



## **Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle**



Università degli Studi di Genova Scuola Politecnica & Facoltà di Ingegneria



Laurea Magistrale in Ingegneria Strategica International MSc in Engineering Technology for  
Strategy and Security of Genoa University

## **Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle**

Advisors: Prof. Francesco Bellotti

Prof. Riccardo Berta

Co: Dr. Matteo Fresta

Dr. Alessandro Pighetti

Candidate: Amirmohammad Dalvand

5872846

[S5872846@studenti.unige.it](mailto:S5872846@studenti.unige.it)



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### Abstract

Transferring reinforcement learning controllers from simulation to physical hardware remains limited by the sim-to-real gap. This thesis develops and validates a CARLA-based digital twin of a 1:10-scale autonomous vehicle and evaluates its use as a training environment for zero-shot sim-to-real transfer of a PPO-based autonomous parking controller.

The physical platform is a modified Traxxas Slash vehicle with an NVIDIA Jetson Orin Nano, eight HC-SR04 ultrasonic sensors, and an Arduino Due for low-level sensing and actuation. The CARLA digital twin replicates the vehicle's wheelbase, steering limits, mass, tire and engine parameters, and sensor placement, and matches its sensing and control timing.

Digital twin fidelity was validated using a deterministic scripted controller executed identically in simulation and reality across 20 paired runs per platform (40 runs total, 213 samples per trajectory at a 0.05 s sampling period). Yaw, lateral acceleration, and lateral offset were compared across three parking phases (slot-search, approach-alignment, reverse-parking) using the Pearson Correlation Coefficient (PCC) and Maximum Normalized Cross-Correlation (MNCC). Yaw showed near-perfect agreement ( $PCC = 0.99\text{--}1.00$  across all phases), lateral offset remained strong ( $PCC = 0.79\text{--}0.98$ ), and lateral acceleration was the most sensitive channel, degrading during the reverse-parking phase ( $PCC = 0.83\text{--}0.85$ ) due to complex lateral coupling under steering reversal.

A PPO controller was then trained entirely within the digital twin, using an 11-dimensional observation space restricted to onboard-equivalent sensing (eight ultrasonic ranges, yaw, and two-axis acceleration) and a continuous action space (steering, throttle, brake). Evaluated over 20 trials per platform with no fine-tuning, the policy achieved a 60% success rate in simulation and 55% on the physical vehicle ( a 5-percentage-point gap). For comparison, an identical policy trained on an unmodified ("vanilla") CARLA vehicle achieved 61% in simulation but only 7% in reality, demonstrating that digital-twin validation, not the learning algorithm itself, is what closes the reality gap.

These results indicate that a quantitatively validated digital twin can support direct zero-shot transfer of a parking policy, and identify ultrasonic perception noise during the final approach ( rather than vehicle dynamics mismatch ) as the dominant remaining source of the reality gap.

**Keywords:** Digital Twin, Sim-to-Real Transfer, Reinforcement Learning, Autonomous Parking, CARLA, Proximal Policy Optimization.



# Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



## Table of Contents

|                                                                            |    |
|----------------------------------------------------------------------------|----|
| Chapter 1 - Introduction .....                                             | 9  |
| 1.1 Background and Motivation .....                                        | 9  |
| 1.2 Problem Statement .....                                                | 11 |
| 1.3 Research Objectives .....                                              | 13 |
| Chapter 2 – Literature Review .....                                        | 16 |
| 2.1 Introduction .....                                                     | 16 |
| 2.2 Autonomous Vehicle Systems .....                                       | 18 |
| 2.2.1 Levels of Vehicle Automation .....                                   | 19 |
| 2.2.2 Architecture of Autonomous Vehicle Systems .....                     | 19 |
| 2.2.3 Sensor Technologies in Autonomous Vehicles .....                     | 21 |
| 2.2.4 Challenges in Autonomous Vehicle Development .....                   | 21 |
| 2.2.5 Autonomous Parking as a Representative Autonomous Driving Task ..... | 22 |
| 2.3 Autonomous Parking Systems .....                                       | 22 |
| 2.3.1 Classical Autonomous Parking Approaches .....                        | 23 |
| 2.3.2 Learning-Based Autonomous Parking .....                              | 24 |
| 2.3.3 Reinforcement Learning for Autonomous Parking .....                  | 24 |
| 2.3.4 Perception for Autonomous Parking .....                              | 25 |
| 2.3.5 Challenges in Reinforcement Learning-Based Parking .....             | 26 |
| 2.3.6 Research Trends and Remaining Challenges .....                       | 26 |
| 2.4 Digital Twin Technology .....                                          | 27 |
| 2.4.1 Evolution of Digital Twin Technology .....                           | 27 |
| 2.4.2 Definition and Characteristics of Digital Twins .....                | 28 |
| 2.4.3 Digital Twin Architecture .....                                      | 28 |
| 2.4.4 Digital Twins in Robotics .....                                      | 29 |
| 2.4.5 Digital Twins for Autonomous Vehicles .....                          | 29 |
| 2.4.6 Digital Twins for Reinforcement Learning .....                       | 30 |
| 2.4.7 Challenges in Digital Twin Development .....                         | 30 |
| 2.4.8 Research Gaps .....                                                  | 31 |
| 2.5 Simulation Platforms for Autonomous Vehicle Research .....             | 31 |
| 2.5.1 CARLA Simulator .....                                                | 32 |
| 2.5.2 Microsoft AirSim .....                                               | 33 |
| 2.5.3 SVL Simulator .....                                                  | 33 |
| 2.5.4 Gazebo .....                                                         | 33 |
| 2.5.5 Webots .....                                                         | 34 |
| 2.5.6 Comparative Analysis of Autonomous Driving Simulators .....          | 34 |
| 2.5.7 Justification for Selecting CARLA .....                              | 34 |
| 2.6 Deep Reinforcement Learning .....                                      | 35 |



