Autonomous Car Simulation using CARLA

Sachin G Holalkeri[1] , Shreevadan K[2] , Shreyas S M[3] , Sudarshnan D[4]

[1 - 3] UG Students, *Department of ISE, Sapthagiri College of Engineering, Bengaluru, Karnataka - 560057, India.*

[4]*Assistant professor, Department of ISE, Sapthagiri College of Engineering, Bengaluru, Karnataka - 560057, India.*

sachinholalkeri@gmail.com, shreevadan10@gmail.com, shreyassingri@gmail.com, sudarsanand@sapthagiri.edu.in

Abstract—
The Autonomous Car is currently 20 million dollars capital. It is expected a growth of 63.1% in the next 10 years. We developed a real-world autonomous vehicle (AV) using the CARLA open source simulator. The research and development is inspired by companies like Zoox, Witricity, Tesla Autopilot, Zscalar and Cruise for reference and inspiration. We have implemented a very realistic driving environment which involves 3D modelling of pedestrians, traffic and environmental conditions.

The simulator uses state estimation and location in an autonomous car using sensors, GNSS and IMUs. It implements Unscented Kalman Filter to locate a car using CARLA simulator data.

This will enable the main tasks of perception in autonomous driving are dynamic and static object detection, as well as examine common methods of computer vision for perception. Methods such as object detection, visual odometry, and tracking, and semantic segmentation are used in the estimation of the drivable surface.

The research implements key planning tasks in autonomous car, which includes mission planning, behavioral planning and local planning. It also implements map development and uses algorithms to execute and design, smooth and optimum trajectories and velocity profiles for safe navigation around obstacles while respecting traffic laws. Numerous hierarchical motion planners are used to navigate through a variety of scenarios in the CARLA simulator, which includes avoiding a vehicle parked in your lane, following a vehicle which is moving in front of it and safely navigating in an intersection.

I. INTRODUCTION

An autonomous car is a vehicle that implements a combination of cameras, sensors, radars, ultrasounds and Computer Vision (CV), to travel between start and destinations without the need for any human interference.

In the past five years, autonomous driving has gone from "perhaps possible" to "now commercially available on the market" and has become a reality.

Autonomous driving can open the door to future systems where computers take control of the art of driving.

In recent years, the automotive industry has made great strides towards a future free of human drivers. Researchers are currently working to overcome the technological, policy and social challenges posed by the spread of autonomous vehicles. These vehicles must be secure, reliable and economical. The use of machine learning technologies and the creation of coordination mechanisms could contribute to these objectives.

The global market for autonomous cars is segmented according to application of the car, vehicle type, product type, technological components (both hardware and software) and geography.

This paper provides a way to apply machine learning to the implementation of automated driving capabilities in a vehicle. It uses computer vision, that trains the onboard computer to understand and interact with the visual world. Using digital images and videos from cameras and deep learning models, machines can precisely classify and identify the objects and respond to what they sense.

This can be performed in a controlled environment using a simulator such as CARLA.

II. CARLA SIMULATOR

CARLA is one open-source simulator for autonomous vehicle driving research. CARLA is engineered from scratch to support development, validate and train autonomous driving systems. In addition to the code and open-source protocols, CARLA also provides open digital assets such as urban layouts, vehicles, buildings and pedestrians which have been created for simulation and can be freely used. The developed simulation platform supports flexibility in the specification of sensor suites and environmental conditions. Some of them are illustrated in Fig 1 given below.

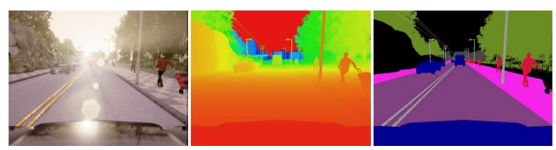

Fig 1: These are the views from camera which is converted to semantic view using computer vision to classify objects in the environment and identify safe drivable area for traveling.

CARLA is engineered for flexibility and realism in rendering physical simulation. It is implemented as an open-source layer on Unreal Engine 4 (UE4), permitting future expansions by the community. This Unreal Engine delivers state-of-the-art rendering quality, realistic physics, core NPC logic and an interoperable plug-in ecosystem. The engine is freely available for non-commercial use. CARLA simulates a dynamic world and offers a simple interface between the world and an agent which interacts with the world. In order to support this function, CARLA is designed as a client-server system, where the server is responsible for running the simulation and rendering the scene. The CARLA's client API is built using Python. It is mainly responsible for the interaction between the server and the autonomous agent through sockets. The client sends commands and meta-commands to the server and then receives sensor readings in return. Commands control the vehicle that include steering, braking, and acceleration.

