RV College of Engineering®

Go, Change the World

Academic Year 2024-25 (ODD Semester)

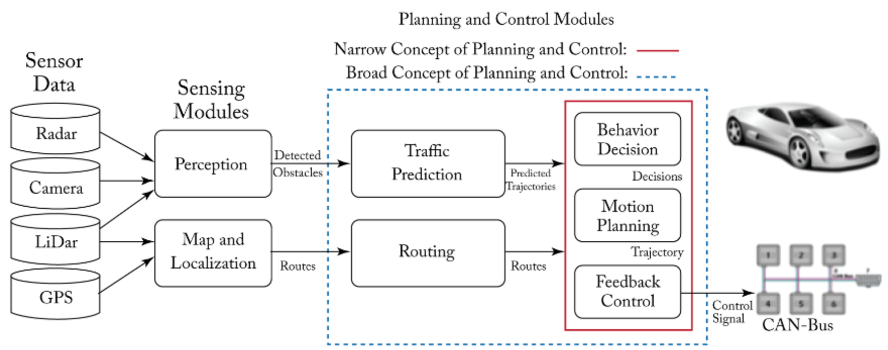
USN | 1 | R | V | | | | | | | |

Department of Artificial Intelligence and Machine Learning

ARTIFICIAL INTELLIGENCE IN AUTONOMOUS VEHICLES (CIE II)

Course Code	**:**	**21AI73G1**	**Date**	**:**	**28/01/2025**
Semester	**:**	**VII**	**Time**	**:**	**09:30:11:30**
Max Marks	**:**	**50**	**Duration**	**:**	**90 Mins**

S. No	Part – A Quiz	M	BT	CO
1	What is semantic segmentation in the context of autonomous vehicles?	2	L1	1
2	Differentiate between global planning and local planning in autonomous vehicles.	2	L2	3
3	Mention any two challenges in ensuring real-time performance in planning and control?	2	L2	1
4	Justify with reasons, why lane-level routing is important for autonomous vehicles.	2	L4	2
5	What is the role of data augmentation in training CNNs for obstacle detection?	2	L2	4

S. No	Part – B CIE	M	BT	CO
1	A perception system for an autonomous vehicle uses stereo flow, optical flow, and scene flow to navigate dynamic urban environments. During testing, the car struggles with accurately tracking fast-moving pedestrians crossing the road and fails to detect depth in poorly textured regions such as plain walls. Analyze how these techniques (stereo flow, optical flow, and scene flow) contribute to the observed performance issues. Suggest a combination of techniques or improvements to address these limitations while ensuring real-time processing and system reliability.	10	L4	2
2	Compare and contrast behavioral decisions with motion planning in autonomous vehicles.	10	L3	2
3	Write the pseudocode for implementing Dijkstra's algorithm to determine the shortest path in a weighted directed graph, where the graph is represented using lane points.	10	L2	2
4	 Fig. 5a. Using the provided diagram, evaluate the effectiveness of the planning and control modules in autonomous driving systems.	10	L4	3

	Consider the integration between the sensing modules and the planning components, particularly in the following scenarios: a. A sudden pedestrian crossing the road. b. Dynamic traffic flow changes requiring route recalibration. c. Propose modifications or alternative approaches to enhance the decision-making process and improve overall system reliability in handling such real-time events. Support your evaluation with relevant justifications based on system architecture.			
5	Distinguish between the Vehicle Bicycle Model and the PID Feed-back Control Systems with neat sketches and appropriate equations.	10	L2	3

M-Marks, BT-Blooms Taxonomy Levels, CO-Course Outcomes

Marks Distribution	Particulars	CO1	CO2	CO3	CO4	L1	L2	L3	L4	L5	L6
	Max Marks	04	32	22	2	2	26	10	22	-	-

Course Outcomes: After completing the course, the students will be able to	
CO1:	Analyze the various driving conditions for autonomous cars and apply AI techniques.
CO2:	Identify various problems involved in developing Autonomous Driving cars and suggest the appropriate Solutions.
CO3:	Integration of advanced driver assistance system with cloud infrastructure for training and modeling.
CO4:	Development of Deep learning techniques to analyze the data for decision-making.
CO5:	Demonstrate the use of modern tools by exhibiting teamwork and effective communication skills.

