1

Advancing Edge AI Perception Platforms and Sensor Fusion for Last-Mile Delivery Autonomous Vehicles

Ovidiu Vermesan[1], Roy Bahr[1], Hans-Erik Sand[2], Simen Marentius Saxegaard[2], Helge Brudeli[3], Petter Emanuelsson[3], and Martin Førrisdahl[3]

[1]SINTEF AS, Norway
[2]NxTECH AS, Norway
[3]Paxster AS, Norway

Abstract

In the rapidly evolving landscape of transportation, mobility, and logistics, the last-mile represents the final and crucial leg of the delivery journey. It involves goods travelling from a transportation hub to the ultimate destination. This is the most expensive and time-sensitive part of the supply chain business model. Challenges include navigating dense urban environments with various types of traffic participants (e.g., pedestrians, bicycles, animals, electric scooters, motorcycles, etc.), dealing with traffic congestion, locating specific delivery points, managing a high density of stops, and handling failed delivery attempts. In this context, the intersection of edge artificial intelligence (AI), autonomous systems, robotics, and sensor fusion in perception and navigation advances the development of last-mile delivery autonomous vehicle (AV) platforms that evolve towards software-defined and AI-defined vehicles (SDVs and ADVs). The advancements include multiple sensor systems for perception and communication (e.g., ultrasound, inertial, LiDAR, radar, camera, V2X, etc.), real-time data processing for localisation, and robust algorithms for navigation and interaction with diverse traffic environments. This chapter presents the concept and the implementation of an AI-based

perception and sensor fusion platform technical solution for autonomous last-mile delivery in controlled traffic environments.

Keywords: edge AI, perception, autonomous vehicle, sensor fusion, object recognition, last-mile delivery.

1.1 Introduction and Background

The future of mobility is intelligent, electrical, autonomous, connected, and shared, affecting all three broad types of mobility: personal mobility (moving individuals or small groups of people), mass transit (moving large numbers of people), and the movement of goods.

The last-mile of logistics refers to the final leg of the delivery journey, typically from a local distribution hub or retail centre to the end recipient's location, such as a home or business [1]. This segment is notoriously the most complex, inefficient, and expensive part of the entire supply chain, often accounting for over 50% of total delivery costs [3].

Last-mile logistics are increasingly automated, and companies that are prepared for this shift are in a stronger position to compete and take the lead. Autonomous last-mile delivery, utilising vehicles ranging from small sidewalk robots to automated vans, promises significant efficiency gains and cost reductions in logistics. As a result, various types of autonomous vehicles for last-mile delivery have emerged as follows [10] and illustrated in Figure 1.1:

- Pedestrian sidewalk vehicles. These are slow vehicles designed to travel at a pedestrian speed of 4-6 km per hour. This low speed offers improved safety and allows the operators to control the vehicle in an emergency.
- Bicycle sidewalk vehicles. These are vehicles designed to travel up to a bicycle speed of 12-15 km per hour.

Vehicle type	**Pedestrian sidewalk vehicles**	**Bicycle sidewalk vehicles**	**On-road delivery vehicles**
Speed (up to)	**4-6 km/h**	**12-15 km/h**	**45-50 km/h**
Traffic zone	**Pedestrian zones**	**Bike paths**	**Road**
Traffic participants	**Pedestrians, cyclists, animals**	**Cyclists**	**Vehicles, cyclists**

Figure 1.1 Types of vehicles for last-mile delivery.

- On-road delivery vehicles. These vehicles are built for on-road delivery at up to 45-50 km per hour. Their software algorithms and sensor systems resemble those of autonomous vehicles.

Driverless technology users utilise autonomous deliveries for several purposes:

- Delivery of goods from warehouses to stores and outlets for restocking inventory and shelves.
- Delivery of goods from stores to end consumers.
- Delivery of goods and parts between the warehouses and production facilities.

Autonomous vehicles (AVs), encompassing road-going vans and smaller sidewalk autonomous delivery vehicles, are emerging as a potentially transformative solution. By eliminating the need for a human driver, AVs offer the potential for 24/7 operation, reduced labour costs (a significant component of last-mile expense), optimised routing, and potentially lower emissions, primarily if electric. They can navigate narrow streets or pedestrian zones inaccessible to larger vehicles and improve delivery times by avoiding human-related delays.

The development of last-mile delivery autonomous vehicle fleets is linked to the evolution of Internet of Robotic Things (IoRT) platforms. IoRT serves as the technological backbone, integrating individual autonomous robotic vehicles into an interconnected system of systems. IoRT combines IoT technologies with robotics, edge computing and AI, allowing for the coordination of large-scale fleets of delivery robots that might otherwise operate alone. Connecting and integrating the last-mile delivery autonomous vehicle into fleets that are coordinated using distributed networks or IoRT platforms enables functions such as remote monitoring, intelligent communication, and the management of the entire delivery process, which can be managed from the distribution centre to the customer's doorstep. The intelligence and real-time responsiveness of the autonomous delivery fleets and the IoRT platforms are significantly enhanced by edge AI. Edge AI embeds data processing and decision-making capabilities directly onto the vehicles themselves, strengthening the processing capabilities and the analytics of each vehicle in the fleet or the IoRT platform. The use of edge AI enables continuous route optimisation, obstacles and traffic participants avoidance, and real-time adaptation to changing environmental conditions, which are critical for navigating pedestrian spaces and complex urban environments safely and efficiently. Edge AI-powered robotics within the IoRT framework calculate the most efficient paths in real-time, ensuring that last-mile delivery is not only automated but also intelligent, fast, and reliable [5, 6].

For autonomous last-mile delivery to become a reality, the core enabling technology is robust perception – the AV's ability to sense, interpret, and understand its complex and dynamic surroundings. Last-mile environments, whether sidewalks or urban streets, present unique perception challenges: close-quarters manoeuvring around pedestrians, cyclists, pets, parked cars, street furniture, and unpredictable obstacles; navigating varied terrain including curbs and uneven surfaces; interpreting complex traffic signals and signs at intersections; precisely identifying the final delivery location (e.g., a specific doorway or porch); managing a high density of stops, and handling failed delivery attempts [2]. Failures in perception can lead directly to collisions, incorrect deliveries, or mission failure.

No single sensor can reliably capture all necessary environmental information under all conditions. Cameras struggle in poor lighting or weather, LiDAR can be expensive and has limitations in adverse weather, radar has lower resolution, and ultrasound has a very short range. Therefore, sensor fusion – the intelligent combination of data from multiple, diverse sensors is key [14]. By integrating complementary data streams, sensor fusion aims to create a unified, comprehensive, and reliable environmental model that is more accurate and robust than what could be achieved with individual sensors alone.

Last-mile delivery autonomous vehicles can operate in fleets with individual vehicles acting as cognitive agents using perception modules to process images, GNSS positions or LiDAR scans for autonomous system decision-making, resulting in actions, such as actuator commands or V2X messages. The high degree of interdependencies between many functional components of autonomous vehicles requires the implementation of new system architectures and new underlying software frameworks. The concepts of developing AI-based last-mile autonomous delivery vehicles are embedding Robot Operating System (ROS) into compact, scalable, AI-based perception, localisation and sensor fusion platforms advancing the solutions and applications for autonomous transport of goods.

An essential aspect of the safe use of last-mile delivery autonomous vehicle technology is determining its capabilities and limitations and communicating these to end users, leading to a state of "informed safety". The first stage in establishing the capability of an autonomous vehicle is defining its Operational Design Domain (ODD). The ODD is defined in [17] as the operating conditions under which a given driving automation system or feature thereof is specifically designed to function and can perform the dynamic driving task (DDT) safely. This includes, but is not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of specific traffic or roadway characteristics. DDT consists of both a tactical driving task and an operational driving task, encompassing all the real-time operational and tactical functions necessary to operate a vehicle in on-road traffic, including lateral vehicle motion control via steering (operational); longitudinal vehicle motion control via acceleration and deceleration (operational); monitoring the driving environment via object and event detection, recognition, classification and response preparation (operational and tactical); object and event response execution (operational and tactical); manoeuvre planning (tactical); and enhancing conspicuity via lighting, sounding the horn, signalling, gesturing (tactical). This excludes the strategic functions, such as trip scheduling and the selection of destinations and waypoints [17].

The ODD defines the functional boundary of the system, and the autonomous system's functional architecture implements the system requirements, considering the ODD, technological constraints, and how high-reliability and safety systems can be designed, built, and tested using realistic sensors/actuators, hardware, software, and AI components.

Autonomous system functional architecture used to implement the different autonomous functions refers to the logical decomposition of the system into sub-functions/sub-components and the data flows between them [7] as illustrated in and listed below:

- **Sense**: Process a variety of sensing modalities.
- **Map**: Provide static and dynamic map data.
- **Localise**: Calculate the vehicle's position, orientation, and motion.
- **Perceive**: Calculate drivable areas and obstacle location and motion.
- **Predict**: Estimate the future motion and movement of dynamic objects.
- **Plan**: Calculate a desired trajectory for a vehicle.
- **Control**: Execute the trajectory as steering, brake/acceleration, and throttle commands.
- **Learn**: Enhance the capabilities through learning from the use cases, scenarios and missions performed.

The integration of the system functions, sub-functions/sub-components into a functional architecture for autonomous vehicles is presented in the Figure.

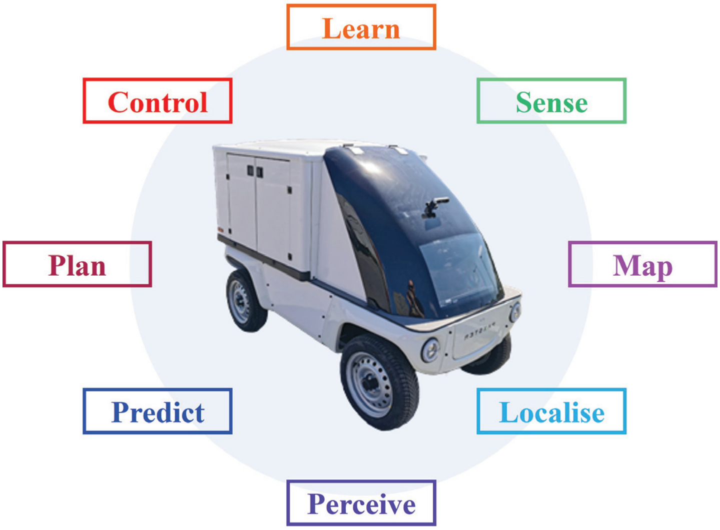

Figure 1.2 Autonomous vehicle functional elements.

