T_b \leftarrow T_b.\text{EXPAND}(\mathcal{B}_{\text{connected}})$
    $\pi \leftarrow \pi.\text{UPDATE}(T_b)$
    $\mathcal{B}_{\text{open}} \leftarrow \mathcal{B}_{\text{open}}.\text{UPDATE}(T_b)$
    $\mathcal{B}_{\text{connected}} \leftarrow \mathcal{B}_{\text{connected}}.\text{UPDATE}(T_b)$
return π

We will now give implementation details for the expansion. The numbers in parentheses refer to the lines in Alg. 2.

Sampling (1): The planner samples a random configuration to extend the current tree towards. The tree growth is randomly biased towards the goal by choosing μ_{b_g} instead of a random sample with a fixed probability $p(b_g)$.

Nearest neighbour search (2): The nearest neighbour selection computes a norm that balances uncertainty and Euclidean distance. As a distance term over uncertainty we use the trace norm $d_\Sigma(b) := \sqrt{\text{tr}(\Sigma_b)}$. As spatial distance to the random sample q_{rand} we use $d_\mu(b) := \|\mu_b - q_{\text{rand}}\|_2$. We select the best neighbour based on a bias-parameter $\gamma \in [0,\ 1]$:

$$b_{near} = \text{argmin}_b \left[\gamma \left(d_\Sigma(b) + \sum_{b' \in Sib(b)} d_\Sigma(b') \right) + (1-\gamma) d_\mu \right]$$

$Sib(b)$ denotes the set of all siblings of b , i.e. the partitions that were reached from the same action. For details about the influence of the γ-parameter, we refer to the CERRT planner [17].

Action selection (3): The planner chooses an action u randomly. The possible actions are: *connect*, which moves particles towards the random sample on a straight line; *guarded*, which moves toward the random sample and stops when contact is gained; *slide*, which moves in contact along a surface, maintaining contact until the contact state changes. These actions are identical to the CERRT planner [17].

Simulation (4): To compute the resulting belief state b' from applying action u in belief b_{near}, we simulate the execution of u for every particle in b_{near} by sampling the noisy motion model. We check b' for joint limits or collisions with links that do not sense contact *(5)*.

Belief state partitioning (6): If b' is valid, we apply the contact sensor model to find potential partitions of the belief state. For each particle $q' \in b'$ we compute $\mathcal{O}(q') = \{o_0, \ldots, o_k\}$. We then cluster the belief b' into $\{b'_{o_0}, \ldots, b'_{o_k}\}$, such that particles with the same measurement are in the same belief. The implementation is different for the two sensor models: For the *tactile* sensor model, we cluster based on the sensor patches that are in contact. For the *force* sensor model, we cluster two particles into different beliefs if the difference between their measured normals is larger than 15°. We estimate the transition probabilities as $p(b'_{o_i}|b,u) \approx \frac{|\mathcal{Q}(b'_{o_i})|}{N_{\text{particles}}}$

Goal connect (9): We add the new beliefs b'_{o_i} to the tree and try to connect them to any belief in $\mathcal{B}_{\text{connected}}$. To do so we simulate a noisy connect action towards every $b_{\text{goal}} \in \mathcal{B}_{\text{connected}}$ resulting in a new distribution b''. We check if b'' lies within the goal belief by testing if $d_M(q) < \varepsilon_M = 2$ for all $q \in b''$, where $d_M(q)$ is the Mahalanobis distance between q and b_{goal}. If this test succeeds, ConCERRT UPDATEs the policy π with all beliefs on the solution path, $\mathcal{B}_{\text{connected}}$ with all new beliefs that were connected to the goal, and $\mathcal{B}_{\text{open}}$ with all new partitions that are not yet connected to the goal, as described in Section IV-B.