Scheme and Solutions

S.No	Quiz	Marks
1	Semantic segmentation is the process of classifying each pixel in an image into a predefined category, such as road, vehicle, pedestrian, or building, to understand the scene at a pixel level.	2
2	Global planning focuses on determining the overall route from the starting point to the destination, often using maps and GPS data. Local planning deals with short-term decisions and maneuvers to navigate immediate obstacles and dynamic elements in the environment.	2
3	Challenges include processing large amounts of sensor data quickly, making rapid decisions, and executing precise control actions, all while maintaining safety and reliability.	2
4	Lane-level routing ensures precise navigation, improves safety by adhering to traffic rules, and enhances the vehicle's ability to handle complex driving scenarios such as highway merges and urban intersections.	2
5	Data augmentation involves applying transformations such as rotation, scaling, and flipping to the training data, increasing the diversity of the dataset and helping the model generalize better to new, unseen data.	2

<table>
<tr><th>S.No</th><th>Question</th><th>Marks</th></tr>
<tr><td>1</td><td>
Analysis of Techniques and Performance Issues:

Stereo Flow:

Strength: Stereo flow calculates depth information by comparing disparities between two synchronized camera views. It works well for textured environments where reliable feature matching is possible.
Limitation: In poorly textured regions, such as plain walls, stereo flow struggles because there are insufficient unique features to match across the stereo images. This leads to inaccurate or missing depth information.
Issue in Context: The failure to detect depth in plain walls is directly linked to stereo flow's reliance on texture for disparity computation.

Optical Flow:

Strength: Optical flow estimates pixel motion between consecutive frames. It is effective for detecting motion and tracking objects, particularly in textured regions.
Limitation: Fast-moving objects, like pedestrians, create large displacements between frames, making it challenging for optical flow algorithms to compute accurate motion due to motion blur or aliasing effects.
Issue in Context: The car struggles to track fast-moving pedestrians due to optical flow's difficulty with high-speed motion and lack of temporal resolution.

Scene Flow:

Strength: Scene flow combines stereo flow and optical flow to provide a 3D motion field, estimating both depth and motion. It is useful for dynamic environments but inherits the limitations of both stereo and optical flow.
Limitation: Scene flow can be computationally expensive and suffers from degraded performance in poorly textured regions (from stereo flow issues) and with fast motion (from optical flow issues).
Issue in Context: The combined limitations of stereo flow and optical flow exacerbate the performance issues in dynamic, urban environments.

Suggested Combination of Techniques:

Primary Vision System: Utilize stereo cameras enhanced with deep learning-based stereo matching algorithms to improve depth perception in poorly textured areas.
Sensor Fusion: Integrate Lidar and radar data to handle the limitations of stereo vision and optical flow, ensuring reliable depth and motion detection in all scenarios.
Scene Flow Improvements: Use lightweight, real-time scene flow models with advanced neural networks for improved 3D motion estimation.
Multi-Modal Prediction: Employ temporal models to smooth object trajectories and compensate for high-speed motion challenges.

</td><td>05

10</td></tr>
</table>

2

Aspect	Behavioral Decisions	Motion Planning
Purpose	High-level decision-making	Low-level trajectory generation
Scope	Strategic and tactical	Operational
Time Horizon	Medium to long-term	Short-term
Inputs	Perception, maps, goals	Perception, behavioral output, dynamics
Computation	Less intensive	Computationally intensive
Focus	Contextual understanding and intent	Precise navigation
Failure Consequences	High-level misjudgments	Unsafe or infeasible trajectories

10

3

```
function Dijkstra_Routing(LanePointGraph(V,E), src, dst)
  create vertex set Q
  create map dist, prev
  for each lane point v in V:
     dist[v] = inf
     prev[v] = nullptr
     add v to Q
  dist[src] = 0
  while Q is not empty:
       u = vertex in Q s.t. dist[u] is the minimum
       remove u from Q
       for each connected lane point v of u:
           candidate = dist[u] + cost(u, v)
           if candidate < dist[v]:
              dist[v] = candidate
              prev[v] = u;
    ret = empty sequence
    u = dst
    while prev[u] != nullptr:
         insert u at the beginning of ret
         u = prev[u]
    insert u at the beginning of ret
    merge lane point in ret with same lane id and return the merged sequence
```

Dijkstra implementation of routing based on weighted directed graph of lane points.