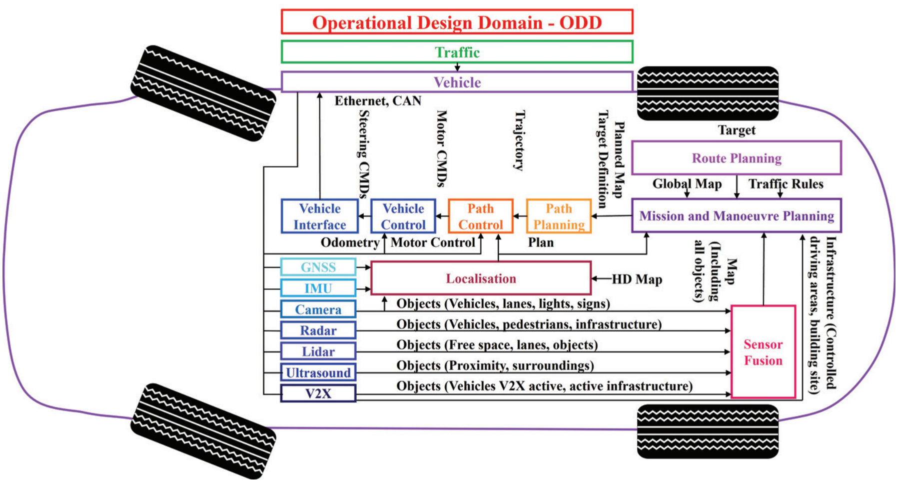

Figure 1.3 Autonomous vehicle functional architecture overview. Source: Adapted from [7].

The scalable functional architecture implements different autonomous functions using AI-based platforms that can navigate autonomously, detecting dynamic obstacles and following an optimal trajectory. The vehicles can recognise actions by processing information acquired from perception sensors and sensor fusion (e.g., GNSS, IMU, camera, radar, LiDAR, V2X). The processes are integrated into the ROS architecture, which shows potential for last-mile delivery autonomous vehicles applications.

The development of last-mile autonomous delivery vehicles is following the latest advances in SDVs to manage and integrate multiple software stacks and hardware components from various suppliers. The convergence of SDV, generative AI, and the Internet of Things (IoT) can pave the way for AI-defined vehicles as the future of mobility [32]. ADVs are the next step in the development of SDVs, focusing on hardware and infrastructure advancements that enable software decoupling from hardware with a hybrid stack that integrates control, processing and AI functionalities seamlessly [32, 34].

1.2 Sensor Fusion in Last-Mile Context

Sensor fusion is the process of intelligently combining data from multiple heterogeneous sensors [14] to generate a more accurate, complete, reliable, and robust representation of the environment than any single sensor operating alone. It leverages the complementary strengths of different sensor modalities while mitigating their weaknesses.

In the specific context of last-mile delivery, sensor fusion aims to build a detailed and dynamic understanding of the immediate surroundings, which is crucial for navigating complex, often cluttered, and highly interactive environments, such as sidewalks, crosswalks, residential streets, and building entrances [15]. Key objectives include:

- Accurate localisation and mapping, especially in GNSS-challenged urban canyons [16].
- Robust detection and tracking of static and dynamic obstacles, including pedestrians, cyclists, pets, and other small, unpredictable actors common in urban/suburban settings [18].
- Reliable perception across diverse and challenging conditions, such as varying lighting (day/night, shadows, glare, etc.), adverse weather (rain, fog, snow, etc.), and sensor occlusions [21].
- Precise identification of navigable paths, drivable surfaces, curbs, and specific delivery locations (e.g., doorsteps, designated drop-off zones) [18].
- Generating a unified environmental model suitable for real-time planning and control of potentially low-speed, highly manoeuvrable delivery vehicles [22].

A typical sensor suite for a last-mile delivery autonomous vehicle aims to provide 360-degree awareness and redundancy by combining sensors with different operating principles and characteristics.

Common sensors components for last-mile delivery autonomous vehicles are presented in the Table 1.1 and described in the following paragraphs.

Cameras are the primary source of rich semantic information for autonomous vehicles [18]. They capture visual details like colour, texture, and shape, enabling the recognition and classification of objects (pedestrians, vehicles, traffic signs, traffic lights, lane markings), interpretation of road signs, and potentially reading delivery labels or house numbers [3]. Cameras provide high-resolution data, are relatively low-cost compared to LiDAR, are passive sensors consuming less power, and are adept at capturing the complex visual details necessary for semantic understanding in diverse urban and pedestrian environments [18]. Their cost-effectiveness is a significant advantage for deploying delivery robots or vans at scale. Multiple cameras can provide a 360-degree field of view [18]. Camera performance degrades significantly in adverse weather conditions (e.g., heavy rain, fog, snow) and challenging lighting (e.g., low light, nighttime, direct glare, shadows) [18]. Monocular cameras struggle with direct and accurate depth estimation, requiring computationally intensive techniques like stereo vision or structure-from-motion or fusion with other sensors like LiDAR or radar. They are susceptible to occlusions, where objects of interest are hidden behind others. Fast motion or vehicle vibration can cause motion blur, particularly with rolling-shutter cameras, potentially impacting the accuracy of perception [18]. Cameras are indispensable for navigating visually complex sidewalks and streets, identifying pedestrians and other vulnerable road users (VRUs) for safe interaction, reading traffic signals and signs crucial for road crossings, and potentially identifying specific delivery addresses or drop-off points [18]. However, the limitations in poor weather and lighting pose significant challenges for achieving reliable 24/7 operation required by many delivery services. While global shutter cameras can mitigate motion blur in dynamic close-quarters environments, they also add to the cost [18]. The need for robust performance across diverse environmental conditions necessitates fusing camera data with other sensor modalities.

LiDAR is the primary sensor for generating accurate, high-resolution 3D maps of the environment and providing precise distance measurements of objects [3]. It creates a point cloud representing the geometry of the surroundings. LiDAR offers excellent depth accuracy and creates detailed 3D

Table 1.1 Sensor Modality Comparison for Last-Mile Delivery AVs

Feature	Camera	LiDAR	Radar	IMU	Ultrasound	V2X
Role	Semantic understanding, object classification, traffic sign, light recognition.	Accurate 3D mapping, depth perception, obstacle geometry.	Detect objects and measure their distance, velocity, and angle in relation to the vehicle.	Motion/ orientation tracking, dead reckoning during GNSS loss.	Very short-range object detection, parking, docking assist.	Extended situational awareness, cooperative perception, non-line-of-sight detection.
Strengths (Last-Mile)	Rich detail for VRU and sign recognition, low cost, low power.	Precise localisation in urban canyons, detects low obstacles, curbs, day, night operation.	Robust performance at distance. Not significantly affected by rain, fog, snow, low-light, night conditions.	Operates without external signals (GPS-denied), high frequency data for fusion/ stabilization.	Very low cost, detects objects extremely close (blind spots), good for tight manoeuvring.	Detects beyond line-of-sight, integrated with infrastructure (e.g. traffic lights).
Limitations (Last-Mile)	Poor performance in bad weather, lighting, poor depth estimation, motion blur.	High cost (major barrier), degraded by heavy precipitation, dust, no colour, texture info.	Lower resolution image of the environment, poor to detect and classify stationary objects, inference from other radars.	Accumulates drift error quickly, needs constant correction, sensitive to vibration/temp.	Very limited range (<5-10m), poor angular resolution, poor performance at speed.	Requires widespread adoption, infrastructure, latency, reliability issues, security, privacy concerns.

Table 1.1 *Continued.*

Feature	Camera	LiDAR	Radar	IMU	Ultrasound	V2X
Cost Factor	Low	High and decreasing.	Medium and lower than that of LiDAR systems.	Low (MEMS) to medium.	Very low.	Medium (requires comms module + potential network, infrastructure costs).
Importance for Last-Mile	Essential (semantics, VRUs, signs).	Highly important (localisation, 3D structure, obstacle detection)	Still expensive. Reliable object detection / tracking in poor weather / lighting conditions.	Essential (localisation continuity, state estimation)	Complementary (near-field safety, docking)	Potentially high (safety, efficiency), but dependent on ecosystem maturity.

representations, which are crucial for localisation, mapping, and obstacle detection [21]. It operates effectively regardless of ambient lighting conditions (day or night) [21]. Its performance is generally more robust in certain adverse weather conditions (like light rain or fog) compared to cameras, although heavy precipitation can still cause significant degradation [21]. Recent advancements have led to more compact and potentially lower-cost solid-state LiDAR units. LiDAR sensors remain relatively expensive, particularly high-resolution, long-range units, posing a significant cost barrier for mass deployment, especially on more miniature, cost-sensitive delivery robots [4]. Performance can be degraded by heavy rain, snow, dust, or fog [21]. LiDAR cannot perceive colour or texture information, making it challenging to classify objects based solely on LiDAR data (e.g., reading traffic signs or distinguishing between visually similar objects) [23]. The point clouds generated can be sparse, especially for distant or small objects. Mutual interference between multiple LiDAR sensors operating in the same area is a potential issue. Traditional mechanical scanning LiDARs have moving parts, raising concerns about long-term reliability and durability, particularly on vehicles operating over bumpy terrain, such as sidewalks. LiDAR is crucial for precise localisation and mapping within complex urban canyons or sidewalk environments where GPS may be unreliable [21]. It excels at detecting low-lying obstacles, curbs, potholes, or changes in terrain that might be missed by cameras alone. The high cost remains a significant challenge for the last-mile business case [4]. The typically lower speeds and shorter operational ranges in last-mile delivery may allow for the use of lower-cost, shorter-range LiDAR sensors compared to those needed for high-speed highway autonomy [14]. Fusion with cameras is essential to add semantic understanding to LiDAR's geometric data.

Radar sensors [19] are one of the key elements for the autonomous vehicle's perception system due to their resilience to adverse environmental conditions. Radar can see through darkness and fog, and to a certain extent through rain, and snow, conditions that severely challenge or blind other sensors, such as cameras. This capability ensures a baseline of operational safety and functionality, regardless of the time of day or weather conditions, providing data on the range, velocity, and angle of other objects with a high degree of accuracy. The synergistic integration of high-frequency radar, 5G communication, and multimodal radar technologies greatly enhances the sensing capability and environmental adaptability of autonomous vehicle perception systems [20]. However, radars present a few challenges. In heavy rainfalls, the radio signals can suffer from attenuation, slightly reducing their effective range. Additionally, in dense urban environments, the radio waves can bounce off multiple surfaces before returning to the sensor. This multi-path reflection, or clutter, can create "ghost" targets, misleading the vehicle's perception system into “thinking” an object is present where there is none. The resolution of radar makes it challenging to classify objects with certainty, as it for examples struggles to distinguish between a pedestrian, a cyclist, or a stationary object, such as a signpost, based on its signature alone.

These challenges are particularly amplified in the context of last-mile delivery for autonomous vehicles. The ODD for these vehicles involves navigating complex and cluttered environments such as residential streets, sidewalks, and loading zones. Standard automotive radars are optimised for detecting large metallic objects, such as other vehicles, and may fail to reliably detect smaller, low-profile, or non-metallic items in these areas, including delivery packages, curbs, children's toys, or pets. The proximity to buildings, parked vehicles, and other street furniture exacerbates the multi-path reflection problem, making it more challenging to maintain a clear and accurate perception of the immediate surroundings.

