Proposal

I. What are you trying to do?

The project will involve single-arm motion planning with a possible extension to two-arm collaboration. The objective is to have the arm reach a specific desired location while avoiding obstacles and then perform specific tasks of interest like welding, hammering, or drilling. For example, given the wall blocking the arm from reaching the welding spot, the arm will position itself around it and then set several locations on the object of interest touching each of the locations without colliding with the fixture it is mounted on, the wall, or the object of interest. The welding spots will have a predefined curve of possible orientations and positions for the end effector of which the arm can choose the most feasible and fastest path to achieve the objective. The arm will be in the welding position for about three seconds at which point it will proceed to the next location. The goal is to find the sequence of control actions that reduces the overall time and optimizes the motion trajectory while achieving the primary objective of collision avoidance. The possible extension is to have two arms collaborating in a way where one arm is holding the bar to be welded while the other arm is performing the spot welding such that the object is fused to the surface.

II. What is your baseline and how is it lacking?

The current baseline is based on a heuristic search, mostly A* and RRT*. But currently, they haven't explored the N DoF systems with N-1 joints avoiding multiple objects at the same time. They mostly talk about using A* to find a path to the goal but don't talk much about how the whole arm can avoid obstacles, even if they do they generally have a very simple environment. This type of planning can be quite lacking in real-world settings as discussed above in the case of welding or using any form of power tools with consideration of the environment for example using a hammer in a hidden place inside hardware. We need to avoid all the obstacles and the joints need to avoid the obstacles and plan a safe path for all the joints and links. Current baselines do not achieve this objective. Some baseline optimizations like MPC etc, are computationally very expensive and are difficult computationally for high dimensional space and even for highly dynamic obstacles. So the current baselines we are making our models against are heuristic-based approaches like A*, sampling-based ones like RRT*, and optimization-based ones like MPC.

III. *What is your alternative/improvement and why do you think it will work?*

The improvement to the current baseline includes potentially learning the objective function that maximizes the distance of the robot arm from the obstacles that are represented with a certain map like Octomap. The potential solution can be to maximize the Euclidean distance of the robot arm to the obstacles as a primary objective function while reaching the goal penalizing the longer duration to reach the target position. We will investigate further methods that could represent the environment such that we can include that into our motion planning algorithm to perform the trajectory while avoiding obstacles. We can investigate methods that deal with SDF(signed distance function) or CDF (Configuration distance function) and use them to make a secondary control objective for designing null space controllers to avoid obstacles in the environment.

Some sampling-based approaches include:

1) Learning-aid graph search algorithm. For search-based algorithms like A* and Dijkstra, there is always a process of discretizing the free space before the search begins. If we discretize the space uniformly the graph is either going to be too coarse or too fine which means end up not finding the optimal path or using too many computational resources. If for different regions we can have different resolutions according to the planning objectives this problem can be solved. Therefore we propose that we train a deep neural network that can help us find different resolutions for different spaces.
2) Learning-aid-sampling-based search. As for sampling-based search, we can use the same general idea to build up a corresponding deep neural network to help us do a better job in the sampling process.

IV. *How will you simulate the problem?*

The problem will be simulated using PyBullet and Franka Emika Panda's seven degrees of freedom robot. We will explore other simulators like Gazebo Ignition, Mujoco, and Maniskill and based on the application choose the most appropriate one for the project.