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



|                                                                  |    |
|------------------------------------------------------------------|----|
| 2.6.1 Fundamentals of Reinforcement Learning .....               | 35 |
| 2.6.2 Deep Reinforcement Learning .....                          | 36 |
| 2.6.3 Value-Based Reinforcement Learning .....                   | 36 |
| 2.6.4 Policy Gradient Methods .....                              | 37 |
| 2.6.5 Continuous-Control Reinforcement Learning Algorithms ..... | 37 |
| 2.6.6 Proximal Policy Optimization (PPO) .....                   | 38 |
| 2.6.7 Reinforcement Learning for Autonomous Parking .....        | 38 |
| 2.6.8 Challenges of Deep Reinforcement Learning .....            | 39 |
| 2.6.9 Comparison of Deep Reinforcement Learning Algorithms ..... | 39 |
| 2.6.10 Justification for Selecting PPO .....                     | 40 |
| 2.7 Simulation-to-Real Transfer in Autonomous Driving .....      | 40 |
| 2.7.1 Sources of the Reality Gap .....                           | 40 |
| 2.7.2 Sim-to-Real Transfer Strategies .....                      | 41 |
| 2.7.3 Digital Twins for Sim-to-Real Transfer .....               | 42 |
| 2.7.4 Validation of Sim-to-Real Transfer .....                   | 42 |
| 2.7.5 Zero-Shot Sim-to-Real Transfer .....                       | 43 |
| 2.7.6 Research Gaps .....                                        | 43 |
| 2.8 Review of Related Work .....                                 | 44 |
| 2.8.1 Conventional Autonomous Parking Systems .....              | 44 |
| 2.8.2 Reinforcement Learning-Based Autonomous Parking .....      | 45 |
| 2.8.3 Digital Twins for Autonomous Vehicle Research .....        | 45 |
| 2.8.4 Simulation Platforms for Reinforcement Learning .....      | 46 |
| 2.8.5 Sim-to-Real Reinforcement Learning .....                   | 46 |
| 2.8.6 Critical Analysis of Existing Literature .....             | 47 |
| 2.8.7 Research Gap .....                                         | 47 |
| 2.8.8 Positioning of the Present Research .....                  | 48 |
| 2.9 Research Gaps and Motivation .....                           | 48 |
| 2.10 Chapter Summary .....                                       | 50 |
| Chapter 3 - System Architecture .....                            | 52 |
| 3.1 Overview of the Proposed Framework .....                     | 52 |
| 3.2 Physical Vehicle Platform .....                              | 54 |
| 3.2.1 Traxxas Slash Platform .....                               | 54 |
| 3.2.2 Steering System .....                                      | 54 |
| 3.2.3 Electronic Speed Controller and Longitudinal Motion .....  | 55 |
| 3.2.4 Power Distribution and Electrical Integration .....        | 55 |
| 3.3 Sensor Suite .....                                           | 56 |
| 3.3.1 Ultrasonic Sensors .....                                   | 56 |
| 3.3.2 Sensor Placement .....                                     | 56 |



