

# Towards Motion Plans That React to Contact Events

Előd Páll   Arne Sieverling   Oliver Brock

**Abstract**—Intentional use of contact gains ground in the motion planning field by exploiting it to reduce uncertainty. Our approach extends contact motion planning to react to contact events. Our reactive planner augments the state space by taking in consideration continuous and discrete events. We differentiate discrete contact events that can occur during the execution due to motion uncertainty. This way, the planner has an increased understanding of the state space. Moreover, the planner generates strategies tailored for the expected events. Thereby the reactive plan becomes more powerful. We validate our assumptions about reactive planning, by extending a non-reactive contact-exploiting motion planner with reactive features. We compare both planners capabilities within a 2D grasping problem. We show that reactivity can deal with problems with high uncertainty where the non-reactive planner fails.

## I. INTRODUCTION

When robots move in the real world, their motion is unavoidably affected by (action, sensing, world model) uncertainty. There are two main approaches of addressing this uncertainty. Sampling-based motion planners often ignore uncertainty; this is by far the most frequently used approach to generating robot motion. In contrast, POMDP-based approaches model uncertainties explicitly and determine safe plans based on these models. These two approaches are two extremes on a spectrum: sampling-based methods are computationally efficient but ignore uncertainty, whereas POMDP-based approaches handle uncertainty but quickly become computationally intractable. Our goal is to be in the middle of this spectrum, hoping to combine computational efficiency of sampling-based approaches with some of the abilities of POMDP-based approaches to handle uncertainty.

The fact that sampling-based motion planners ignore uncertainty can lead to collisions during motion execution. Then we know that our plan has failed. But at the same time, collision, i.e. contact with the environment, provides important information that might be able to disambiguate the uncertainty that led to the collision. To realize this, a planner has to react to the contact event by updating its knowledge about the current state and by selecting the most appropriate plans for that updated state. We call such a planner "reactive" as it incorporates new information during the execution of the plan and reacts to that information by choosing appropriate courses of actions.

We will extend a motion planner that can plan contact events by incorporating the ability to include alternative sub-plans that are selected based on contact events.

An example of an informative contact event is shown in Figure 1. A gripper is actuated with motion uncertainty and the gripper can end up in contact with its left or right finger, e.g. state  $x_1$  or  $x_2$  respectively. First, a reactive planner would

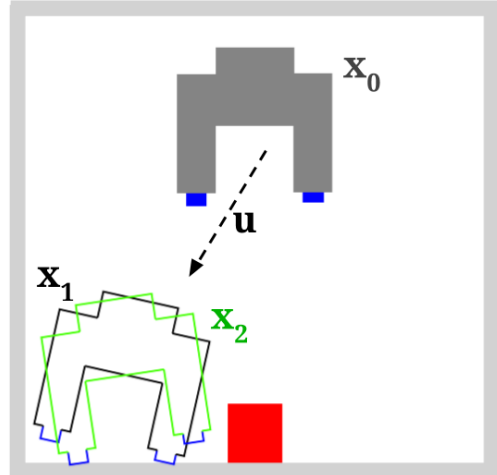


Fig. 1. Example for a gripper translating and rotating in the plane. The red square is the grasable object.  $x_0$  is the initial state.  $x_1$  and  $x_2$  are possible states after executing action  $u$  with rotational uncertainty. Our method disambiguates states  $x_1$  and  $x_2$  by using the feedback of the tactile sensors mounted on the fingertips (blue).

disambiguate states. The planner would differentiate  $x_1$  from  $x_2$  by detecting distinct contact signals for the left and right fingers respectively. Second, the planner handles the detected contact events separately. For example, the 2D gripper is in state  $x_1$ , then the gripper is rotated clockwise until the right finger reaches contact, while in state  $x_2$  the gripper is rotated counter-clockwise.