IMUs measure the vehicle's linear acceleration and angular velocity using accelerometers and gyroscopes [16]. This data is integrated over time to estimate changes in velocity, position, and orientation (roll, pitch, yaw). They are fundamental for state estimation and enable dead reckoning navigation during periods when external positioning signals, such as GPS, are unavailable [16]. IMUs provide high-frequency motion data (often 100 Hz or higher), completely independent of external signals or environmental conditions, allowing continuous operation in tunnels, urban canyons, dense foliage, or indoors [16]. They are relatively low-cost, especially Micro-Electro-Mechanical Systems (MEMS) based units, compact, and consume little power [16]. IMU data is critical for stabilising perception data from other sensors (compensating for vehicle motion) and for providing the motion inputs needed for sensor fusion algorithms, such as Kalman filters [24]. The primary limitation of IMUs is drift, minor errors in acceleration and angular velocity measurements accumulate over time, leading to rapidly increasing errors in the estimated position and orientation [16]. This necessitates frequent corrections using absolute positioning sensors (such as GPS) or relative positioning derived from other sensors (e.g., LiDAR/camera-based SLAM). IMUs are sensitive to temperature changes and vibrations, which can affect their accuracy [16]. Accurate calibration is crucial, but it can be complex [16]. Magnetometers, sometimes included for heading reference, are unreliable in urban environments due to magnetic interference from buildings, vehicles, and infrastructure [25]. IMUs are indispensable for last-mile navigation due to frequent GPS signal degradation or loss in urban canyons, underpasses, or near tall buildings [16]. The high-frequency data helps maintain a smooth estimate of the vehicle's state, which is crucial for controlling robots navigating potentially uneven sidewalks or making frequent stops and starts. Cost-effective MEMS IMUs are generally sufficient, but robust fusion with GPS, LiDAR-SLAM, or visual odometry is essential to bind the inherent drift [16].

Ultrasonic sensors use high-frequency sound waves to detect the presence and distance of objects at very short ranges [26]. They operate on the principle of measuring the time-of-flight of emitted sound pulses reflecting off nearby objects. They are relatively inexpensive and easy to integrate. They can detect objects close to the vehicle (within a few meters), effectively covering blind spots often missed by cameras or LiDAR [23]. Their performance is largely unaffected by lighting conditions (work in darkness) or the colour/transparency of the object [27]. They are relatively robust in some adverse weather conditions [27]. Ultrasound sensors have a minimal detection range, of up to 4-5 meters depending on the sensor and conditions [26]. Their angular resolution is poor due to broad beam patterns, making it difficult to distinguish between closely spaced objects, determine object shape, or precisely locate small objects [27]. Performance degrades significantly at higher vehicle speeds [23]. They can be susceptible to interference from external ultrasonic noise sources [28]. They may struggle to detect soft, sound-absorbing materials [27]. Their primary utility in last-mile delivery is for low-speed, close-quarters manoeuvring, such as parking assistance for vans, docking at a specific delivery point, navigating very narrow passages, or detecting immediate low-lying obstacles like curbs right next to the vehicle or robot [26]. They can serve as a safety sensor for detecting the presence of people near loading doors [29]. Due to their limited range and resolution, they are unsuitable for primary navigation or obstacle avoidance at typical operational speeds but serve as a valuable, cost-effective complementary sensor for near-field safety and precision manoeuvring.

V2X encompasses technologies (primarily DSRC/IEEE 802.11p and C-V2X/cellular) that enable vehicles to communicate wirelessly with other vehicles (V2V), roadside infrastructure (V2I), pedestrians (V2P, often via smartphones), and the network or cloud (V2N) [30]. Its key role in perception is enabling cooperative perception, where sensor data and derived information are shared among connected entities [30]. V2X can dramatically extend a vehicle's perception range and awareness beyond the line-of-sight limitations of its onboard sensors [30]. By sharing data (raw sensor data, processed object lists, or intent information), vehicles can "see" around corners or through obstructions via the sensors of other connected agents. V2I communication can provide critical information, such as traffic signal phase and timing (SPaT), road hazard warnings, and work zone alerts [30]. This enhanced situational awareness can significantly improve safety and traffic efficiency, enabling coordinated manoeuvres such as platooning [30]. C-V2X offers the potential advantage of leveraging existing cellular infrastructure for V2N, potentially providing more exhaustive coverage compared to DSRC's reliance on dedicated Roadside Units (RSUs) [31]. The effectiveness of V2X, particularly V2V and V2I for cooperative perception, heavily depends on widespread adoption and deployment – a significant network effect challenge. Communication channels have limitations in terms of latency, reliability, bandwidth, and range, which can affect the timeliness and quality of shared perception data. Ensuring the security and authenticity of V2X messages is paramount to prevent malicious attacks (e.g., false hazard warnings, Sybil attacks) [30]. Privacy concerns exist regarding the sharing of vehicle data. Standardisation is still evolving, with ongoing debate and regional differences between DSRC (IEEE 802.11p/ITS-G5) and C-V2X (LTE-V2X, 5G-V2X) hindering global interoperability [31]. Deploying the necessary infrastructure (RSUs for DSRC, potentially upgraded cellular networks for C-V2X) involves significant cost and effort. Cooperative perception via V2X is potentially very valuable in dense, occluded urban environments typical of last-mile routes, allowing a delivery robot or van to perceive pedestrians or vehicles hidden from its direct view [30]. V2I communication providing traffic light status is crucial for safe intersection negotiation. V2N connectivity can be used for real-time updates to delivery routes, receiving customer instructions, remote monitoring, or potentially teleoperation under challenging situations. However, reliable connectivity (cellular or RSU coverage) might be inconsistent across all delivery zones, including dense urban areas or more remote suburban neighbourhoods. Security is especially critical for autonomous delivery vehicles, as it prevents theft, hijacking, or disruption of service.

GNSS plays a key role in autonomous last-mile delivery vehicles by providing essential positioning, navigation, and timing information. It enables these vehicles to accurately determine their location and navigate to delivery destinations, facilitating efficient route optimisation and real-time tracking. One of the primary strengths of GNSS is its global coverage, enabling positioning data to be available virtually anywhere. Its high accuracy, especially when complemented by augmentation systems like Real-Time Kinematic (RTK), can achieve centimetre-level precision, which is essential for operating in complex urban environments. Additionally, GNSS provides real-time data updates that support continuous adjustments during deliveries, making it a cost-effective solution widely available for implementation. GNSS has limitations as signal interference can occur in urban environments, where buildings or tunnels obstruct satellite signals, leading to degraded performance. Multipath effects, where signals bounce off surfaces, can further compromise accuracy. Latency issues may arise, affecting the system's responsiveness in dynamic traffic situations, and extreme weather conditions or satellite outages may challenge the reliability of GNSS. To effectively utilise GNSS in last-mile delivery vehicles, specific requirements must be met. Integrating GNSS with other technologies, such as inertial navigation systems (INS), LiDAR, and cameras, is vital for enhancing accuracy and reliability. Robust software algorithms are needed to process GNSS data and compensate for environmental errors. Energy-efficient solutions are essential to ensure continuous operation without overburdening the vehicle's power resources.

Real-time data exchange must be established to optimise routes and ensure safety while also adhering to safety standards compliance.

1.3 Autonomous Vehicle Architecture for Last-Mile Delivery

Autonomous vehicle for transport of goods considers the use of scalable processing capabilities at the edge with AI-based functions implemented into the perception domain and covering the edge computing capabilities implemented into vehicles of different sizes using the same generic architecture as illustrated in Figure 1.4 [7].

1.3.1 Localisation and High-Definition Map

Localisation is critical for the safe and efficient operation of last-mile delivery autonomous vehicles. By employing high-definition (HD) maps that detail urban infrastructure, such as road types, curb locations, and traffic signals, these vehicles can navigate complex environments more effectively.

The maps incorporate real-time data updates to reflect changing road conditions, enhancing navigational accuracy and safety.

1.3.2 Perception Implementation

Perception in last-mile delivery autonomous vehicles integrates AI to identify and classify various objects within the vehicle's vicinity. This involves

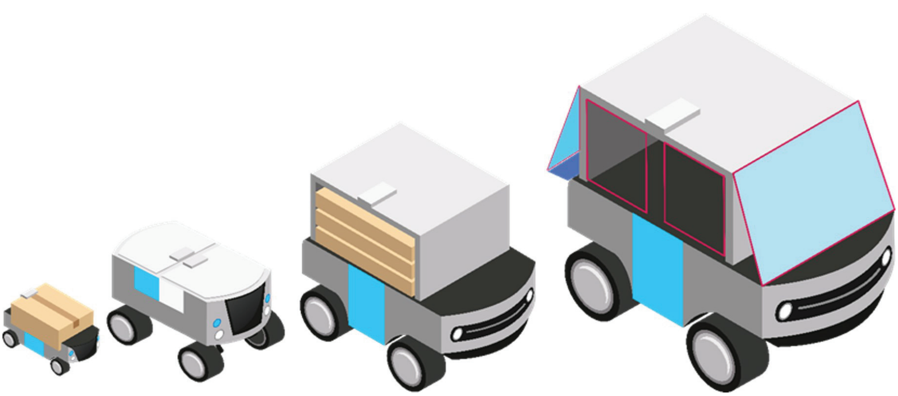

Figure 1.4 Vehicles and scalability. Source: [7].

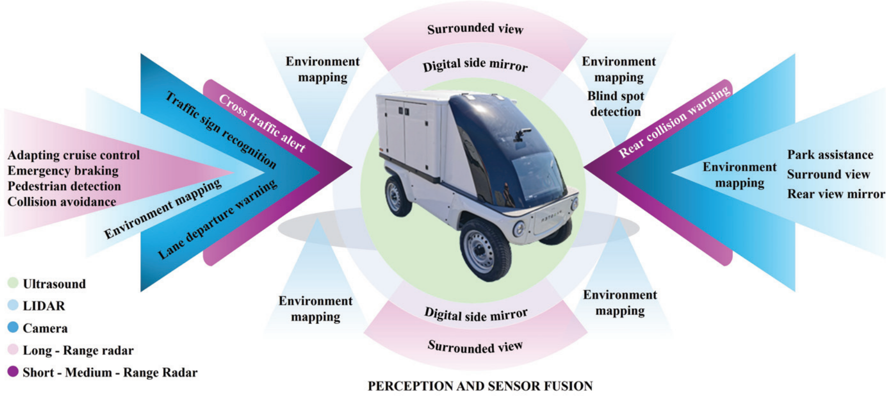

Figure 1.5 Perception and sensors fusion. Source: [8].

a fusion of data from multiple sources, where machine learning (ML) algorithms help predict potential obstacles and dynamic changes in the environment.

The resultant data enhances situational awareness and informs decision-making processes. An overview of the sensors used in the perception and sensor fusion platforms for autonomous vehicles for last-mile delivery of goods is illustrated in Figure 1.5.