ISBN: 979-85-27243-61-1

III. Proposed Methodology

The proposed system includes the major tasks of autonomous driving planning, consisting of local planning, behaviour planning and mission planning. The system is trained to find the shortest route through the road network using Unscented kalman filter, Dijikstra's and A* algorithm. It uses finite state machines to identify and plan safe behaviours to design optimal, secure and smooth trajectories and speed profiles for safe navigation in the environment while respecting traffic rules. It creates maps of occupancy grid map of static elements in the surrounding and learns how to handle them for effective crash control. The system will construct a fully autonomous driving planning solution, to take you from one place to another while behaving like a typical driver by maintaining the vehicle safety at all times. This includes implementing a hierarchical motion planner to navigate through series of real life simulated scenarios of CARLA simulator, like following a lead vehicle, avoiding a parked vehicle in the lane, safely navigate in an intersection, entering and leaving a roundabout, passing slower vehicles. It is trained for real-world randomness and is robust to environmental changes.

The basic architecture of the implemented autonomous system can be decomposed into six standard modules. They are

- Environment Perception
- Environment Mapping
- Motion Planning
- Sensor Fusion
- Controller
- System Supervisor

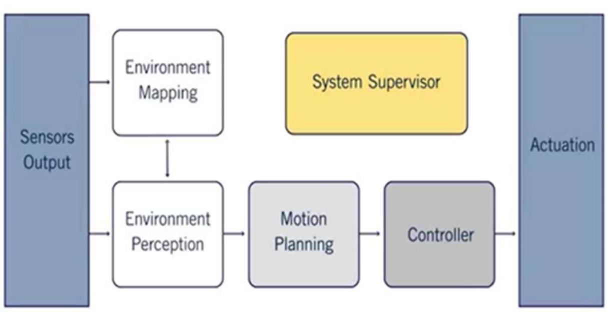

Fig 2: Basic architecture of the autonomous driving software system.

These modules explains about how the system receives the inputs, how computations are made based on these and what outputs they generate. Let us discuss these modules more in detail.

A. Environment Perception and Mapping

The raw sensor measurement is passed to two modules (environment mapping and perception) to understand the environment around the car.

Environment perception involves identification of vehicles in space and then classify the elements in the environment for the driving.

Environment mapping involves mapping the objects around the autonomous vehicle for wide range of uses for example avoiding collision.

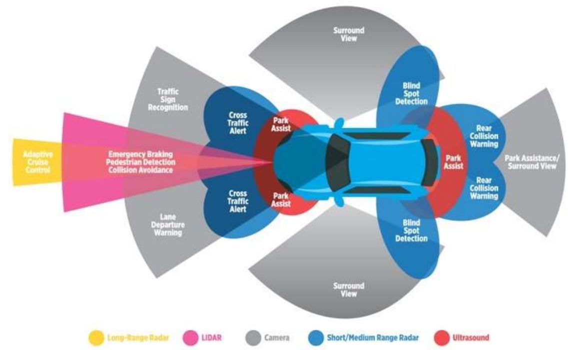

Fig 3: Sensor positioning in the autonomous vehicle and its functions.

These are the sensor data we have taken into consideration.

- Global Navigation Satellite Systems (GNSS) – GNSS is a catch-all term for a satellite system that can be used to pinpoint a receiver's position anywhere in the world. To compute a GPS position fix in the Earth-centred frame, the receiver of GNSS uses the speed of light to calculate distances to each satellite based on time of signal arrival. A GNSS works through trilateration via pseudo ranging from at least 4 satellites.
- Inertial measurement Unit - An IMU is typically composed of

  Gyroscopes – It is a spinning disc that maintains a specific orientation relative to inertial space, providing an orientation reference. They measure angular rotation rates about three separate axes.

