Weekly Report 2

Lukas Evers

July 26, 2022

1 Collection of Scenarios for Autonomy and Safety Tests

1.1 Scenarios Derived From Intel RSS Guidelines

Table 1: RSS Guideline Scenarios

Name of Scenario	**RSS Rule**	**Description**
SafeDistance	Safe linear and lateral distance	An obstacle gets too close to the robot. Obstacle gets spawned near the current location of the robot
RightOfWay	Right of way is given	Another robot drives into the path of the robot.
Occlusions	Limited Visibility	The robot drives through a densely occupied area with many occlusions in the lidar data ahead.
CollisionAvoidance	Avoid collisions	Robot performs a dodge maneuver if anything gets too close in the zone what is considered to be a safe distance.

1.2 Scenarios Derived Fom Sense - Plan - Act - Principle

Table 2: Sense-Plan-Act Scenarios

Name of Scenario	**Activity**	**Description**
LidarCrash	Sense	Crash of the Lidar node.
OdomCrash	Sense	Crash of the Odometry/Localization node.
NoPathFound	Plan	No valid path to goal is found.
PlannerCrash	Plan	Path Planning Node crashed.
ForcedCollision	Act	A collision with an obstacle is forced.
BatteryLow	Act	The battery is too low to at the destination.
MotorFailure	Act	The motor has a hardware failure.
NetworkFailure	Act	The method to transmit the motor commands crashes.

2 Current Behaviour vs. Planned Behaviour during Scenarios

Table 3: Current Behaviour vs. Behaviour Planning Extension

Scenario	**Current Behaviour**	**With Behaviour Planning**
SafeDistance	Robot replans to avoid the obstacles, but does not factor in the speed into what is considered safe.	The robot adapts the allowed safe distance to the current speed.
RightOfWay	The robot would replan the path around the around but not predict where things will be moving.	The robot would calculate the point of collision and search for strategies to avoid the crash in the future. Slowing down, slowing down hard, or planning around.
Occlusions	Robot is unaware of occlusions and sudden appearances of dynamic obstacles	The robot slows down in areas with occlusions.
CollisionAvoidance	The robot would replan the path around the around but not predict where things will be moving.	The robot would calculate the point of collision and search for strategies to avoid the crash in the future. Slowing down, slowing down hard, or planning around.
LidarCrash	The robot would continue to drive, but it has no information about the environment anymore.	The robot would continue to drive with lowered speed and try to restart the Lidar. Taking into consideration the motion prediction of obstacles. After not receiving information for a longer period the robot would come to a stop.
OdomCrash	Robot would drive indefinitely and never reach the goal	The robot tries to change the method of localization and to meanwhile restart or reset the odometry.
NoPathFound	The robot would never move.	The robot tries to restart the planner and replan. Else, it tries to change the path planning algorithm. If planning is still not possible the robot tries to move to a different starting position.
PlannerCrash	The robot would never move.	The robot will reuse the last path and drive slower and restart the path planning node.
ForcedCollision	The robot would not be able to move away from the crash.	The robot slowly reverses out of the crash and resets navigation and odometry at the last known safe location.
BatteryLow	The robot would take goals and execute the commands even though the battery would fail during the driving.	The robot would calculate if the battery would be enough to make it to the goal and only then would start to drive.
MotorFailure	The robot would not be able to move.	The robot would not be able to move.

3 Robot Definitions

- Turtlebot 3 Waffle
- ROS2 Foxy
- Ubuntu 20.04
- Lidar
- Wheel Odometry
- Raspberry Pi Camera
- Add an additional IMU Sensor in the Simulation

4 Next Week

- Work on creating the first BT nodes with ROS Interfaces
- Make concrete Sub-Scenarios
- Create a template for scenario creation (Launch file(s), Gazebo Setup, etc.)