Milestone Three

System Integration

ARTICLE 1: Objective

The third milestone focuses on the integration of previously learned concepts—perception, control, and sensor filtering—into a comprehensive autonomous driving system. The goal is to enable the autonomous vehicle to navigate dynamically and respond to real-world driving scenarios. Specifically, the system should be capable of:

-Obstacle Detection and classification: The vehicle must identify and react to static and dynamic obstacles, such as pedestrians, other vehicles, and traffic cones, ensuring safe navigation.
-Adaptive Cruise Control: The vehicle should be able to follow a lead car, maintaining a safe distance, and adjusting its speed appropriately.
-Lane Change Maneuvers: When encountering slower-moving vehicles in the same lane, the system should execute safe and efficient lane changes to maintain a smooth flow of traffic.
-Emergency Stops: The vehicle should be able to detect unexpected obstacles (e.g., a pedestrian suddenly crossing the road) and perform an emergency stop to avoid collisions.

ARTICLE 2: Requirements

You are required to deploy your system in the previously built tracks while following the following regulations.

Straight line track: Implement a straight-line track where the vehicle must detect and stop for a pedestrian crossing the road and then continue once the path is clear. you must set at least 5 pedestrians in random positions on the track

Lane Change Track: Implement a track where the vehicle moves a in a straight line and stops for a pedestrian crossing the road, changes lanes to avoid a slower-moving vehicle or traffic cones, and then continues and stops for a pedestrian crossing the road then continue.

Circular Track: Implement a circular track where the vehicle navigates the path while avoiding randomly placed static obstacles such as traffic cones and responding to dynamic obstacles like pedestrians.

Custom track creation:

You must design and implement a unique track that incorporates various challenges (e.g., mixed static and dynamic obstacles, complex navigation patterns) and demonstrates their ability to handle unpredictable scenarios.

The track can be part of your campus, part of your living block, any road intersection, Note that: Complicated designs is not necessary but it needs to be good enough to test all the capabilities of your car as mentioned above. You should describe your design choices and the rationale behind them.

Each team is required to build a map of their own design that reflects part of the real world (part of campus, part of your living block, road intersection) the map doesn't need to be accurate nor complicated.

ARTICLE 3: Submission criteria:

1- You are required to record simulation videos for the re-implemented tracks and the innovative track, showcasing how the vehicle handles the specified scenarios.

2- You are required to submit a technical report that includes the following:

- System Overview: Describe the integrated system and how each module contributes to handling the specified scenarios.
- Track Descriptions: Detail the implementation and challenges of each re-implemented track and the innovative track.
- Algorithm Description: Explain any modifications to the algorithms used for the additional complexities.
- Performance Analysis: Analyze the system's performance on each track, while providing graphs that validates the performance of your system, discussing any challenges encountered and how they were addressed.

3- You are required to prepare a presentation that summarize the objectives, approach, implementation, and results of your work.

Also prepare a demo walkthrough of the simulation videos, highlighting key features and behaviors of the autonomous system on each track.

Note: All plotting, and graphs must be made using MATLAB software, also you can use it to tune your controller and filter the noises.

ARTICLE 4: Notes

- Codes will be subjected to a plagiarism checker to validate authenticity.
- No late submissions will be accepted.

ARTICLE 5: Simulation

Mounted Sensors:

- Odometer sensor.
- Mono-Camera with resolution 960x480.
- Velocity sensor.
- Velodyne-32 lidar

Topics:

```
/SteeringAngle #subscriber
#  Type: std_msgs/Float64
#  info: Used to set the steering angle
#  uint: Degree
/cmd_vel #subscriber
# Type: std_msgs/Float64
# info: Used to set the force on gas pedal range from 0~1 (0 for no pressure
on gas pedal and 1 for full pressure on gas pedal).
/odom # publisher
# type: nav_msgs/Odometry
# info: Used to publish the position and velocity of the vehicle
# uint: (position is in meter realtive to the world) (orinetation is quatrion
and you must transoform it into euler) (Velocity is in meter/sec)
/Imu
# type: sensor_msgs/Imu
# info: Used to publish the linear acceleration and angular velocity and
angular position of the vehicle
# uint: (linear acceleration is in m/s^2 realtive to the world) (angular
velocity is rad/sec)
/velodyne_points
# type: sensor_msgs/PointCloud2
# info: Used to publish Lidar points cloud
# uint: point distance is in meter
/brakes
# type: std_msgs/Float64
# info: Used to brake apply brake pressure from 0.0 to 1.0
# unit: No unit
/startSimulation
# type: std_msgs/Bool
# info: Used to start the simulator if the the message is "TRUE"
# unit: No unit
/pauseSimulation
# type: std_msgs/Bool
# info: Used to pause the simulator if the the message is "TRUE"
# unit: No unit
/stopSimulation
# type: std_msgs/Bool
# info: Used to stop the simulator if the the message is "TRUE"
# unit: No unit
/image
# type: sensor_msgs/Image
# info: Used to publish the image from the camera in the simulator
# unit: No unit
```

Link of the simulator:

The simulator is available on EVER Academy at Udemy Platform in Autonomous Track Milestone Three.