Sensor fusion, AI processing and decision-making

The overall system architecture of the autonomous vehicle comprises controllers for the perception, sensor fusion and actions for the vehicle's actuators based on the sensed environment, objectives, and constraints. It is divided into three primary blocks: detection, perception, and decision policy, as illustrated in Figure 1.6 [9].

The main system management components consist of the Operating System (OS) based on Linux Ubuntu, and the middleware based on the ROS.

ROS1 is a high-level API for evaluating sensor data and controlling actuators.

The integration activities on sensor fusion, combines homogeneous and heterogeneous data from different sources like the perception sensors (LiDAR, cameras, ultrasonic sensors, etc.) to facilitate AI processing, decision-making, and planning.

The perception workflow for image recognition, object detection and tracking are illustrated in Figure 1.7 [9]. Technology wrappers are used to integrate different protocols, data formats, and interfaces seamlessly.

AI processing and the autonomous systems are integrated using perception sensors for mapping the environment, HW/SW components for the acquisition, processing, aggregation, analysis, and interpretation of data, AI-based algorithms and methods for situation assessment, action planning, cognitive decision-making, and actuators for acting on the steering, braking and propulsion systems.

Various AI frameworks, such as Python, PyTorch, Keras, and TensorFlow, as well as several machine vision libraries like OpenCV, SimpleCV, the Point

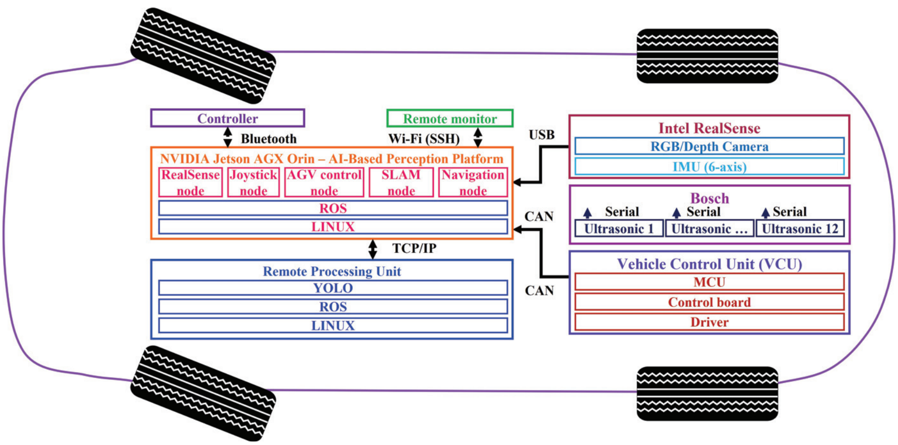

Figure 1.6 Autonomous vehicle for last-mile delivery – Platform integration components. Source: Adapted from [9].

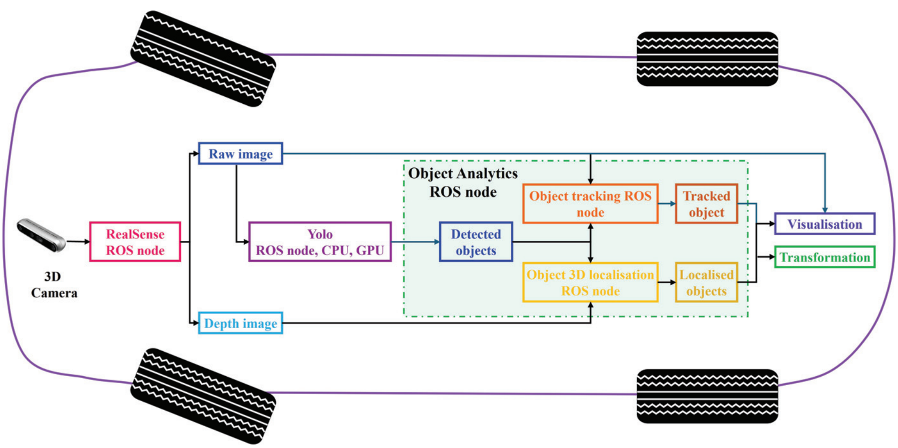

Figure 1.7 Perception workflow for image recognition, object detection and tracking. Source: Adapted from [9].

Cloud library, and YOLO, and AI-based computing platforms like NVIDIA Jetson AGX Orin, were evaluated for integration into different layers of autonomous vehicle architecture.

These AI-based frameworks, algorithms, libraries, and platforms were utilised during various phases of the autonomous vehicle platform demonstrator's development.

The decision-making relies on sensor fusion and AI processing.

Furthermore, the vehicle platform features several actuators and control units, including the electric steering servo and throttle/speed control for autonomous driving, speech information/recognition for use-case service/security purposes, NFC/mobile locking/unlocking systems, and wireless/wired emergency stops for safety reasons.

Perception sensors and navigation

The platform was integrated with the NVIDIA Jetson AGC Orin, which provides AI-based perception and sensor fusion capabilities [9]. To abstract the sensor brand and interface from Orin, parsing of the ultrasonic sensor electronic control unit (ECU) data were implemented and an interface provided (independent of ultrasonic system used).

Having the vehicle control unit (VCU) interpret the sensor data also enables it to have an emergency brake function. This function is set so that if the vehicle is autonomous mode, the vehicle's VCU sends a signal to disable drive to the CAN relay, which in turn triggers engaging of the electronic park brake.

To get the ultrasonic sensors to have an impact on the vehicle motion there needs to be several interfaces defined where the data and information can flow. There are two distinct types of interfaces in form of CAN and ROS topics. An illustration of the interfaces is given in Figure 1.8.

The ultrasonic sensor hardware abstraction layer (HAL) provides an interface that includes the CAN output from the VCU interface but provides it as an ROS topic that other nodes inside NVIDIA Jetson AGX Orin platform can make use of. The topic is described in a message and provides information for each sensor in a separate variable.

Tracking System for Platooning

Platooning of autonomous vehicles refers to a formation of multiple autonomous vehicles travelling closely together in a single-file line, with the lead vehicle controlling the speed and direction for the following vehicles. The group of connected autonomous vehicles exchange information, allowing

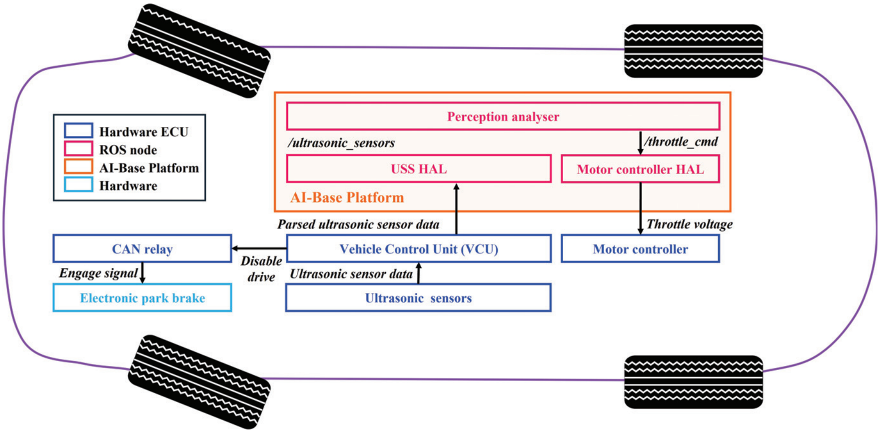

Figure 1.8 System overview. Source: Adapted from [9].

them to drive in a coordinated manner, with very small spacings, while still travelling safely at relatively high speeds [36].

The fleets of last-mile delivery autonomous vehicles utilise platooning coordinated driving style to enhance logistics efficiency, increase road capacity, improve traffic flow, reduce delivery times, and load goods from common warehouse hubs.

Platooning requires the development of robust and reliable V2V communication, multi-sensor perception systems, control algorithms, and infrastructure, including dedicated lanes or specific road configurations, to function optimally.

Information on the vehicles' speeds, positions, accelerations, decelerations, and other relevant data for the vehicles in the platoon, as well as for those joining or leaving the platoon, is crucial, as all the vehicles in the platoon need to react efficiently and safely in real-time.

Several information flow topologies (IFTs) have been traditionally used in the literature, such as predecessor-following (PF), two-predecessor-following (TPF), and bidirectional (BDL). The advancement of communication systems increased the use of more general schemes such as r-predecessor following (rPLF). The dynamic platoon nature, with vehicles changing their relative position over time, also adds complexity to the topology of communications [36].

The platooning style of driving can be implemented for autonomous vehicles equipped with V2X connectivity by conveying traffic information (e.g., GNSS, speed, or signal timing) and using either unlicensed V2X (ITS-G5) or cellular V2X (LTE-V2X/NR-V2X) [37].

The platooning of last-mile delivery autonomous vehicles can be implemented using the vehicle's perception sensors (e.g., cameras, ultrasound) for areas with good visibility, and an alternative solution can complement the V2X system.

The following section presents the implementation of a tracking system for platooning, as demonstrated in the ECSEL JU AI4CSM project [9], utilising an NVIDIA Jetson AGX Orin processing platform running the Ubuntu Linux operating system, which offers AI capabilities for the vehicle's perception domain. The vehicle's onboard unit communicates with its sensors through the ROS operating system as middleware.

The autonomous vehicle, illustrated in the Figure 1.15, is equipped with multiple perception sensors, including LiDAR, a depth camera, and ultrasound sensors.

For the implementation, the Intel RealSense D455 RGB-D depth camera was used as a sensor to detect the logo mark placed on the rear of the lead vehicle, serving as a target for the follower vehicle to follow.

The logo in Figure 1.9 is attached to the rear of the vehicle. The logo design can make it difficult to distinguish it from other circular objects or signs with straight lines within the circle, such as no stopping/parking signs.

To detect the logo, two different detection model frameworks were tested, YOLOv5 [38] and YOLOv8 [39]. YOLO is a computer vision model developed by Ultralytics and is part of the "You Only Look Once" (YOLO) family of models, known for their high inference speed, making them suitable for

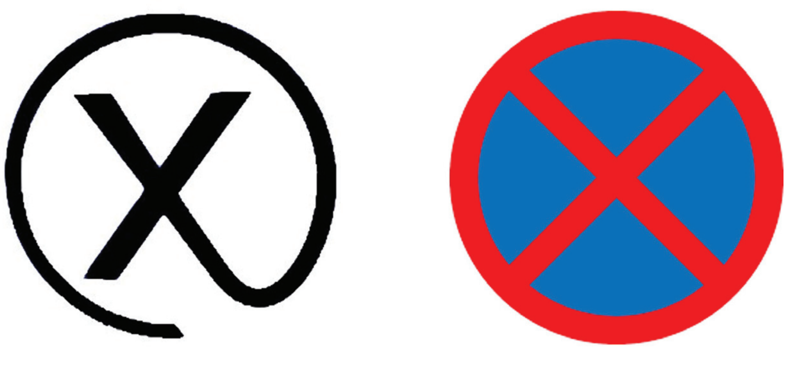

Figure 1.9 Tracking system logo and no stopping/parking sign.

real-time applications. Both frameworks are implemented in PyTorch, which contributes to their ease of use, speed, and accuracy.