Algorithm 2 T.EXPAND()

Input: $\mathcal{B}_{connected}$
1: $q_{\text{rand}} \leftarrow \text{RANDOM\_CONFIG}()$
2: $b_{\text{near}} \leftarrow \text{NEAREST\_NEIGHBOUR}(q_{\text{rand}}, T)$
3: $u \leftarrow \text{SELECT\_ACTION}(q_{\text{rand}}, b_{\text{near}})$
4: $b' \leftarrow \text{SIMULATE}(q_{\text{rand}}, b_{\text{near}}, u)$
5: **if** IS_VALID(b') **then**
6:   $\mathcal{B}_{\text{contingencies}} \leftarrow \text{BELIEF\_PARTITIONING}(b')$
7:   **for all** $b'' \in \mathcal{B}_{\text{contingencies}}$ **do**
8:     $T \leftarrow T.\text{ADD\_BELIEF}(b'')$
9:     $T \leftarrow T.\text{GOAL\_CONNECT}(b'', \mathcal{B}_{\text{connected}})$
10: **return** T

V. Experiments and Results

We evaluate the planner in simulation and real world experiments to show that a) ConCERRT scales up to high dimensional problems compared to other belief space motion planners. b) the contingency branches of ConCERRT allow to solve problems with significantly higher uncertainty than comparable non-contingent contact-based planners. c) ConCERRT policies are robust enough to be executed on real world systems. We implemented all experiments using the Robotics Library [26]. All experiments were carried out on a desktop computer with an Intel i5 3.5GHz processor. Table I gives the values of the planner's parameters for all experiments.

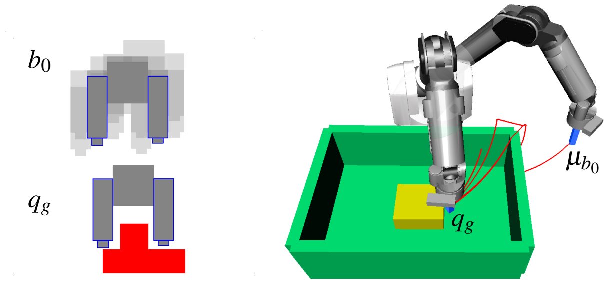


Fig. 5. *Left*: 2-DOF gripper problem with initial distribution b_0 and goal configuration q_g. The finger tips on the gripper (gray) can sense contact with the object (red). *Right*: 7-DOF problem. The right configuration shows the mean of the initial belief μ_{b_0}. The goal configuration q_g is inside the green container next to the yellow box. The blue rod at the end-effector is a force sensor. One policy computed by ConCERRT is shown with red lines.

Param.	**Description**	**2D gripper**	**7D sim robot**	**7D real robot**
t [min]	time budget	10	16.66	16.66
γ	contact/free-space bias	0.7	0.9	0.85
N	number of particles	50	40	100
δ_{step}	simulation step size	0.05	0.5	0.5
$p(b_g)$	goal bias probability	0.1	0.1	0.3
σ_{init}	initial uncertainty	$[\sigma,\sigma]$	0	$[2,2,2,3,3,3,3]\times 10^{-2}$
σ_{motion}	motion uncertainty	$[\sigma,\sigma]$	$[\sigma,\sigma,\sigma,\sigma,\sigma,\sigma,0]$	$[1,1,1,2,2,2,0]\times 10^{-2}$
ε_d	dist. threshold to goal	0.2	0.03	0.035

TABLE I
PLANNING PARAMETERS

A. *ConCERRT scales to high dimensional problems*

Most belief space planners rely on pre-defined discretization which lets them fail in complex environments. We now validate in simulation that ConCERRT scales to high-dimensional state space with complex contact states. The first experiment (Fig. 5 left) models a gripper with contact sensors on the fingers and the fingertips (similar to the setup in Koval et al. [24]). The gripper can translate in two dimensions. Compared to a similar problem from the POMDP literature [20], there are no outer walls limiting the workspace which increases the difficulty of the problem. In the second experiment (Fig. 5 right) a Barrett WAM 7-DOF arm must reach into a rectangular container and touch a yellow obstacle. To reduce uncertainty, it can measure the contact normal of the surfaces with a stick-shaped end-effector.

The results for the second problem (Fig. 7) prove that the planner is efficient enough to compute policies directly in 7-dimensional configuration space. The planner finds policies under significant motion uncertainty that slide along the walls of the container to localize the yellow box. We do not show results for uncertainties $\sigma > 0.4$, because the simulation of the sliding action becomes unreliable for extremely high motion uncertainties. Without a fixed discretization, this problem is not solvable for POMDP-based motion planners [20, 23] which become intractable under the larger number of DOF and complex contact manifolds. ConCERRT also handles a significant amount of motion uncertainty. This relaxes the assumptions of related approaches [24] which require the inverse kinematics of the robot and fully observable joint states. Thus, we are able to solve a larger set of problems. ConCERRT also relaxes the assumptions of Particle-RRT motion planners [14] as it does not require reversible actions and fully observable joint states.