  Accelerometers - They measure the acceleration relative to free-fall - this is also called the proper acceleration or specific force. They measure accelerations along three orthogonal axes.
- LIDAR- LIDAR emits laser light and measures distance using the time-of-flight equation. The device scans are stored as points that can be manipulated using common spatial operations like rotation, translation, scaling to identify position and movement of objects. The Iterative Closest Point (ICP) algorithm is used by LIDAR for localizing the autonomous car. As Fig 4 the device scatters laser beams in all the directions to identify boundaries. The module uses dynamic Lidar positioned overhead of car.

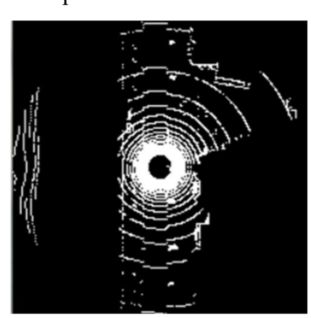

Fig 4: Lidar processed output

- RADAR – Radar has been used for decades to compute the velocity, range and angle of objects. On the road, radar is playing a vital role of driving, which constitute in the development of autonomous car. The Frequency-Modulated Continuous Wave (FMCW) radar system is used to detect the range and velocity of targets through stretch processing.
- Ultrasound Sensor - Ultrasonic sensors was initially introduced to vehicles as sensors of parking assistance systems. Ultrasonic sensors detect obstacles by transmitting and receiving ultrasound, a frequency is beyond the upper limit of human hearing at 20 kHz. To measure the distance from an object, the ultrasonic sensor transmits ultrasonic pulses, and measures the time taken for reception of reflected pulses. Distance between the nearest obstacle is calculated based on the propagation time of the received echoes.
- Camera - The cameras provide images that AI-based Deep-Learning programs can analyse with a high level of precision. The cameras use visual data received from the optics in the lens to the computer vision software for more-in-depth analysis. With the implementation of neural networks and computer vision algorithms, objects can be identified to provide the car's information as it drives. This allows the car to slow down when traffic is on, avoid collisions, make safe lane changes, and even read the traffic signs on the roads or highways.

  Cameras are used for computing visual depth perception which are done using stereo algorithms. The cameras used are set to generate images of ratio 720px X 1280px which is sent as input for computer vision programs for object identification and perception.

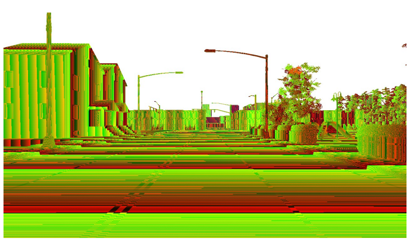

Fig 5.1: RGBD camera depth raw input.

The process of incremental estimation of the vehicle posture by examining the changes induced by motion on the images of its onboard cameras generating visual odometry for car.

Semantic segmentation of the image can be carried out with the aim of labelling each pixel of an image with a corresponding class of what is represented. As we make predictions for each pixel of the image, this task is often called dense prediction.

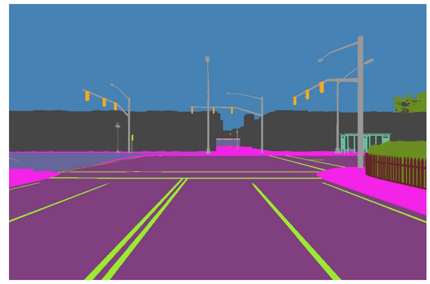

Fig 5.2: Semantic Segmentation of the environment.

B. Motion Planning

Motion planning for autonomous vehicles has been developed and greatly improved in the past years. Most fully autonomous vehicles have their own motion planning to directly control a desired trajectory that builds both the desired trajectory and velocity.

Motion planning can be broken into hierarchy of sub problems.

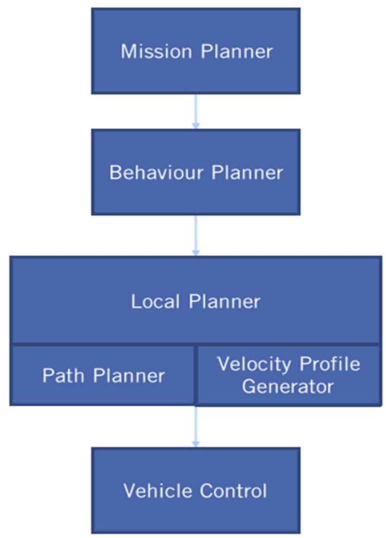

Fig 6: Hierarchy of motion planning sub problems.