We evaluate our reactive planner's capability to generate alternative sub-plans for contact events on a grasp planning problem from the POMDP literature. We show how efficiency and robustness is achieved by solving grasping problems with motion uncertainty. Our results show that despite of substantially high uncertainty the reactive planner can find motion plans while the non-reactive version of the planner fails. This happens due to the fact that the reactive planner finds alternative strategies for contact events while the non-reactive planner considers those events as failures.

### A. Related work

Recent work in the contact motion planning field, Kaijen et. al [1] present grasping POMDPs where the gripper fingers have multiple binary contact sensors. Another work by Phillips-Grafflin et al. [3] implemented an RRT based motion planner to find robust strategies under significant motion uncertainty while using contact sensing. Sieverling et al. [4] took a step further and extended a sampling based motion planner not only with contact sensing but also with actions

that exploit contact to reduce uncertainty over the robot state.

## II. CONTACT-EXPLOITING RRT (CERRT)

We extend Contact-Exploiting RRT (CERRT) motion planner with reactivity. It is an RRT-based contact motion planner that finds manipulation strategies under robot state, motion, and world uncertainty by interleaving motion in free space with motion in contact, like sliding along a surface. The planner assumes free space motions increase state uncertainty, while contact motions reduce the uncertainty.

We describe two aspects of the CERRT motion planner: set of states and actions, because both will be the subject of our reactive approach.

The planner uses a combined state of belief over configuration and fully-observable contact  $x = (Q, C)$ , where the belief is represented a set of particles  $Q = \{q_1, \dots, q_N\}$  and the contact state is a set of surface pairs between the robot and the environment:

$$C = \{(s_{robot}, s_{world})_1, \dots, (s_{robot}, s_{world})_M\}$$

The CERRT planner integrates contact exploiting actions to increase the robustness of the plan. To do so, it uses three different actions:

- 1) *connect* is the free space motion action, it is a straight line connect between a node and sample in configuration space and it also enables the planner to break contact;
- 2) the *guarded* action moves in the desired direction until a new contact is established;
- 3) the *slide* action moves along a surface until the contact state is changed.

Both *guarded move* and *slide* facilitate motion in contact and they can reduce or keep low uncertainty in one dimension.

The planner uses random sampling and forward simulation to extend a node  $x_i$  while applying a motion error  $\delta_\alpha$  to each particle in the belief. The resulting state is added to the tree if the particle set is in a consistent contact state. As CERRT does not handle inconsistent contact states, it is possible that the planner fails to find a solution when:

- the initial state uncertainty is too high to execute any action successfully
- the motion uncertainty is too high
- the environment surface is highly fractured in small segments

To overcome these limitations, we extend the planner with reactive features which we will present in the following section.

## III. REACTIVE CERRT (R-CERRT)

We extended our previous contact motion planner CERRT toward a planner that reacts to contact events. We assume that the resulting reactive planner becomes more robust. In the following, we describe the main features of our approach that enable reactivity. The core elements of our reactive approach are:

- extending state awareness by using more powerful contact measurement than binary sensors,

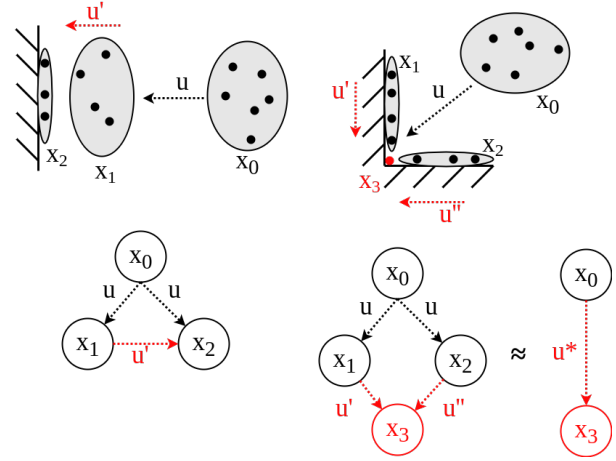


Fig. 2. **Top left:** a *connect* action result is almost in contact, the free space particle  $x_1$  cluster is merged to  $x_2$  with a *guarded* move. **Top right:** a *guarded* move results in two  $x_1$  and  $x_2$  clusters on different surfaces, they are merged in a new state  $x_3$  with a *sliding* action. **Bottom left** graph shows merging two existing cluster. **Bottom right** graph presents a merge into a new state.