B. *Contingent planning increases robustness*

Contingency plans capture many possible execution states and find appropriate reactions. Therefore, they should be more robust than plans without contingencies. To validate this, we ran the ConCERRT planner on the simulation scenarios (see Sec.V-A), varying the amount of uncertainty.

We compare our planner against two baselines. The first baseline is an uncertainty-unaware RRT-Connect with goal bias (RRTCon) [27]. The second baseline is the conformant CERRT planner [17]. To compare contingent and non-contingent planners we must define a suitable scoring function taking into account both the planning time and the quality of the plan. Therefore, we compute the score as $P_{\text{success}} = P(\pi)\cdot\frac{N_{\text{succ}}(t)}{N}$, where $P(\pi)$ is the success probability of the policy (equal to 1 for CERRT) and $\frac{N_{\text{succ}}(t)}{N}$ is the ratio of found solutions. We run 30 experiments per setup for the 2-DOF and 10 experiments for the 7-DOF problems.

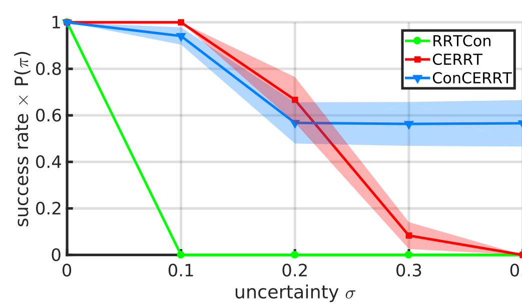


Fig. 6. Success rate of RRTCon, CERRT, and ConCERRT on the 2-DOF gripper problem as function of the position uncertainty. ConCERRT still finds solutions for $\sigma > 0.3$ where the conformant planner fails.

In Fig. 6 we show results for the 2-DOF problem. The RRT Connect always returns a trajectory which fails even with little uncertainty ($N_{\text{succ}} = 0$ for $\sigma \geq 0.05$). The solution quality of the CERRT planner drops to 0% while ConCERRT solutions' quality stays over 50%, even for the highest uncertainty.

The result in Fig. 7 shows that ConCERRT substantially outperform the baselines in terms of robustness to uncertainty if the dimensionality increases. In Fig. 8 we show how the solution quality improves over planning time for the 7-DOF problem with different values of uncertainty. Interestingly, low uncertainties do not necessarily lead to lower computation times for ConCERRT. We believe this is due to the planner committing to suboptimal contingencies too early, i.e. choosing a contingent plan when a conformant strategy would be possible which lead to one failed plan for $\sigma = 0.01$ and one for 0.5. The planner finds policies that succeed in

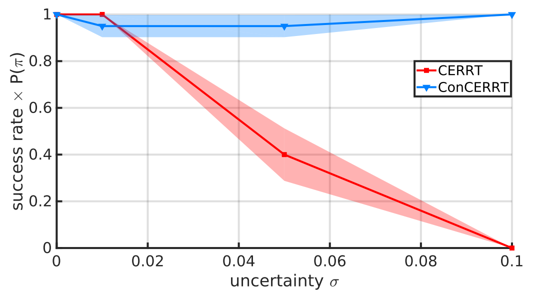


Fig. 7. Success rate comparison of CERRT and ConCERRT for a 7-DOF manipulator with force sensing. Our contingent planner can handle up to ten times higher uncertainty than a conformant planner.

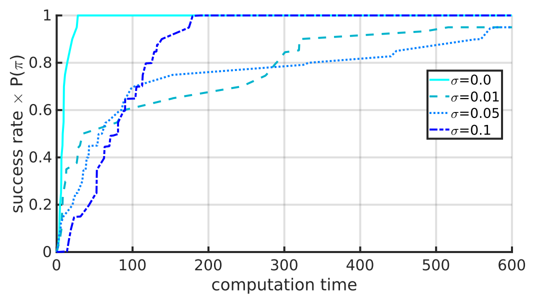


Fig. 8. The success probability for different ConCERRT policies with increased computation time. The solution quality of the policy increases over time until all possible contact events are covered.