YOLOv8 is a further development of YOLOv5, not only in terms of detection architecture but also in the development framework. YOLOv8 provides enhanced documentation and a streamlined setup for training and deploying a detection model. The use case presented in this paper addresses the training of a simple detection model. As a result, the nano model for both YOLOv5 and YOLOv8 was used, as it provides optimal inference speed and an architecture suitable for our detection problem.

To create an effective detection model, the dataset is the most critical factor. Datasets with diverse photos, logos on vehicles, and t-shirts in different lighting conditions were collected for training the models.

The base dataset contains approximately 1000 images, where 10% of the pictures do not contain any logos, as many false positives were encountered during training on images with logos only.

Augmentation techniques are critical when training a robust model. YOLOv5 and YOLOv8 models have built-in augmentation techniques, which we utilised with the default settings. In addition to the built-in augmentation techniques, we augmented the dataset by pasting logos onto images from images recorded with a vehicle, as illustrated in Figure 1.10.

The training results for the best-performing YOLOv5 and YOLOv8 models showed that training for YOLOv8 is more stable. The mAP (mean Average Precision) metric for object detection, mAP50-0.95, converges to a higher value for YOLOv8, indicating that the model can predict the correct class with greater confidence.

To enhance the system's robustness, a tracking algorithm was implemented on top of the detection model. The tracking algorithm is built into YOLOv8, which is the ByteTrack AI algorithm [39].

The algorithm can detect and continuously track multiple objects by assigning each one a unique ID, considering all detected objects (not just high-confidence ones), which can improve its tracking accuracy even in challenging conditions, such as occlusion.

The ByteTrack AI algorithm can be applied directly, with no final tuning required, using only the bounding boxes with pixel-level information from the detection model. Figure 1.11 illustrates a constructed case where two logos appear. Since tracking is implemented, the following vehicle knows which one to follow, as it has an assigned leader ID.

This implementation makes the system more robust against false positives, as these will receive unique IDs, and the following vehicle is locked to its leader's ID.

Figure 1.10 Logos pasted on driving data.

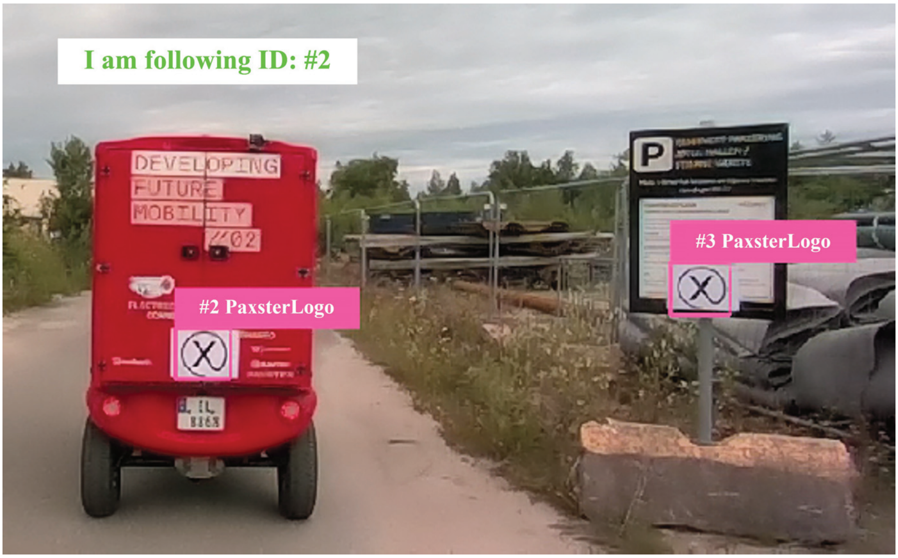

Figure 1.11 The detection and tracking module returning one unique bounding box.

1.3.3 Prediction, Decision-Making, Planning and Route Optimisation

The last-mile delivery autonomous vehicles decision-making framework utilises predictive algorithms to assess potential interactions and outcomes based on current vehicle positioning and environmental factors. By analysing previously collected data, these vehicles can efficiently plan paths that minimize risks and enhance delivery speed. Planning algorithms incorporate multi-layered decision-making processes that prioritize safety and operational efficiency.

Through machine learning algorithms, AI assesses historical data, including traffic patterns and weather conditions, to optimize delivery routes. This reduces delivery times and operational costs, allowing for quicker and more efficient deliveries.

1.3.3.1 Odometry and path planning

Vehicles with rear-wheel drive and front-wheel steering, is referred to as Ackermann steering. When the vehicle makes a turn, the wheel on the outer side of the turn has a slightly larger turning radius than the wheel on the inner side. The sharper the turn, the greater the difference in turning radii between the wheels.

Most vehicles, uses kingpins to control each wheel, with a single servo to control the vehicle's heading. In an Ackermann-steered vehicle, the heading can be simplified to a bicycle model, where the average angle of the front wheels is directly correlated with the vehicle's heading. The inner and outer wheel angle is different depending on left or right turn (counterclockwise or clockwise turns) and must be taken into considerations during the odometry and path planning development. Odometry, based on sensor fusion and the data from the integrated motion sensors, is used to estimate the position over time and improve the position accuracy, velocity and attitude.

The configuration is illustrated in Figure 1.12, where $\emptyset$ is the steering angle of the front wheels. For our scenario the steering angle of the front wheels $\emptyset$ is derived from the pure pursuit equation and given below [50, 51], where $\overrightarrow{y}$ is the vector pointing from the rear axle to the camera, $\overrightarrow{z}$ is the vector pointing from rear axle to the tracked object (tuneable look-ahead), and $\overrightarrow{L}_a$ is the length of wheelbase.

$$\emptyset = \frac{2L_a}{|\overrightarrow{z}|}\sqrt{1-\left(\frac{\overrightarrow{y}\,\overrightarrow{z}}{|\overrightarrow{y}|\,|\overrightarrow{z}|}\right)^2} \tag{1.1}$$

Pure pursuit is an algorithm for path tracking that determines the angular velocity command required for the autonomous vehicle to navigate from its present location to a designated look-ahead point in front of it [50, 51]. The linear velocity is considered constant, allowing for adjustments to the vehicle's linear speed at any time. The algorithm continuously adjusts the look-ahead point along the path in accordance with the vehicle's current

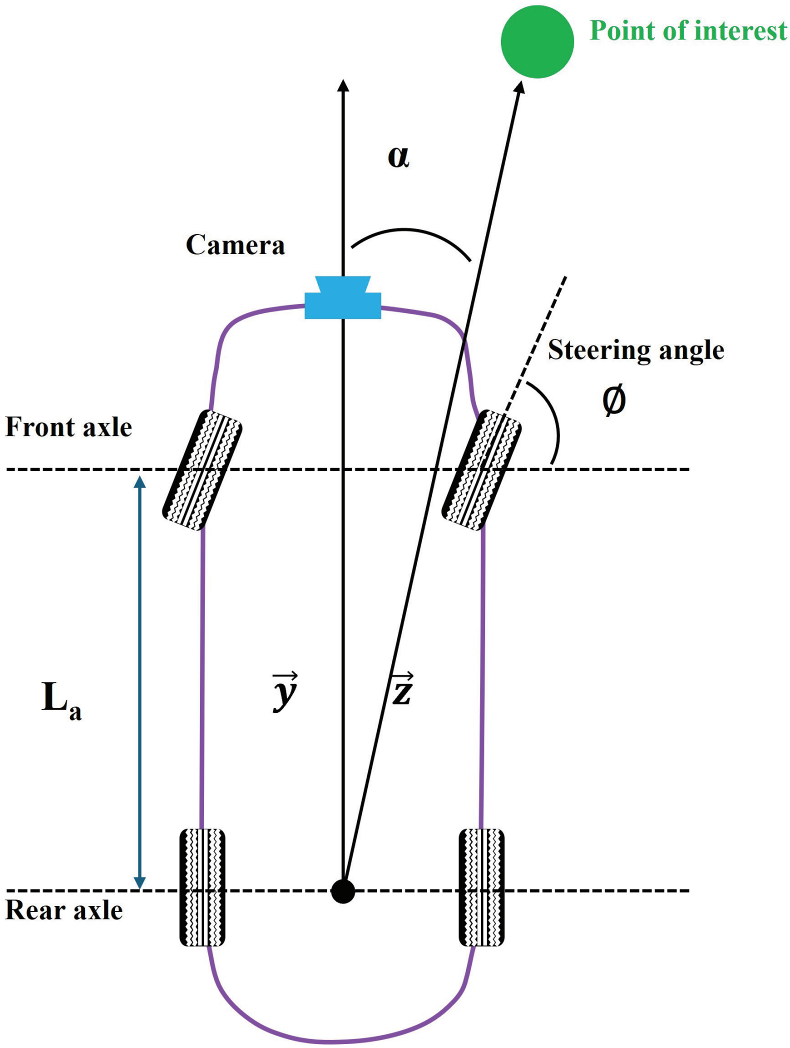

Figure 1.12 Odometry pure pursuit algorithm calculation illustration [50, 51].

position, effectively enabling the vehicle to consistently pursue a point in front of it until it reaches the end of the path. The pure pursuit controller and its algorithms runs continuously in a loop to keep a vehicle on right track as illustrated in Figure 1.13 and described below.

As the vehicle advances, the following process loop immediately repeats, continuously updating the steering and speed commands to ensure the vehicle remains aligned with the path [50, 51].

- **Define Path:** Initially, a trajectory is assigned for the vehicle to follow.
- **Determine current position:** The system identifies the vehicle's current location and its orientation.
- **Find next point:** The algorithm examines the defined path ahead to identify the next point.
- **Compute steering angle:** Based on its current position, orientation, and the next point's location, the controller inputs these values into the pure pursuit algorithm to determine the necessary steering angle, to ensure the curvature of an ideal circular arc that the vehicle can follow smoothly.
- **Actuate angle and speed:** The computed steering angle and a speed command is transmitted to the vehicle's steering and motor controllers.

