Robotics: Assignment I

Pioneer 3-DX Simulation

Biomechatronics Engineering, Li-Wei Yang, B07611001

Demo videos link:

https://drive.google.com/drive/folders/1J5BcP0GtEsNWr45t_ejvX1Q2X2li61UL?usp=sharing

Part C: Keyboard Control

Design philosophy: set translational acceleration = 50/-50 mm/s^2 when press “i/,” ; set rotational acceleration = 10/-10 degree/s^2 when pressed “j/l”. When press “k”, the translational and rotational velocity are set to 0. The maximum translational velocity is 500/-500 mm/s; the maximum rotational velocity is 90/-90 degree/s. When the robot is moving, translational deceleration = -20/20 mm/s^2, rotational deceleration = -5/5 degree/s^2, to simulate friction force.

Part D: Collision Avoidance using Sonar Sensor

Design philosophy: the moving method is the same as described in part C except the maximum translational velocity is 300/-300 mm/s. When any of the 16 sonars detect objects close to the robot (range < 600), the maximum translational velocity is set to 50/-50 mm/s; the maximum rotational velocity is set to 10/-10 degree/s, to prevent collision.

Part E: Robot Pose and Odometry

Design philosophy: First calculate the angle robot should rotate to face the destination, as well as the distance the robot should move. Second, set the heading of the robot to face the destination, then move the distance calculated in the first part. When translate is done, orientate the robot to face Th. then the process is done. The user can set multiple times of destinations, if the user choose to close the simulation, the robot would go to initial position.

Odometric pose is different form the true pose because the origin of the two are different. Odometry coordinates are relative to the robot itself, while true coordinates are relative to the world origin. To align the two, the odometric pose need to be offset.

Bonus Round: Obstacle Avoidance

Design philosophy: an action group called avoid has different actions in different priority. If the robot stalls, the robot will back up and recover. If the bumper on the robot hit obstacles, the robot would back up and turn. With two AvoidFront action, the robot can avoid obstacles near and far (225 and 450). If the above action is not triggered, the robot would move to the given goal. I was trying to let the user to input goals multiple times, just like part_e, yet the action GoToStraight seems have some problem so the program can only input goal once.