Project description—Resilient motion planning

Background

Following recent discussions on the science of resilient robotic autonomy [1], we will investigate the problem of *resilient motion planning*. Here, we are interested in computing a path that can still be executed even if one of the robot's system's fails. Specifically, we will look at robot manipulators which often have some redundancy built in. For example, to reach any point in space, we are required to have a six degree of freedom (DOF) arm. However, many manipulators are endowed with seven DOFs. This extra flexibility allows a planner to choose between many options when attempting to connect two points. Typically, one option is chosen and the rest are discarded. In this project we will investigate how this redundancy can be used in a systematic way in order to achieve a resilient motion planner.

Problem formulation

We will consider the problem of motion planning for a redundant manipulator with d degrees of freedom. Given two configurations q_1, q_2, a local path ℓ connecting q_1 and q_2 is said to be *resilient* to the i'th degree of freedom if

$$\forall \tau \in [0,1], \ell^i[\tau] = \ell^i[\tau].$$

Namely, if the i'th degree of freedom does not change along ℓ (here $\ell^i[\tau]$ denotes the i'th degree of freedom of the configuration along ℓ at time τ). Similarly, a path $\pi = q_1, \ldots, q_n$ is said to be *resilient* to the i'th degree of freedom if all local paths connecting q_i to q_{i+1} are resilient.

Now, let Q_{free} be a description of the free space and let q_s and $Q_t \subset Q_{\text{free}}$ be a start configuration and set of target configurations, respectively. Let p be the probability that after a certain action is taken, one of the robots degrees of freedom fails and stays fixed. Our problem calls for computing a plan connecting the start and the target configurations that maximizes the probability to reach $q_t \in Q_t$. In contrast to the standard motion-planning problem where a plan is represented as a path, here the output will be a policy

$$\pi : Q_{\text{free}} \times 2^{[d]} \rightarrow Q_{\text{free}}.$$

Namely, given a configuration q and a set of failed degrees of freedom $J_{\text{failed}} \subseteq [d]$ the policy π outputs a configuration q' such that the path connecting q and q' is resilient to every failed degrees of freedom in J_{failed}.

Notice that this formulation naturally balances path length (longer paths have a higher probability of some failure occurring along the path and resilience (longer paths may be more resilient).

Solving the problem via sampling-based methods

Our approach will be inspired by the RRT algorithm. We build a meta tree $\mathcal{T}$ rooted at q_s. At each iteration, we sample a random configuration q_{rand} and find q_{near}, the nearest neighbor of q_{rand} in $\mathcal{T}$.

Now, to be able to compute a resilient path from q_{near} towards q_{rand}, we consider a ball $\mathcal{B}_\delta(q_{\text{rand}})$ of radius δ centered at q_{rand}. We then proceed to compute a set of resilient paths connecting q_{near} to any configuration in $\mathcal{B}_\delta(q_{\text{rand}})$ and add them to $\mathcal{T}$. The algorithm terminates after a certain time limit has been exceeded. Once $\mathcal{T}$ has been computed, we need to run a search algorithm to compute the policy π.

Project goals

1. Given a tree $\mathcal{T}$ together with a label on each edge stating the resilience of each edge, suggest a search algorithm to compute a policy π.
2. Implement the algorithmic framework for a hyper-redundant planar manipulator (say with $d = 4$ degrees of freedom).
3. Evaluate the performance of your algorithm on different benchmarks. What are the computational bottlenecks? Can you suggest alternative algorithms¿

References

[1] Kostas Alexis. Towards a science of resilient robotic autonomy. *CoRR*, abs/2004.02403, 2020.

[2] Benjamin J. Cohen, Sachin Chitta, and Maxim Likhachev. Search-based planning for manipulation with motion primitives. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2902–2908, 2010.