Continuous monitoring is crucial for ensuring the safe operation of last-mile delivery autonomous vehicles. Real-time data is transmitted to centralised systems where remote operatives can oversee vehicle performance

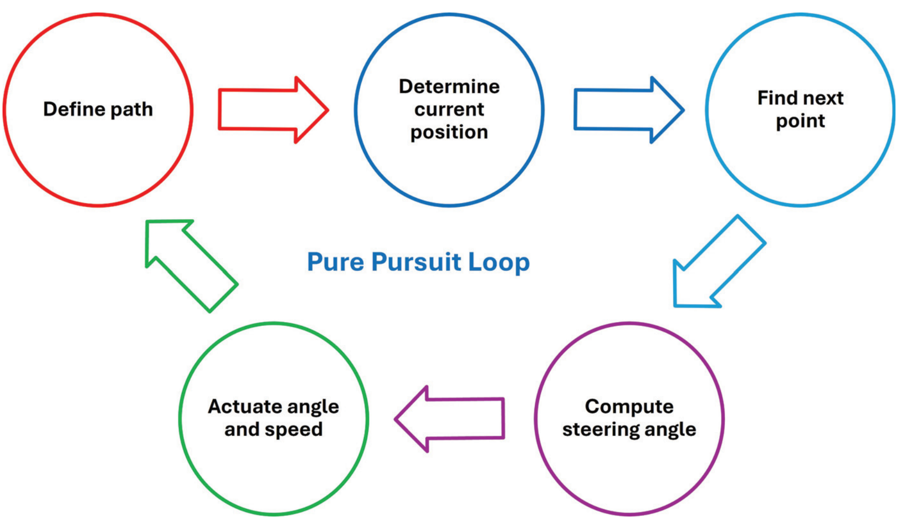

Figure 1.13 The pure pursuit loop to keep the vehicle on track [50, 51].

and intervene when necessary. This hybrid approach helps manage the unexpected challenges presented by complex urban environments.

Safety strategies for last-mile delivery autonomous vehicles involve redundant systems to prevent malfunction and protocols for engaging human overseers in critical situations. Security considerations also encompass the protection of data communication between vehicles and management systems to prevent unauthorized access and ensure operational integrity.

1.4 Edge AI Platforms

In the context of autonomous vehicles, an edge AI platform refers to the integrated hardware, software, and edge AI stack, as well as the data responsible for executing the complex computational tasks required to perform autonomous functions and intelligent decision-making. This includes processing vast amounts of sensor data, running sophisticated AI algorithms for perception, sensor fusion, localisation, path planning, and motion control, and ultimately making real-time driving decisions.

For last-mile delivery AVs, the definition of an AI platform is heavily influenced by the specific constraints of the application. Delivery vehicles, robots and smaller vans operate under stringent size, weight, power, and cost constraints compared to larger robotaxis or trucks.

This requires highly optimised, edge-computing platforms. High-performance Graphics Processing Units (GPUs) are used for training and complex inference, while autonomous vehicle platforms rely on GPUs and specialised, power-efficient AI accelerators such as Field-Programmable Gate Arrays (FPGAs), Neural Processing Units (NPUs), Tensor Processing Units (TPUs), or custom ASICs integrated into System-on-a-Chip (SoC). NVIDIA's Jetson AGX Orin platform is an example targeted at such edge applications, including last-mile delivery vehicles and robots.

The software stack typically runs on a real-time operating system (RTOS) or a standard OS like Linux with real-time extensions, often utilising middleware like ROS for modularity and communication between different software components (perception, planning, control).

The platform hosts the AI libraries (e.g., TensorFlow, PyTorch, etc.) and the specific algorithms implementing the vehicle's autonomous capabilities. The key distinction for last-mile AI platforms is the focus on energy efficiency, computational density, and cost-effectiveness, which are suitable for scalable deployment on smaller, battery-powered vehicles.

1.4.1 Robot Operating System

ROS is not a traditional operating system in the sense of Windows or Linux. Instead, it is a flexible framework of software libraries and tools that simplify the creation of complex robot applications [45, 46].

ROS is an open-source framework for writing robot software, providing a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms. It has also been recently used in autonomous vehicles, as seen in the case of this implementation for last-mile delivery autonomous vehicles and other autonomous systems. ROS is rather a middleware, a set of software frameworks for robot software development [35, 40, 47, 48].

There are two versions of ROS: ROS 1, which evolved with community contributions, and ROS 2, released in 2017. ROS 2 incorporates real-time capabilities, improved security, and better support for distributed systems by leveraging the Data Distribution Service (DDS) standard [45]. A table of key differences between ROS 1 and ROS 2 can be seen in Table 1.2 [11].

ROS provides services expected from an operating system, including hardware abstraction, low-level device control, implementation of commonly

Table 1.2 Summary of ROS 2 Features Compared to ROS 1 [11]

Category	ROS 1	ROS 2
Network Transport	Tailored protocol built on TCP/UDP	Existing standard (DDS), with abstraction supporting addition of others
Network Architecture	Central name server (roscore)	Peer-to-peer discovery
Platform Support	Linux	Linux, Windows, macOS
Client Libraries	Written independently in each language	Sharing a common underlying C library (rcl)
Node vs. Process	Single node per process	Multiple nodes per process
Threading Model	Callback queues and handlers	Swappable executor
Node State Management	None	Lifecycle nodes
Embedded Systems	The ROSSerial client library used for small, embedded devices	The micro-ROS stack integrates microcontrollers with standard ROS 2
Parameter Access	Auxiliary protocol built on XMLRPC	Implemented using service calls
Parameter Types	Type inferred when assigned	Type declared and enforced

used functionality, message passing between processes, and package management.

The core of ROS is its anonymous publish/ subscribe messaging system. A process (called a “node”) that has information to share can publish it to a specific “topic”. Other nodes interested in that type of information can subscribe to the topic to receive the messages, which creates a modular, decoupled architecture where different parts of the system can be developed and tested independently [42, 43] .

The integration of ROS into autonomous vehicles involves several key aspects, such as [41]:

- **Hardware abstraction**, where ROS provides a standardised interface to a wide variety of sensors and actuators, meaning that a high-level autonomous driving algorithm can be developed independently of the specific hardware being used. In this context, a “LiDAR driver” node could publish data from a particular brand of LiDAR to a standardised topic, and a perception node could subscribe to that topic without needing to know the specifics of the LiDAR hardware.
- **Inter-process communication**: Autonomous vehicles have a multitude of processes running concurrently: perception, localisation, planning, and control. ROS’s messaging system allows these processes to communicate with each other in a reliable and time-synchronised manner, even if they are running on different computers within the vehicle.
- **Ecosystem and tools**: ROS has an ecosystem of tools for visualisation, simulation, and data logging. Tools like RViz enable developers to visualise sensor data and the vehicle’s state in 3D, while Gazebo provides a realistic simulation environment for testing algorithms without requiring a physical vehicle.

ROS is used as a platform for developing perception and sensor fusion systems [49] in the implementation of the last-mile delivery autonomous vehicle presented in this chapter.

As a result, several features of the platform were analysed, evaluated and integrated into the vehicle architecture. These elements are described below:

- **AI integration**: The modular nature of ROS makes it easy to integrate AI and machine learning libraries. The typical approach used was to have a ROS node that utilises a library such as TensorFlow or PyTorch to perform object detection or semantic segmentation on camera images. The results of this process (e.g., the locations of other vehicles and pedestrians) were then published to a ROS topic for other nodes to use.

- **Sensor fusion**: The implemented autonomous vehicles rely on a variety of sensors, including cameras, LiDAR, ultrasound, IMUs, etc. ROS provided a framework for fusing the data from these different sensors to create a more accurate and robust understanding of the environment. A “sensor fusion” node could subscribe to topics containing data from the camera, LiDAR, and IMU, and then use a filter (like a Kalman filter) to combine this data and produce a unified representation of the environment.

The original implementation of the functions for the last-mile delivery autonomous vehicles was implemented in the original version of ROS (ROS 1). The test results show that the system has some limitations in terms of real-time performance and security, and the newer version, ROS 2 [44, 45], is integrated into the new vehicle design due to the following [35]:

- **Real-time capabilities**: ROS 2 is built on top of the DDS standard, which provides real-time, reliable, and scalable communication. The capabilities are necessary for autonomous vehicle implementation to reduce delays and increase the system’s robustness.
- **Security**: ROS 2 includes a better security framework that provides features like authentication, encryption, and access control, which are essential for protecting the vehicle from cyberattacks.
- **Quality of Service (QoS)**: ROS 2 allows specifying QoS policies for each publisher and subscriber, which enables the control of aspects like reliability, durability, and latency, ensuring that critical data is delivered in a timely and reliable manner.

A set of principles and specific requirements guides the design of ROS 2, including distribution, abstraction, asynchrony, and modularity, as well as several design requirements such as security, integration of embedded systems, use of diverse communication networks, real-time computing, and product readiness.

The ROS 2 APIs provide access to communication patterns, such as services and actions, which are organised under the concept of a node. ROS 2 also provides APIs for parameters, timers, launch, and other auxiliary tools, which can be used to design a robotic system. ROS 2 issues a request-response style pattern, known as services. Request-response communication provides a clear association between a request and its corresponding response, which can be helpful when ensuring that a task was completed or received. A unique communication pattern of ROS 2 is the action. Actions are goal-oriented and asynchronous, providing communication interfaces with request-response capabilities, periodic feedback, and the ability to be cancelled. The middleware architecture of ROS 2 consists of several abstraction layers distributed across many decoupled packages. These abstraction layers enable multiple solutions for the required functionality, such as various middleware or logging solutions. Additionally, the distribution across various packages allows users to replace components or take only the necessary pieces of the system, which may be important for certification [11, 12].

Figure 1.14 displays the layers within ROS 2 as it is a set of software libraries and tools for building robot and autonomous systems applications. ROS2 builds upon DDS and contains a DDS abstraction layer. Users do not need to be aware of the DDS APIs due to this abstraction layer. This layer enables ROS2 to have high-level configurations and optimises the utilisation of DDS. Additionally, due to the use of DDS, ROS2 does not require a master process [7, 11, 12].

The client libraries provide access to the core communication APIs. They are tailored to each programming language to make them more idiomatic and take advantage of language-specific features. Communication is agnostic to how the system is distributed across compute resources, whether they are in the same process, a different process, or even a different processing unit. A

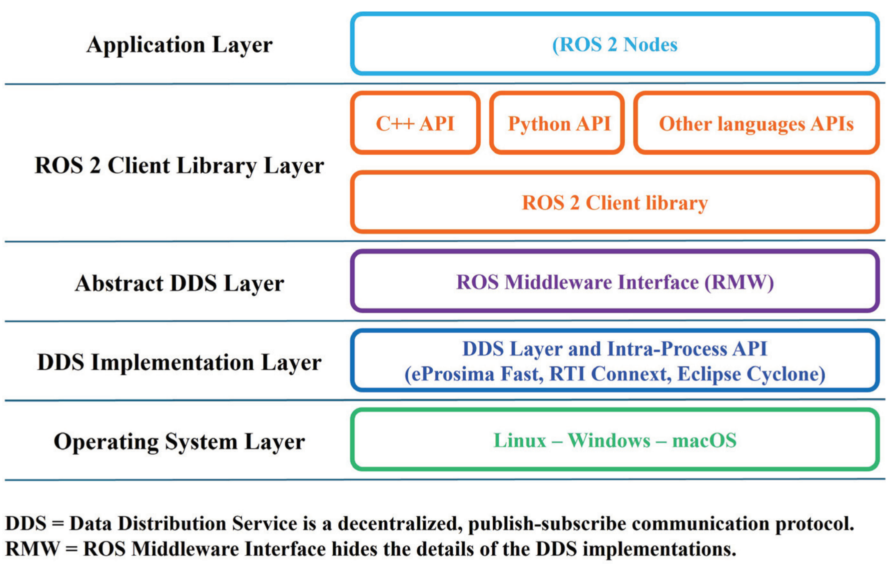

Figure 1.14 ROS 2 architecture [7, 12].

user may distribute their application across multiple machines and processes, and even leverage cloud compute resources, with minimal changes to the source code. ROS 2 can connect to cloud and edge resources over the internet.