- boosting the system model to handle continuous parameter space and discrete events, like configuration space parameters and contact sensor events respectively,
- alternative strategy generation for different contact events.

Our reactive approach considers not only motion uncertainty but also takes into account foreseeable contact sensory events. The contact events help to differentiate between the states that have inconsistent contact information. CERRT drops all this set of states with ambiguous contact state.

### A. State space

In order to disambiguate these states, we use richer contact perception. We extended the binary contact state description with the contact normal  $\hat{n}$ , in addition to the contact surfaces pairs:

$$C = \{(s_{robot}, s_{world}, \hat{n})_1, \dots, (s_{robot}, s_{world}, \hat{n})_M\}$$

We intend to build our plan on contact information that will be gained during execution. Thus, we can say that our reactive plan uses contact feedback on-line. We are going to map force torque sensor measurements to contact states. Because of this, the reactive planner can focus on minimizing the joint position uncertainty while planning and the contact state uncertainty will be reduced during the execution of the plan.

In order to react to contact events, our reactive planner detects and clusters the particles based on the extended contact state. If an action resulted in an inconsistent contact state, the belief is split and the new states have only matching contact state particles. We call the resulting new beliefs "sibling states" because they are from the same parent state and reached with the same action.

### B. Reacting to contact events

The detected contact events can be handled in different ways, like processing during planning or post-processing. Moreover, one can design various strategies to process the

sibling states, such as merging the sibling in an existing state or into a new one, or ignore them but use them in the planning process. Each approach has its benefits and shortcomings. We are not going to analyse the differences. In this paper we give a simple example how reactive behaviour can be integrated in a motion planner.

We define two strategies to handle the sibling states:

- merge siblings into one of the existing sibling state
- merge siblings in a new state

A strategy uses one of the existing actions: *connect* 1, *guarded* move 2, or *slide* 3. We define the primary sibling to be the belief in which we merge all the siblings, including the newly created state.

Both strategies are shown in Figure 2. On the left, the first strategy is illustrated where the robot is almost in contact with the surface. The particles in contact are in the primary sibling and the rest of the particles in free space are in a secondary sibling. The secondary sibling is merged into the primary with a *guarded* motion 2.

The second strategy is on the right of Figure 2, the state  $x_1$  and  $x_2$  are merged in a new state  $x_3$ , considered as the primary. The uncertainty is reduced in both dimensions with the *slide* action 3. The same slide action is applied on  $x_1$  and  $x_2$  if action  $u$  resulted in all the three sibling states  $x_1$ ,  $x_2$  and  $x_3$ .

However, we represent the sibling states with a graph, the planner processes the nodes as a tree, like shown on the bottom leftmost graph in Figure 2. We add only the primary sibling to the search tree if all siblings are successfully merged.

Finally, when the planner reaches the goal and the path from the start configuration includes a sibling state the resulting reactive plan includes all sibling states and merging actions. Thus, the reactive plan can have multiple branches in parallel as shown at the bottom of Figure 2.

#### IV. EXPERIMENTS AND RESULTS

We validated our assumption that reactivity increases the robustness of a contact motion planner. For this purpose, we conducted a qualitative comparison between the reactive CERRT and the non-reactive CERRT. Our aim is to show, reactivity can solve a realistic grasping problem where CERRT fails.

Thus, we compared the planning success rate of the reactive to the non-reactive CERRT on a 2D grasping problem in simulation as shown in Figure 1. The task is to move the end effector such that the red rectangle to be between the fingers of the gripper. The end effector has three degrees of freedom: translation in  $X$  and  $Y$  axis and rotation. The contact sensors are placed at the end of the grippers fingers. The sensors are represented by the blue boxes and can sense with all of its three sides the contact force with the environment (gray walls) or the object (red rectangle).

