EVER-V-Competition

Task Two

LiDAR Sensors:

LiDAR (or Light Detection and Ranging) technology is an ideal solution for examining the surface of the earth, assessing information about the ground surface, or creating a digital twin of an object, Laser scanning systems employ this technology, using ***LiDAR*** data to map 3D models and digital elevation.

LiDAR has several advantages over other sensors for robot navigation. First, ***LiDAR*** is accurate and reliable, as it can measure distances with sub-centimeter precision and is not affected by ambient light or weather conditions. Second, ***LiDAR*** is fast and responsive, as it can capture and process data in real time and cover a wide range of angles and distances. Third, ***LiDAR*** is versatile and adaptable, as it can work with different types of robots and environments and can be integrated with other sensors and algorithms.

One of the main challenges for robot navigation is to perceive the environment and identify the relevant features and obstacles. ***LiDAR*** can help robots achieve this by providing rich and detailed information about the geometry and structure of the scene. For example, ***LiDAR*** can help robots detect walls, floors, ceilings, doors, windows, furniture, and other objects. ***LiDAR*** can also help robots classify and segment objects based on their shape, size, color, and texture. Furthermore, ***LiDAR*** can help robots track the motion and position of dynamic objects, such as people, animals, or vehicles.

Another challenge for robot navigation is to plan the optimal path and actions to reach a goal or perform a task. ***LiDAR*** can assist robots in this process by providing a 3D map of the environment that can be used for localization and mapping. For example, ***LiDAR*** can help robots determine their location and orientation in relation to a global coordinate system or create a local map of their surroundings that can be updated as they move. ***LiDAR*** can also help robots generate and evaluate different trajectories and maneuvers that can avoid collisions and optimize efficiency and safety.

The final challenge for robot navigation is to execute the planned path and actions with precision and smoothness. ***LiDAR*** can enhance robots' performance in this aspect by providing feedback and control signals that can adjust the robot's speed, direction, and posture. For example, ***LiDAR*** can help robots measure their distance and angle to the target or obstacle and correct their errors or deviations. ***LiDAR*** can also help robots sense and adapt to the terrain and surface conditions and modify their gait or mode of locomotion accordingly.

Path Planning:

Path planning lets an autonomous vehicle, or a robot find the shortest and most obstacle-free path from a start to goal state. The path can be a set of states (position and/or orientation) or waypoints. ***Path planning*** requires a map of the environment along with start and goal states as input. The map can be represented in different ways such as grid maps, state spaces, and topological roadmaps. Maps can be multilayered for adding bias to the path.

<u>Path Planning Algorithms:</u>

Grid-based search algorithms, which find a path based on minimum travel cost in a grid map. They can be used for applications such as mobile robots in a 2D environment. However, the memory requirements to implement grid-based algorithms could increase with the number of dimensions, such as for a 6-DOF robot manipulator.

Sampling-based search algorithms, which create a searchable tree by randomly sampling new nodes or robot configurations in a state space. Sampling-based algorithms can be suitable for high-dimensional search spaces such as those used to find a valid set of configurations for a robot arm to pick up an object. Generating dynamically feasible paths for various practical applications makes sampling-based planning popular, even though it does not provide a complete solution.

Trajectory optimization algorithms, which formulate the path planning problem as an optimization problem that considers the desired vehicle performance, relevant constraints, and vehicle dynamics. Along with generating dynamically feasible trajectories, they can also be applied for online path planning in uncertain environments. However, depending on the complexity of the optimization problem, real-time planning can be prohibitive.

Path planning, along with perception (or vision) and control systems, comprise the three main building blocks of autonomous navigation for any robot or vehicle. Path planning adds autonomy in systems such as self-driving cars, robot manipulators, unmanned ground vehicles (UGVs), and unmanned aerial vehicles (UAVs).

Path Tracking:

A ***path tracking*** control system developed for autonomous mobile robots driven by wheels is described. In conventional approaches, the path is usually planned by smooth curves with curvature-continuity and a path tracking controller is independently designed to compensate the path error occurring in the navigation. However, smooth path planning is difficult to execute on-line due to the computational burden. In addition, the conventional ***path tracking*** algorithm often causes unpredictable tracking motion when large path error occurs.

- ***Vehicle Path Tracking Using Pure Pursuit Controller:***

The Pure Pursuit algorithm is a simple, yet effective control strategy used to guide a vehicle along a desired path by continuously adjusting its steering angle based on its current position relative to a reference path.

Reference Path Generation: A reference path is defined, typically represented as a series of waypoints or a smooth curve, that the vehicle should follow.

Sensing and Localization: The vehicle utilizes sensors such as GPS, lidar, radar, or cameras to determine its current position and orientation relative to the reference path. This information is often obtained through localization algorithms such as SLAM (Simultaneous Localization and Mapping).

Target Point Selection: The Pure Pursuit algorithm selects a target point on the reference path ahead of the vehicle. This point is usually a certain distance (lookahead distance) along the path. The selection of this point aims to minimize the lateral error (distance between the vehicle and the path).

Steering Control: Based on the selected target point and the current position of the vehicle, the Pure Pursuit controller calculates the appropriate steering angle required to navigate towards the target point. This calculation takes into account factors such as the vehicle's kinematics and dynamics.

Feedback Control Loop: The steering angle is continuously adjusted based on feedback from the sensors, ensuring that the vehicle follows the reference path accurately.

Execution: The calculated steering angle is then sent to the vehicle's steering system to control its direction.

- ***Vehicle Path Tracking Using Stanley Controller***

Vehicle path tracking using the ***Stanley controller*** is another method commonly employed in autonomous vehicle navigation, similar to the Pure Pursuit controller. The Stanley controller, also known as the Stanley Method or Stanley Steering.

Reference Path Generation: As with Pure Pursuit, a reference path is defined for the vehicle to follow. This path can be represented as a series of waypoints or a smooth curve.

Sensing and Localization: The vehicle utilizes sensors such as GPS, lidar, radar, or cameras to determine its current position and orientation relative to the reference path. This information is typically obtained through localization algorithms like SLAM.

Error Calculation: The Stanley controller calculates the cross-track error, which is the lateral distance between the vehicle's current position and the reference path. This error is used to determine the corrective steering angle needed to bring the vehicle back on track.

Control Law: The Stanley controller employs a control law that combines the cross-track error and the heading error (difference between the vehicle's current orientation and the tangent direction of the reference path at the nearest point) to calculate the steering command.

Steering Control: Based on the calculated steering command, the vehicle's steering system adjusts the steering angle to guide the vehicle back onto the reference path.

Feedback Control Loop: The steering angle is continuously adjusted based on feedback from the sensors, ensuring that the vehicle stays on course.

The Stanley controller offers advantages such as simplicity and robustness in handling various road conditions and environments. It can effectively handle sharp turns and complex paths.