The client libraries rely on an intermediate interface that provides standard functionality to each client library. This library is written in C and is used by all the client libraries, although it is not required for their operation. The middleware abstraction layer, called RMW (ROS Middleware), provides the essential communication interfaces. The vendors for each middleware implement the RMW interface and are made interchangeable without code changes.

Users may choose different RMW implementations, and thereby different middleware technologies, based on various constraints such as performance, software licensing, or supported platforms. The network interfaces (e.g. topics, services, actions) are defined, and ROS 2 defines these types using specific format files.

Communications are agnostic to the location of endpoints within machines and processes. Nodes written as components can be allocated to any process as a configuration, allowing multiple nodes to be configured to share a process, thereby conserving system resources or reducing latency.

1.5 Future Considerations and Research

1.5.1 Deployment Considerations

Autonomous vehicles for last-mile delivery can reshape the final step of the supply chain, from the distribution centre to customers' doorsteps. The primary advantage of autonomous delivery is the potential for significant cost reduction and increased efficiency. These vehicles can operate around the clock, resulting in faster delivery times and improved fleet utilisation. They can also be designed to be more environmentally friendly, contributing to more sustainable logistics practices.

The deployment of unmanned ground and aerial autonomous vehicles and mobile robots requires careful planning and consideration of various factors such as regulatory compliance, integration with existing logistics infrastructure, and public perception. Addressing these factors is essential for the successful adoption of autonomous delivery vehicles in urban areas. Partnerships with local governments and logistics providers can facilitate smoother integration and expansion of services.

Figure 1.15 The autonomous vehicle [10].

The success of the technology depends on navigating complex urban environments, which include everything from busy streets to unpredictable behaviour of pedestrians and traffic participants.

The path to widespread adoption of last-mile delivery autonomous vehicles is filled with challenges. The technology is still evolving, and ensuring the safety and reliability of autonomous systems is a top priority as the vehicles must be able to navigate a wide range of real-world scenarios, from inclement weather to unexpected road closures.

Public acceptance and the impact on the workforce are also critical considerations. Gaining the trust of consumers and integrating these vehicles into daily life without causing disruption is key.

Generative AI can enhance the interaction between autonomous vehicles and users through natural language processing, allowing users to communicate with delivery systems using voice commands, while facilitating seamless user experiences where customers can track orders or interact directly with service interfaces.

Addressing future trends and overcoming technological, regulatory, and social challenges, while developing new technologies and AI-assisted tools, is key to unlocking the full potential of autonomous last-mile delivery.

1.5.2 Future research

Autonomous last-mile delivery applications are leveraging advancements in edge AI platforms and sensor fusion technologies (camera, LiDAR, IMU, ultrasound, V2X, etc.), specifically tailored to meet the unique requirements and constraints of these applications, including close-quarters navigation, interaction with VRUs, and severe cost and power limitations.

Key architectural concepts, including various levels and strategies for sensor fusion, are employed alongside enabling algorithms (e.g., deep learning methods such as CNNs and Transformers) and hardware components (e.g., GPUs, specialised accelerators, and edge computing platforms). Significant integration challenges, encompassing sensor calibration, data synchronisation, computational load management, power consumption, cost-effectiveness, and fault tolerance, still need to be addressed.

The pressing research challenges are achieving robust perception in adverse conditions, reliable VRU detection, handling sensor limitations, and ensuring the safety and validation of autonomous systems.

Key future research directions include novel fusion architectures, end-to-end learning, self-supervised methods, enhanced V2X integration, explainable edge AI (XAI) and interpretable edge AI (IAI) for fusion, lightweight models for edge deployment, and advanced simulation for validation, all aimed at advancing the safety, reliability, and efficiency of autonomous last-mile delivery.

Future work includes the concept for a dedicated architecture and a multimodal AI-based autonomous vehicle platform for perception, automated control, and decision-making in delivering goods in controlled environments. The work addresses evaluating the integration of data from multiple sensors. Multimodal AI and generative AI enable real-time correlation and a more comprehensive understanding of the vehicle's surroundings, allowing for vehicle control through voice and gesture commands.

Future work plans to address the adoption of new concepts provided by Software-Defined and AI-Defined Vehicles (SDV/AIDV) architectures, where the vehicle's features and functionality are determined by the holistic interplay between sensors/actuators, the hardware, software, AI platforms, and data and ROS is well-suited to this new paradigm, as it provides a flexible and modular platform for developing and deploying a wide range of applications.

As last-mile delivery autonomous vehicles operate in fleets and are connected, there is a growing trend towards offloading the computational tasks to the edge. ROS 2's support for DDS makes it easy to extend the vehicle's communication system to include edge-based services.

Further research is addressing the advancements of autonomous delivery vehicles, focusing on enhancing the human-machine interfaces between

AI-driven vehicles and humans operating individual vehicles or fleets, improving security measures to protect against tampering, and refining the sustainability aspects of autonomous vehicle operations through various use cases and business models. Human-autonomous system interaction and the development of new human-machine interfaces require further investigation to enhance trust, acceptance, and the successful integration of last-mile delivery autonomous vehicles and IoRT into daily life.

Future research combining last-mile delivery autonomous vehicles, IoRT, and edge AI needs to focus on advancing decentralised swarm intelligence, enabling fleets of autonomous vehicles to collaborate and make collective decisions to improve scalability and resilience, allowing the fleet to adapt to unforeseen events in real-time. Significant research should be directed towards developing more advanced and energy-efficient edge AI algorithms for predictive analysis, anticipating delivery demand, and optimising V2X communication.

Robust security, privacy-preserving protocols, and trustworthiness within the IoRT framework are crucial for safeguarding sensitive delivery data and protecting autonomous systems from cyber threats, making this an important future area of research.

Further research investigation includes the concept for a dedicated architecture and a multimodal AI-based autonomous vehicle platform for perception, automated control, and decision-making in delivering goods in controlled environments. The research also includes evaluating the integration of data from multiple sensors, combined with decision support that integrates small language models, vision language models, and agentic AI. Multimodal AI and generative AI enable real-time correlation and a more comprehensive understanding of the vehicle's surroundings, enabling the implementation of vehicle control through voice and gesture commands.

Another key focus is on standardisation and the development of AI-assisted tools that improve the capabilities of frameworks like ROS and increase the efficiency of designing last-mile delivery autonomous vehicles.

1.6 Conclusion

Early implementations of last-mile delivery autonomous vehicle technologies demonstrate significant potential for cost reduction and efficiency enhancement in last-mile logistics. Nonetheless, lessons from these operations underscore the complexity of urban environments and the need for continuous adaptation of technologies to meet real-world challenges. Key takeaways include prioritising safety, enhancing AI-based decision-making across diverse scenarios, and maintaining robust validation methodologies for autonomous systems.

AI integrates data from multiple sensors, including cameras, LiDAR, IMU, ultrasound sensors, and GNSS, to create a comprehensive understanding of the vehicle's surroundings. This fusion of sensory information enhances the vehicle's decision-making capabilities, as it can analyse and interpret complex datasets to ascertain the safest and most efficient operation.

AI serves as the backbone of autonomous delivery systems, driving their efficiency, safety, and user engagement. By harnessing advanced algorithms and technologies, these systems can provide faster, reliable, and more cost-effective delivery solutions, revolutionising the logistics and delivery sectors. As AI technology continues to develop, its role in enhancing autonomous delivery operations is expected to grow, paving the way for even more innovative solutions in this area.

Acknowledgements

This publication has received funding through the projects ECSEL Joint Undertaking (JU) AI4CSM, Chips JU EdgeAI, and Chips JU MOSAIC. The ECSEL JU AI4CSM "Automotive Intelligence for Connected Shared Mobility" project was supported by the ECSEL Joint Undertaking and its members, including top-up funding from Austria, Belgium, the Czech Republic, Italy, Latvia, Lithuania, the Netherlands, and Norway under grant agreement No. 101007326. The Chips JU EdgeAI, "Edge AI Technologies for Optimised Performance Embedded Processing," project is supported by the Chips Joint Undertaking and its members, including top-up funding from Austria, Belgium, France, Greece, Italy, Latvia, the Netherlands, and Norway, under grant agreement No. 101097300. The Chips JU MOSAIC "A Mosaic of Essential Electronic Components and Systems (ECS) for our Automated Digital Future in Industry and Mobility" project is supported by the Chips Joint Undertaking and its members including top-up funding by Austria, Belgium, Czech Republic, Denmark, France, Greece, Israel, Italy, Latvia, Netherlands, Norway, Poland, and Türkiye under grant agreement No 101194414. Funded by the European Union. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or Chips Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

The authors would like to acknowledge the contributions of Espen Teigen, ISPAS AS, Norway, for the software implementation of the odometry and path planning modules described in this article during his tenure at NxTECH AS, Norway.

References

[1] M. G. Augusto et al., "Autonomous Van and Robot Last-Mile Logistics Platform: A Reference Architecture and Proof of Concept Implementation," Logistics, vol. 9, no. 1, pp. 10–10, Jan. 2025, https://doi.org/10.3390/logistics9010010.

[2] BotPenguin, "Last Mile Delivery," Botpenguin.com, 2024. https://botpenguin.com/glossary/last-mile-delivery.

[3] Wise Systems, "The State of the Autonomous Last-Mile Market in 2023," Wise Systems, Nov. 02, 2023. https://www.wisesystems.com/blog/autonomous-last-mile-market-2023/

[4] V. Engesser, E. Rombaut, L. Vanhaverbeke, and P. Lebeau, "Autonomous Delivery Solutions for Last-Mile Logistics Operations: A Literature Review and Research Agenda," Sustainability, vol. 15, no. 3, p. 2774, Feb. 2023, https://doi.org/10.3390/su15032774.

[5] O. Vermesan et al., "4 Internet of Robotic Things -Converging Sensing/Actuating, Hyperconnectivity, Artificial Intelligence and IoT Platforms." Available at: https://www.riverpublishers.com/pdf/ebook/chapter/RP_9788793609105C4.pdf

[6] O. Vermesan et al., "Internet of Robotic Things Intelligent Connectivity and Platforms," Frontiers in Robotics and AI, vol. 7, Sep. 2020, https://doi.org/10.3389/frobt.2020.00104.