We considered no motion uncertainty and the initial position uncertainty is normal distributed with a relatively low standard deviation. This way we facilitate uncertainty reduction with contact exploiting motions for both planners. In order to

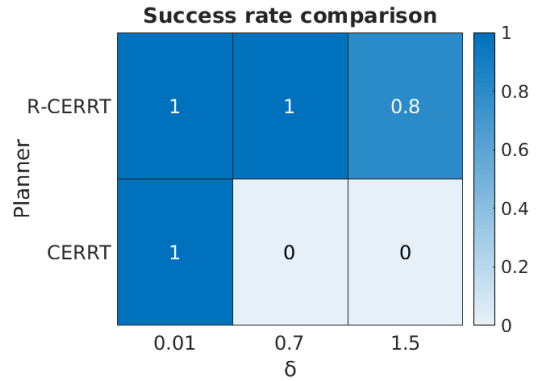


Fig. 3. Success rate of CERRT and Reactive CERRT.

compare robustness of the solutions, we varied the standard deviation of initial angular position uncertainty, because this state variable increases the frequency of inconsistent contact states.

For each experiment we set the CERRT’s weight between free-space and contact motion to  $\gamma = 0.7$ , which results in increased sliding motion. The standard deviation of the initial position is 0.01 while the standard deviation of the initial orientation is  $\delta \in \{0.01, 0.7, 1.5\}$ . We ran ten experiments for each  $\delta$  with the same computation time budget, 600[sec] for both planners.

The average success rate is shown in Figure 3. As expected, both planners can solve the problem for low uncertainty. As the initial uncertainty increases, CERRT fails while R-CERRT still finds a solution with high success rate. The reason is that CERRT can’t find an action that leads to a consistent contact state, while R-CERRT can differentiate those states.

The increased success rate comes at a cost of higher computation time. We already observed in CERRT that forward simulation of motion in contact is the most time consuming operation. Note that all sibling merging actions aim to use contact. Thus the reactive CERRT is more computation intensive. The higher computation load of the reactive planner is observed in the total number of nodes expended. R-CERRT expends three times less nodes than CERRT.

Nonetheless, the results show that for the same planning budget, the reactive planner is able to find a solution while CERRT fails for high initial uncertainty. The reason is that random sampling is not enough to handle the high rotational uncertainty while disambiguation with simple alternative strategies compacted in the reactive planner is more powerful.

#### V. CONCLUSION

In this paper we described how a contact motion planning algorithm can benefit from reactivity. Reactive planning builds on the knowledge that some state uncertainty can be diminished during the execution of the plan by sensing contact events.

We incorporated reactivity in the Contact-Exploiting RRT, a non-reactive motion planner. Our reactive planner is able to anticipate events and react appropriately by integrating alternative strategies into the motion plan.

We validated the resulting reactive contact motion planner on a 2D grasping problem in simulation. We compared the success rate of the reactive planner to the non-reactive CERRT. The results show increased robustness in spite of increased uncertainty. We believe there is room for improvement like detecting or learning from experience what is an informative contact event and how these event should be efficiently handled. We believe that our reactive contact motion planner successfully combined computational efficiency of sampling based approaches with the ability to handle uncertainty to some extent as POMDP-based approaches handle uncertainty.

#### ACKNOWLEDGEMENTS

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

#### REFERENCES

- [1] Kaijen Hsiao, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Grasping POMDPs. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4685–4692, 2007. ISSN 10504729. doi: 10.1109/ROBOT.2007.364201.
- [2] Steven M LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.
- [3] Calder Phillips-Grafflin and Dmitry Berenson. Planning and Resilient Execution of Policies For Manipulation in Contact with Actuation Uncertainty. *Workshop on the Algorithmic Foundations of Robotics*, 2016.
- [4] Arne Sieverling, Clemens Eppner, and Oliver Brock. Interleaving Motion in Contact and in Free Space for Planning Under Uncertainty. *under reviewing in International Conference on Intelligent Robots and Systems*, 2017.