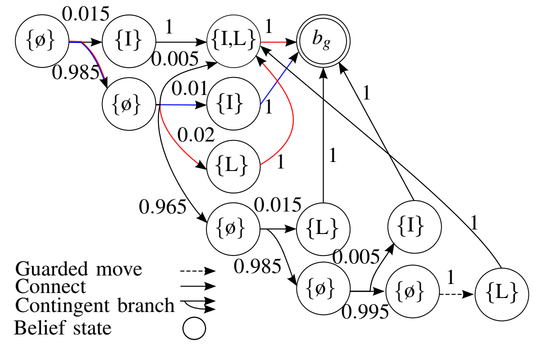


Fig. 9. The ConCERRT policy as executed on the real robot. Circles are belief states with contact state I (index finger), L (little finger), I,L (both fingers), or ø (no contact). The edges with the respective probabilities are actions that move the robot to the goal state b_g. The edge coloring shows the path taken by the robot in Fig. 10

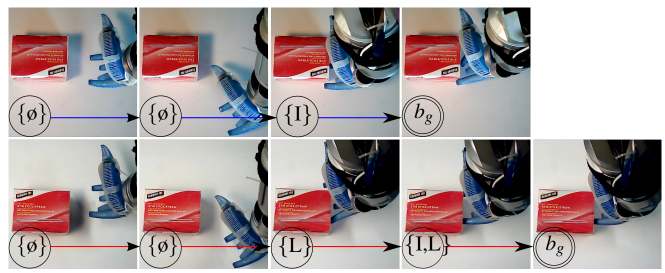


Fig. 10. Snapshots of two executions of the same policy found by the ConCERRT planner. *Top*: Box displaced $+2$ cm—the index finger makes contact first. *Bottom*: Box displaced -4 cm—the little finger makes contact first. The shown paths correspond to the blue/red edges in Fig. 9

50% of the runs under one minute. The policies improves as time goes on, approaching 100%. This shows the anytime property of ConCERRT.

C. Real robot experiment

To validate the policies generated by ConCERRT in a real world application, we applied the generated policies on a 7-DOF Barrett WAM robot arm with a soft hand (RBOHand 2 [28]) as end-effector. The experiment is inspired by the problem in Koval et al. [24] where a WAM arm equipped with a contact sensing hand localizes an object on a table surface. Similarly, our task is to sequence contact motions that reduce uncertainty enough such that the hand stops centered in front of the box. Fig. 1 shows the experimental setup. The fingers of a soft hand deform when they contact the environment which results in a measurable pressure change. We use large changes in pressure as a proxy for contact sensing. We only use the partially inflated index and little fingers as contact sensors. To find a policy we run the ConCERRT planner as initially but we exclude the slide action from planning as reliable sliding is hard to implement with a soft manipulator.

ConCERRT consistently finds feasible policies in 16.66 minutes. One computed policy π with $P(\pi) = 1.0$ is shown in Fig. 9. The policy executes multiple motions in front of the box, expecting no contact, however the policy also contains contingencies for the contact case. The most likely path through the graph does four of these free space motions and then executes a guarded move to ensure the final contact. To evaluate the robustness of the policy, we move the box 0, 2, 4, and 6 cm to the left and right relative to the hand's initial position. We execute the policy four times for each displacement while keeping the initial hand position constant. Fig. 10 shows an exemplary execution of two strategies with two different box displacement.

For the given uncertainty model, CERRT is not able to find a conformant solution, thus we can only compare the execution to a uncertainty-unaware planner such as RRT-Connect (RRT). Here we assume that the robot executes the RRT trajectory perfectly but is not aware of the moving box. Thus the position error of the RRT is equal to the position of the box relative to the hand. The results in Fig. 11 show that the selected ConCERRT policy is robust up to 4 cm uncertainty and it can handle 6 cm to the right but starts to fail when the box is moved further to the left.

Contact sensing via pressure sensors is not fully reliable. A wrong contact event triggered in six executions out of the 28 runs. In one case this failure could be detected automatically as the contact observation was not part of the policy. In another case, a wrongly detected contact triggered a false reaction, resulting in a high error in the final hand position (at +4 cm), while the other five cases were within the 3.5 cm error bound. We expect to mitigate these failures in the future by equipping the soft fingers with deformation sensors [29].