[7] O. Vermesan, "Embedding ROS and Edge AI-Based Perception Capabilities in AV Platforms," European Conference on EDGE AI Technologies and Applications – EEAI 2023, 17-19 October 2023, Athens, Greece. https://edge-ai-tech.eu/wp-content/uploads/2023/11/2023-10-19_Presentation_C03_2_Conference_EDGE-AI_Athens_EEAI_2023.pdf.

[8] O. Vermesan, "Trustworthy Edge AI Solutions in Autonomous Vehicle Perception and Sensor Fusion," European Conference on EDGE AI Technologies and Applications – EEAI 2024, 21-23 October 2024 Cagliari, Italy. https://edge-ai-tech.eu/wp-content/uploads/2024/11/2024-10-23_Presentation_B03-3_Conference_EDGE-AI_Cagliari_EEAI_2024.pdf.

[9] O. Vermesan, (Ed.), “Report on integration activities for reliable, robust and secure perception systems and platforms for ECAS vehicles. Deliverable D5.15 ECSEL JU AI4CSM project.,” Dec. 2024.

[10] O. Vermesan and R. Bahr, “Fremtidens mobilitet” in Elektronikknett, 12 June 2025, (in Norwegian) https://www.elektronikknett.no/ai4csm-autonome-kjoretoy-eu/fremtidens-mobilitet/3207829,

[11] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics, vol. 7, no. 66, May 2022, https://doi.org/10.1126/scirobotics.abm6074.

[12] Y. Maruyama, S. Kato, and T. Azumi, “Exploring the performance of ROS2,” Proceedings of the 13th International Conference on Embedded Software - EMSOFT ’16, 2016, https://doi.org/10.1145/2968478.2968502.

[13] L. Li, W. Zhang, X. Wang, T. Cui and C. Sun, “NLOS Dies Twice: Challenges and Solutions of V2X for Cooperative Perception,” in IEEE Open Journal of Intelligent Transportation Systems, vol. 5, pp. 774-782, 2024, https://doi.org/10.1109/OJITS.2024.3492211.

[14] J. Koon, “Solving the Last-Mile Delivery Problem,” Semiconductor Engineering, Jul. 06, 2023. https://semiengineering.com/solving-the-last-mile-delivery-problem/

[15] HowToRobot, “Delivery robots: Automating the last mile,” HowToRobot, 2024. https://howtorobot.com/expert-insight/delivery-robots-automating-last-mile

[16] GuideNav, “What are the Advantages and Disadvantages of Inertial Measurement Units (IMUs)?,” GuideNav: The Global Standard in Inertial Navigation Excellence, 2024. https://guidenav.com/what-are-the-advantages-and-disadvantages-of-inertial-measurement-units-imus%EF%BC%9F/

[17] The British Standards Institution, “Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) - Specification.” 2020, ISBN 9780539067354, Avalable at: https://www.bsigroup.com/globalassets/localfiles/en-gb/cav/pas1883.pdf.

[18] TechNexion, “Embedded cameras in delivery robots – their role and impact,” *TechNexion*, Jan. 10, 2025. https://www.technexion.com/resources/embedded-cameras-in-delivery-robots-their-role-and-impact/

[19] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, “Sensor and Sensor Fusion Technology in Autonomous Vehicles: a Review,” Sensors, Vol. 21, No. 6, p. 2140, Mar. 2021, https://doi.org/10.3390/s21062140.

[20] Z. Feng, "Application and Development of Radar Sensors in Autonomous Driving Technology," Applied and Computational Engineering, Vol. 140, No. 1, pp. 48–52, Mar. 2025, doi: https://doi.org/10.54254/2755-2721/2025.21277.

[21] HESAI, "The Role of Lidar in the Future of Autonomous Vehicles," *HESAI*, Jun. 12, 2023. https://www.hesaitech.com/the-role-of-lidar-in-the-future-of-autonomous-vehicles/.

[22] A. Singh, "Transformer-Based Sensor Fusion for Autonomous Driving: A Survey," 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Paris, France, 2023, pp. 3304-3309, https://www.doi.org/10.1109/ICCVW60793.2023.00355.

[23] G. Mittermaier. "How important are sensors for autonomous vehicles?," Md-elektronik.com, Apr. 25, 2025. https://www.md-elektronik.com/en/how-important-are-sensors-for-autonomous-vehicles/

[24] Fiveable, "Inertial measurement units (IMU)", Autonomous Vehicle Systems Class Notes, Fiveable, 2024. https://library.fiveable.me/autonomous-vehicle-systems/unit-2/inertial-measurement-units-imu/study-guide/ff0fWwVL7g5SSoEF

[25] JOUAV, "A Complete Guide to Inertial Measurement Unit (IMU)," *JOUAV*, 2025. https://www.jouav.com/blog/inertial-measurement-unit.html

[26] Leadvent Group, "The Essential Role of Sensor Accuracy in Autonomous Vehicle Safety," *Leadventgrp.com*, Sep. 10, 2024. https://www.leadventgrp.com/blog/the-role-of-sensor-accuracy-in-ensuring-av-safety

[27] Fiveable "Ultrasonic sensors", Autonomous Vehicle Systems Class Notes, Fiveable 2024. https://library.fiveable.me/autonomous-vehicle-systems/unit-2/ultrasonic-sensors/study-guide/uTe34rwjgIartZFT

[28] Caradas, "Ultrasonic Sensors in Cars" *Caradas*, Dec. 02, 2023. https://caradas.com/understanding-ultrasonic-sensors-in-cars/

[29] G. Sankar, "How embedded vision is transforming last mile delivery with delivery robots," *e-con Systems*, Oct. 26, 2022. https://www.e-consystems.com/blog/camera/applications/autonomous-mobile-robots-amr/how-embedded-vision-is-transforming-last-mile-delivery-with-delivery-robots/

[30] T. Huang *et al.*, "V2X Cooperative Perception for Autonomous Driving: Recent Advances and Challenges," *arXiv.org*, Nov. 06, 2023. https://arxiv.org/abs/2310.03525

[31] Y. He, B. Wu, Z. Dong, J. Wan and W. Shi, "Towards C-V2X Enabled Collaborative Autonomous Driving," in IEEE Transactions on Vehicular Technology, vol. 72, no. 12, pp. 15450-15462, Dec. 2023, https://www.doi.org/10.1109/TVT.2023.3299844.

[32] S. Raghavan, B. Pistone, D. Gibb, J. Gasser, N. Pednekar, and S. Marzani, "Software-defined Vehicles, GenAI, IoT – The Path to AI-Defined Vehicles, Amazon Web Services," *Amazon Web Services*, Mar. 13, 2025. https://aws.amazon.com/blogs/industries/software-defined-vehicles-genai-iot-the-path-to-ai-defined-vehicles/.

[33] S. Gajendra, "What are AI-Defined Vehicles and Why They Are the Future of Automotive?" Arm Newsroom, May 28, 2025. https://newsroom.arm.com/blog/what-are-ai-defined-vehicles.

[34] A. Jose and B. Daniel, "Reimagining / Redefining SDVs: Embracing the AI-Defined Vehicle Era," Frost & Sullivan, Jun. 10, 2024. https://www.frost.com/growth-opportunity-news/reimagining-redefining-sdvs-embracing-the-ai-defined-vehicle-era/.

[35] P. Estefo, J. Simmonds, R. Robbes, and J. Fabry, "The Robot Operating System: Package reuse and community dynamics," Journal of Systems and Software, vol. 151, pp. 226–242, May 2019, https://doi.org/10.1016/j.jss.2019.02.024.

[36] M. Martínez-Díaz, C. Al-Haddad, F. Soriguera, and C. Antoniou, "Platooning of connected automated vehicles on freeways: a bird's eye view," Transportation Research Procedia, vol. 58, pp. 479–486, Jan. 2021, https://doi.org/10.1016/j.trpro.2021.11.064.

[37] O. Vermesan et al., "Automotive Intelligence Embedded in Electric Connected Autonomous and Shared Vehicles Technology for Sustainable Green Mobility," Frontiers in Future Transportation, vol. 2, Aug. 2021, https://doi.org/10.3389/ffutr.2021.688482.

[38] G. Jocher, "ultralytics/yolov5," GitHub, Aug. 21, 2020. https://github.com/ultralytics/yolov5

[39] G. Jocher, A. Chaurasia, and J. Qiu, "YOLOv8 by Ultralytics," GitHub, 2023. https://github.com/ultralytics/ultralytics

[40] M. Quigley *et al.*, "ROS: an open-source Robot Operating System." Available at: https://robotics.stanford.edu/$~$ang/papers/icraoss09-ROS.pdf.

[41] A. Ademovic, "An Introduction to Robot Operating System: The Ultimate Robot Application Framework," *Toptal Engineering Blog*. https://www.toptal.com/robotics/introduction-to-robot-operating-system.

[42] H. Kutluca, “Robot Operating System 2 (ROS 2) Architecture,” Medium, Jan. 14, 2021. https://medium.com/software-architecture-foundations/robot-operating-system-2-ros-2-architecture-731ef1867776.
[43] Ultralytics, “ROS Quickstart,” *Ultralytics.com*, 2024. https://docs.ultralytics.com/guides/ros-quickstart/.
[44] S. Dam, “AZoRobotics,” *AZoRobotics*, Jul. 20, 2013. https://www.azorobotics.com/Article.aspx?ArticleID=140
[45] Open Robotics, “ROS.org | Powering the world’s robots,” *Ros.org*, 2020. https://www.ros.org/
[46] L. Shalom, “Introduction to Robot Operation System (ROS) - DZone IoT,” *dzone.com*. https://dzone.com/articles/ros-robotic-operation-systems
[47] R. Ospina and K. Itakura, “Obstacle detection and avoidance system based on layered costmaps for robot tractors,” *Smart Agricultural Technology*, vol. 11, p. 100973, Aug. 2025, https://doi.org/10.1016/j.atech.2025.100973.
[48] F. Fatima, M. H. Akhter, M. Omama, and S. A. Khan, “ROS-Enabled Autonomous Vehicle Architecture within CARLA: A Comprehensive Overview,” *Transportation Research Procedia*, vol. 84, pp. 129–136, Mar. 2025, https://doi.org/10.1016/j.trpro.2025.03.055.
[49] K. Gil, J. Yuk, and J. Shin, “Integrated path planning and control for autonomous vehicle platooning,” *Control Engineering Practice*, vol. 164, pp. 106470–106470, Jul. 2025, https://doi.org/10.1016/j.conengprac.2025.106470.
[50] R.C. Coulter, “Implementation of the Pure Pursuit Path Tracking Algorithm”, *The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania*. January 1992. https://www.ri.cmu.edu/pub_files/pub3/coulter_r_craig_1992_1/coulter_r_craig_1992_1.pdf
[51] H. Jain and P. Babel. “A comprehensive Survey of PID and Pure Pursuit Control Algorithms for Autonomous Vehicle navigation”, *Arxiv org*. September 2024. https://arxiv.org/pdf/2409.09848