COSC69.13/269, Spring Term 2021, PA4 - Auction-based task allocation, Anmol Chachra

My ros package multirobot_auction_tasks implements sequential acution mechanism in a multi robot setting. It has an auctioneer and multiple auctionees/bidders that collectively make up an auction. By default, there is 1 auctioneer and 3 auctionee (auctioneer itself is an auctionee + 2 others). Auctioneer randomly creates and publishes N waypoints, where N is the number of robots in the environment, and use 'sequential auction' algorithm to allocate each of them tasks. Note that a robot can be assigned multiple tasks.

Method description

When you run the launch file, 3 robots are spawned and each of them runs a tf broadcaster node that broadcasts their inital position in 'world' reference frame accessible via 'odom' topic. Additionaly, 1 robot (robot_0) runs the 'auctioneer' node and other 2 (robot_1 and robot_2) runs the 'auctionee' node.

Concept starts at the auctioneer node where 'Auctioneer' cass object is initialized and then sets and publishes the randomly generated waypoints, that are atleast 'd' distance apart. These randomly generated waypoints are then read by the subscriber in the auctionee node. This indicates them to enter the auction and use those waypoints to make and send bids. The bids and the robot name making those bids are sent using the 'BiddingService' in the Auctioneer node. When the bids are received by the auctioneer, it updates the bidding pool, note down the robotname in a set. When the auctioneer receives N - 1 bids, where N is the total number of robots spawned, it proceeds to allocating the tasks.

Task allocation is done based on the smallest bid for the available tasks. To acieve this Auctoneer maintains a list of 1s and 0s with size equal to number of tasks, where 1 being task is available and 0 otherwise. It iteratively selects the minimum bid, the robot name and the task associated with that bid. If the robot name is the auctioneer itself, it makes itself go to that waypoint and updates its position and orientation for fresh bids in the next round; else, it sends that task number to respective robot running the Auctionee node and the action server, where the tasks are queued for execution. Note that in my design, the auctioneer if receives a task executes it first, whereas if the auctionee/biddere received tasks are queued and executed after all the tasks are alloted.

Evaluation

Program was evaluated for 3 robots in gazebo where auctioneer is robot_0 and bidders/auctionee are robot_1 and robot_2. Their positions are (-2, 1, 0), (2, 1, 0) and (0, 2, 0) respectively. Their orientations are (0, 0, 0), (0, 0, 1.57) and (0, 0, 1.57) respectively. My program works really well and it can be verified by checking the screen output. Also, please see the demo video in the media folder for demonstration.