VI. Conclusion

We presented a novel contingent contact-based motion planner that combines the efficiency of sampling-based mo-

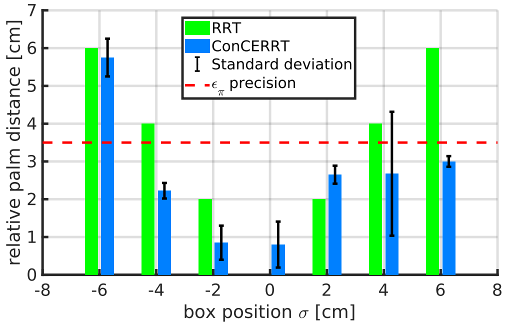


Fig. 11. Relative position of hand and object after execution of different policies for different object displacements. The real robot can localize the box until up to 4 [cm] position uncertainty.

tion planners with an efficient uncertainty reasoning. Our planner, Contingent, Contact-Exploiting RRT (ConCERRT), exploits contact sensing to anticipate informative contact events that can happen during execution of the plan under uncertainty. We demonstrated that the planned contingencies allow our planner to solve problems with higher uncertainty than comparable non-contingent planners. Compared to POMDP-based motion planners, ConCERRT scales to high dimensional problems with a non-trivial contact manifold. This is because our planner discretizes the state space during planning and only reasons about reachable contact events. We demonstrated this by successfully running experiments on a 7-DOF robot both in simulation and in the real world. There are some limitations to the ConCERTT planner, due to the assumption of a fully observable contact: The planner cannot deal with unreliable contact sensors and it needs edges in the environment to disambiguate its state. Additionally, the planner does not take path length into account and might compute long paths. However, it was shown before that asymptotically optimal planning is feasible for similar problems [16]. We believe this will also transfer to ConCERRT.

References

[1] L. Pryor and G. Collins, "Planning for contingencies: A decision-based approach," *Journal of Artificial Intelligence Research*, vol. 4, pp. 287–339, 1996.

[2] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[3] 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.

[4] 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.

[5] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 897–915, 2010.

[6] 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.

[7] 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.

[8] K. Y. Goldberg, "Orienting polygonal parts without sensors," *Algorithmica*, vol. 10, no. 2, pp. 201–225, 1993.

[9] 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.

[10] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[11] R. Platt Jr, R. Tedrake, L. Kaelbling, and T. Lozano-Pérez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems*, 2010.

[12] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 723–730.

[13] N. A. Melchior and R. Simmons, "Particle RRT for path planning with uncertainty," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2007, pp. 1617–1624.

[14] 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.

[15] C. Guan, W. Vega-Brown, and N. Roy, "Efficient planning for near-optimal compliant manipulation leveraging environmental contact," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 215–222.

[16] F. Wirnshofer, P. S. Schmitt, W. Feiten, G. von Wichert, and W. Burgard, "Robust, compliant assembly via optimal belief space planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[17] A. Sieverling, C. Eppner, F. Wolff, and O. Brock, "Interleaving motion in contact and in free space for planning under uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4011–4017.

[18] J. Zhou, R. Paolini, A. M. Johnson, J. A. Bagnell, and M. T. Mason, "A probabilistic planning framework for planar grasping under uncertainty," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2111–2118, 2017.

[19] 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.

[20] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Grasping POMDPs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 4685–4692.

[21] 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.

[22] K. M. Seiler, H. Kurniawati, and S. P. Singh, "An online and approximate solver for POMDPs with continuous action space," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2290–2297.

[23] M. Koval, D. Hsu, N. Pollard , and S. Srinivasa, "Configuration lattices for planar contact manipulation under uncertainty," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2016.

[24] M. Koval, N. Pollard, and 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.

[25] 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.

[26] M. Rickert and A. Gaschler, "Robotics library: An object-oriented approach to robot applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 733–740.

[27] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2000, pp. 995–1001.

[28] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *The International Journal of Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[29] V. Wall, G. Zöller, and O. Brock, "A method for sensorizing soft actuators and its application to the RBO Hand 2," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4965–4970.