- Mission planner is the highest level planner. It focuses on the mission which is to navigate to destination at the map level. It can be solved with graph based methods (Dijkstra's, A*).
- Behavioural planner decides when it is safe to proceed. It takes pedestrians, vehicles, cyclists into consideration. It also looks at regulatory elements such as traffic lights and stop signs.
- Local Planning generates feasible, collision-free paths and comfortable velocity profiles. This profile is then fed to the controller which makes the necessary adjustments accordingly.

C. Sensor Fusion

Sensor fusion is the process of blending sensor data with information from many sensors. Sensor fusing plays a vital role in planning. It collects data from all the sensors simultaneously and fuses it to generate a more precise

ISBN: 979-85-27243-61-1

perspective of the environment with the assistance of an occupancy grid map and simultaneous localization and mapping (SLAM).

The sensor fusion can be classified into three major steps as follows.

- Data analysis: Data from the sensors come at a delayed time and the frequency would be different which has to be processed and merged. The algorithm combines several heterogeneous sources of sensory data to attain more precise and synthetic readings.
- Feature Level: The features express information calculated on board by each sensor. These features are then sent to the fusion node to feed the fusion algorithms. This process generates smaller information spaces for the data level fusion, which is preferable in terms of computational load.
- Decision Level: The decision level is the procedure for selecting a hypothesis from a set of hypotheses generated by individual decisions of several sensors. It is the highest level of abstraction and uses information that was formerly been developed through preliminary data or at the level of the processing function. The main job of decision level fusion is to use a meta-level classifier while data from the nodes are preprocessed by extracting certain features to identify the environment with greater precision and overcome hardware faults like sensor damage of foggy weather where camera visibility is low.

D. Controller

This module tracks and determines the best steering angle, gear settings, throttle pedal position and brake pedal position to accurately follow the planned path. It consists of

- Model Reference Point – This involves building a dynamic model of a car using kinematic bicycle model as a starting point and converting it to a standard state space representation.
- Longitudinal Control – For Longitudinal control Feedforward and feedback controllers are used together. Feedforward controller provides predictive response, non-zero offset. Feedback controller corrects the response compensating for the disturbances and errors in the model.
  The output of the feedforward and the feedback control blocks the throttling and the braking signals to accelerate or decelerate the vehicle to keep the vehicle velocity close to the reference velocity.

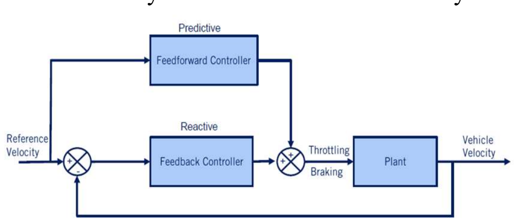

Fig 7.1: Structure of longitudinal control.

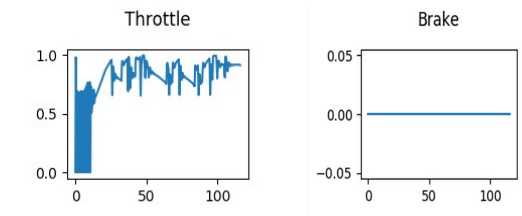

Fig 7.2: Graph indicating throttle and brake output.

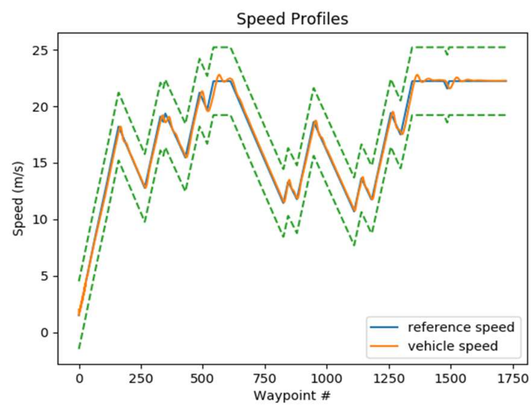

Fig 7.3: Graph indicating achieved vehicle speed against reference speeds.

- Lateral Control – The model predictive control (MPC) approach is enforced for controlling the active front steering system of the autonomous vehicle. At each time step, a trajectory is calculated over a finite horizon, and the MPC controller computes the updated steering angle in order to follow the trajectory.
  The conversion of the tire forces into steering commands is performed through the use of the MPC block. The resulting low-level control is executed by the vehicle's steering and throttle controls. The feedback loop is then closed using the applicable actuation signals. The MPC is a low-level controller that uses the tire forces as a low-level control unit inside the feedback loop. The vehicle's control system is then optimized to implement the appropriate steering and throttle commands.