10

4

Scenario a: A Sudden Pedestrian Crossing the Road
Evaluation:

- **Sensing Module**:
  - The sensing module must rapidly detect the pedestrian's presence using sensors like cameras, Lidar, or radar.
  - In a sudden crossing, latency in detection or failure to track fast movement could delay the decision-making process.
- **Planning Module**:
  - **Behavioral Planning**: Should prioritize safety by triggering an emergency braking or evasive maneuver.
  - **Motion Planning**: Needs to compute a collision-free trajectory, factoring in dynamic constraints like the car's speed and the pedestrian's movement.
  - **Integration Issues**: If the sensing module provides noisy or delayed data, the planning module may misjudge the pedestrian's position or trajectory, leading to delayed or incorrect actions.
- **Control Module**:
  - Effectiveness depends on how quickly the control module executes the emergency braking or steering maneuver generated by the motion planner.
  - Poor integration between planning and control (e.g., mismatched timing or lack of smooth trajectory following) can result in unsafe or uncomfortable braking.

Weaknesses Identified:

- Limited response time due to delays in sensing or planning.
- Potential over-reliance on single-sensor modalities (e.g., camera-only pedestrian detection).

10

Scenario b: Dynamic Traffic Flow Changes Requiring Route Recalibration
Evaluation:

- **Sensing Module**:
  - Continuous monitoring of traffic flow changes (e.g., road closures, congestion, or lane blockages) is critical.
  - Failure to detect updated road or traffic information in real-time can lead to poor route decisions.

- **Planning Module**:
    - **Behavioral Planning**: Should handle high-level route recalibration based on input from global navigation systems or traffic updates.
    - **Motion Planning**: Needs to execute lane changes or detours safely without disrupting local traffic flow.
    - **Integration Issues**: Insufficient communication between behavioral and motion planning layers could result in suboptimal recalibration, such as abrupt or unsafe lane changes.
- **Control Module**:
    - Needs to execute recalibrated trajectories smoothly to maintain passenger comfort and vehicle stability.

Weaknesses Identified:

- Behavioral planning may lack adaptability to sudden changes in traffic flow.
- Route recalibration might not adequately consider the immediate traffic context, leading to conflicts with other vehicles.

Proposed Modifications and Enhancements

1. **Enhanced Sensing Integration**:
    - Use **sensor fusion** to combine inputs from cameras, Lidar, radar, and V2X (vehicle-to-everything) communication for robust environment perception.
        - Justification: This reduces single-point failures and improves pedestrian detection in complex or occluded scenarios.
    - Implement **temporal consistency models** to track objects over time, ensuring accurate prediction of fast-moving pedestrians.
2. **Improved Planning Hierarchies**:
    - **Predictive Models**:
        - Incorporate intent prediction for pedestrians and vehicles using machine learning models, enabling proactive decisions.
        - Justification: Early prediction of pedestrian crossings or lane changes improves reaction time.
    - **Dynamic Route Recalibration**:
        - Use real-time traffic updates from external sources (e.g., V2X) and integrate global and local planners seamlessly to optimize route choices.
        - Justification: This ensures the planning module can handle both high-level changes (e.g., detours) and immediate constraints (e.g., local traffic).
    - **Hierarchical Planning Integration**:
        - Tight integration between behavioral and motion planning modules with shared optimization frameworks (e.g., Model Predictive Control or hybrid planning systems).
        - Justification: Reduces delays and ensures consistent decision-making across high-level and low-level planning.
3. **Robust Control Algorithms**:
    - Implement **adaptive control algorithms** that account for real-time updates from the motion planner and adjust vehicle dynamics (e.g., acceleration or steering) smoothly.
    - Use **fail-safe mechanisms**, such as emergency stop overrides, for critical safety scenarios.
        - Justification: Ensures the control module can handle abrupt changes without compromising safety.
4. **Real-Time Prioritization Mechanisms**:
    - Incorporate a **critical event detection module** that prioritizes urgent scenarios (e.g., sudden pedestrian crossings) over routine behaviors.
    - Justification: Focuses computational resources on high-priority tasks, ensuring timely responses.
5. **Simulation-Based Testing and Validation**:
    - Use high-fidelity simulators to test the entire system under edge-case scenarios, such as unpredictable pedestrian behavior or sudden traffic changes.
    - Justification: Identifies and resolves weaknesses in sensing, planning, and control integration before real-world deployment.

5

Bicycle Model:

10

The steering wheel rotation angle δ is defined as the angle between the front wheel direction and the vehicle body, where the ground contact points of front and rear wheels (p_f and p_r) satisfy the following properties:

$$(\dot{p}_r \cdot \hat{e}_y) \cos(\theta) - (\dot{p}_r \cdot \hat{e}_x) = 0$$
$$(\dot{p}_f \cdot \hat{e}_y) \cos(\theta + \delta) - (\dot{p}_f \cdot \hat{e}_x) \sin(\theta + \delta) = 0,$$

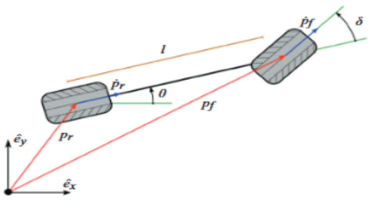
where $\dot{p}_f$ and $\dot{p}_r$ are the instant speed vector of the front and rear wheel at their ground contact point. Consider the scalar projections of the rear wheel speed at the x-axis and y-axis: $x_r := p_r \cdot \hat{e}_x$ and $x_y := p_r \cdot \hat{e}_y$, along with the tangential speed at the rear wheel $v_r := \dot{p}_r \cdot (p_f - p_r)/\| p_f - p_r \|$, then the above constraints between p_f and p_r can also be written as:

$$\dot{x}_r = v_r \cos(\theta)$$
$$\dot{y}_r = v_r \sin(\theta)$$
$$\theta = v_r \tan(\delta)/l,$$

where l represents the length of the vehicle (distance between the front axis center and the rear axis center). Smilarly, the relationship between the front wheel variables can be written as:

$$\dot{x}_f = v_r \cos(\theta + \delta)$$
$$\dot{y}_f = v_r \sin(\theta + \delta)$$
$$\theta = v_f \sin(\delta)/l.$$

Note that the scalar variables of front and rear wheel speeds satisfy: $v_r = v_f \cos(\delta)$.



PID CONTROL

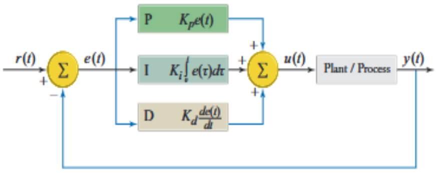


Figure 6.10: PID-based feedback-control system (based on [13]).

As for the feedback control module in autonomous vehicle, the task is to control the vehicle to follow the upstream motion planning output trajectory as closely as possible. We thus propose using the methodology in [15], and leveraging two PIC controllers to separately control the steering-wheel angle δ and the forward speed V_s. At a given time frame n, the PID controller for the steering-wheel angle is as follows:

$$\delta_n = K_1\theta_e + K_2 l_e / V_s + K_3 \dot{l}_e + K_4 \sum_{i=1}^{n} l_e \Delta t .$$

The variables θ_e and l_e are all tracking error terms between the actual pose and the desire pose on the motion-planning output trajectory point at this time frame n. For each time frame at this time frame n, the corresponding pose on the motion planning output trajectory point is addressed as the *reference point*. θ_e represents the angle difference between the vehicle pose heading and the reference point heading, while l_e tracks lateral difference between the vehicle actual lateral position and the reference point lateral position. V_s is the forward speed. The coefficient K_1 and K_2 are meant for the P controller, while K_3 governs the differential part (D controller) and K_4 the integral part (I controller). Given that this steering-angle controller serves for direction, the other PID controller is more about the forward speed V_s along the longitudinal direction (s-direction), and controls throttle/brake output. This controller considers the difference between the actual vehicle pose curvature and the reference point curvature. From these curvatures, we can design a function to track the forward speed error. Given this desired forward speed to track and the actual forward speed xxx, the PID controller for the forward speed can be written as:

$$V_e = V_{desired} - V_s$$
$$U_V = K_p V_e + K_I \sum V_e \Delta t + K_D \Delta V_e / \Delta t,$$

where K_p, K_I, and K_D separately represent the gain for the proportional, integral, and differential part, and U_V represents throttle/brake output for this given time frame n.

These two PID controllers discussed here are the most typical and basic implementation practices for the feedback *control* module in autonomous vehicles. To make an even better passenger experience in autonomous driving, more complicated feedback control systems will be necessary to further track and tune variables such as curvature and jerk. The problem of generating delicate and accurate control to enforce an object to follow a pre-defined trajectory is not a unique problem for autonomous driving, there are many existing solutions [15].