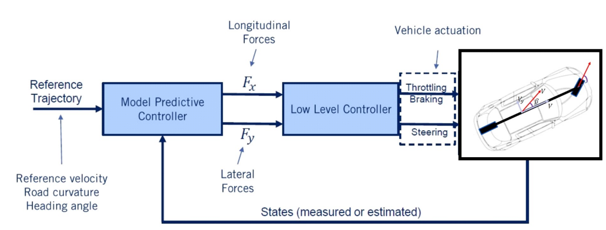

Fig 8.1: MPC structure for lateral control.

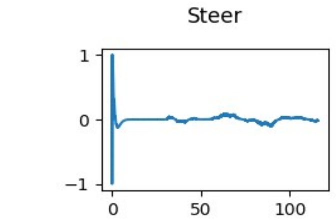

Fig 8.2: Graph indicating the steering angle set at the position of the car.

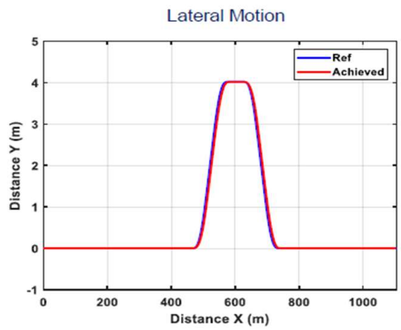

Fig 8.3: Graph indicating achieved lateral motion against reference lateral motion.

E. System Supervisor

The system supervisor monitors all parts of the software stack and hardware outputs to ensure that all systems are working as intended. It is also responsible to report any issues within the system.

IV. CONCLUSION

The Autonomous vehicle remains one of the hottest topics in the field of research. This paper introduces the basic principle of autonomous car based on algorithms.

This paper tells about the implementation details of development of maps, and use algorithms for safe navigation around obstacles while respecting traffic laws.

In practical applications, these simulations can be applied in real world which removes human interactions with the vehicles.

We can apply this in real world with minimal training of real world environment.

REFERENCES

[1] B.A. Francis and M. Maggiore, "Flocking and Rendezvous in Distributed Robotics",2016.

[2] Khaled Sailan and Klaus.Dieter Kuhnert, "Modeling And Design Of Cruise Control System With Feedforward For All Terrian Vehicles", 2013.

[3] Paolo Falcone, Francesco Borrelli, Jahan Asgari, Hongtei Eric Tseng, and Davor Hrovat, "Predictive Active Steering Control for Autonomous Vehicle Systems", 2007.

[4] Jarrod M. Snider, "Automatic Steering Methods for Autonomous Automobile Path Tracking", 2009.

[5] Eric A. Wan and Rudolph van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation", 2001.

[6] A. D. KING, "Inertial Navigation – Forty Years of Evolution", 1998

[7] Haoyang Ye and Ming Liu, "LiDAR and Inertial Fusion for Pose Estimation by Non-linear Optimization", 2017.

[8] Charles R. Qi1, Wei Liu, Chenxia Wu, Hao Su, Leonidas J. Guibas, "Frustum PointNets for 3D Object Detection from RGB-D Data", 2018.

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", 2016.

[10] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", 2016.

[11] Dr. Faruk Uysal, "Autonomous Systems and RADAR", 2019

[12] Jelena Kocić, Nenad Jovičić, and Vujo Drndarević, "Sensors and Sensor Fusion in Autonomous Vehicles", 2018.

[13] Wenyuan Xu, Chen Yan, Weibin Jia, Xiaoyu Ji, and Jianhao Liu, "Analyzing and Enhancing the Security of Ultrasonic Sensors for Autonomous Vehicles", 2018.

[14] Junqing Wei, Jarrod M. Snider, Tianyu Gu, John M. Dolan and Bakhtiar Litkouhi, "A Behavioral Planning Framework for Autonomous Driving", 2014 .

[15] Patel, Kaival Kamleshkumar, "A Simulation Environment with Reduced Reality Gap for Testing Autonomous Vehicles" (2020). Electronic Theses and Dissertations. 8305.