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



|                                                  |    |
|--------------------------------------------------|----|
| 3.3.3 Sensor Limitations .....                   | 57 |
| 3.4 Embedded Computing Architecture .....        | 57 |
| 3.4.1 NVIDIA Jetson Orin Nano.....               | 57 |
| 3.4.2 Arduino Due.....                           | 57 |
| 3.4.3 Serial Communication Interface .....       | 58 |
| 3.5 Software Architecture.....                   | 58 |
| 3.5.1 High-Level Control Layer.....              | 58 |
| 3.5.2 CARLA Integration Layer .....              | 58 |
| 3.5.3 Real-Time Control Loop.....                | 59 |
| 3.6 Continuous Action Space Design .....         | 59 |
| Chapter 4 - CARLA Digital Twin Development ..... | 60 |
| 4.1 Introduction .....                           | 60 |
| 4.2 CARLA Simulator Overview .....               | 60 |
| 4.3 Digital Twin Design Objectives .....         | 61 |
| 4.3.1 Dynamic Consistency .....                  | 61 |
| 4.3.2 Perception Consistency .....               | 62 |
| 4.3.3 Control Interface Consistency.....         | 62 |
| 4.3.4 Temporal Consistency.....                  | 62 |
| 4.4 Vehicle Model Adaptation .....               | 62 |
| 4.4.1 Base Vehicle Selection .....               | 62 |
| 4.4.2 Wheelbase Adjustment.....                  | 63 |
| 4.4.3 Steering System Calibration.....           | 63 |
| 4.4.4 Vehicle Mass and Inertia .....             | 64 |
| 4.4.5 Tire Radius and Ground Interaction.....    | 64 |
| 4.4.6 Throttle and Torque Calibration .....      | 64 |
| 4.5 Ultrasonic Sensor Emulation.....             | 65 |
| 4.5.1 Motivation.....                            | 65 |
| 4.5.2 Sensor Modelling Approach.....             | 65 |
| 4.5.3 Sensor Placement Replication.....          | 66 |
| 4.5.4 Sensor Limitations in Simulation.....      | 66 |
| 4.6 Parking Environment Modelling .....          | 67 |
| 4.6.1 Parking Scenario Design.....               | 67 |
| 4.6.2 Non-Player Vehicles.....                   | 67 |
| 4.6.3 Environment Determinism.....               | 67 |
| 4.7 Synchronization and Timing Design .....      | 68 |
| 4.7.1 Synchronous Simulation Mode.....           | 68 |
| 4.7.2 Physics Timestep.....                      | 68 |
| 4.7.3 Control Frequency Matching .....           | 68 |



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



|                                                         |    |
|---------------------------------------------------------|----|
| 4.7.4 Importance of Timing Consistency .....            | 68 |
| 4.8 Summary .....                                       | 69 |
| Chapter 5 - Digital Twin Validation Methodology .....   | 70 |
| 5.1 Introduction .....                                  | 70 |
| 5.2 Validation Objectives .....                         | 71 |
| 5.3 Deterministic Reference Policy .....                | 71 |
| 5.3.1 Motivation for a Scripted Controller .....        | 71 |
| 5.3.2 Open-Loop vs. Closed-Loop Validation .....        | 72 |
| 5.3.3 State-Machine Architecture .....                  | 72 |
| 5.3.4 Slot-Search Phase .....                           | 73 |
| 5.3.5 Approach-Alignment Phase .....                    | 73 |
| 5.3.6 Reverse Parking Phase .....                       | 73 |
| 5.4 Experimental Setup .....                            | 74 |
| 5.4.1 Parking Environment .....                         | 74 |
| 5.4.2 Initial Conditions .....                          | 74 |
| 5.4.3 Number of Experimental Runs .....                 | 74 |
| 5.4.4 Data Acquisition .....                            | 75 |
| 5.5 Validation Metrics .....                            | 75 |
| 5.5.1 Motivation for Statistical Metrics .....          | 75 |
| 5.5.2 Pearson Correlation Coefficient (PCC) .....       | 76 |
| 5.5.3 Maximum Normalized Cross-Correlation (MNCC) ..... | 76 |
| 5.5.4 Evaluated Signals .....                           | 77 |
| 5.6 Uniform and Non-Uniform Sampling .....              | 77 |
| 5.6.1 Motivation .....                                  | 77 |
| 5.6.2 Uniform Sampling .....                            | 77 |
| 5.6.3 Non-Uniform Sampling .....                        | 78 |
| 5.7 Signal Alignment and Processing .....               | 78 |
| 5.8 Importance of Phase-Based Validation .....          | 78 |
| 5.9 Summary .....                                       | 79 |
| Chapter 6 - Reinforcement Learning Controller .....     | 80 |
| 6.1 Introduction .....                                  | 80 |
| 6.2 Reinforcement Learning Formulation .....            | 81 |
| 6.2.1 Markov Decision Process Representation .....      | 81 |
| 6.2.2 Episodic Learning Structure .....                 | 82 |
| 6.3 Observation Space Design .....                      | 82 |
| 6.3.1 Design Philosophy .....                           | 82 |
| 6.3.2 Ultrasonic Sensor Observations .....              | 82 |
| 6.3.3 Vehicle Orientation .....                         | 83 |



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| 6.3.4 Lateral Acceleration .....                                              | 83 |
| 6.3.5 Final Observation Vector .....                                          | 83 |
| 6.4 Continuous Action Space .....                                             | 84 |
| 6.4.1 Motivation for Continuous Control.....                                  | 84 |
| 6.4.2 Steering Control.....                                                   | 84 |
| 6.4.3 Longitudinal Control.....                                               | 84 |
| 6.5 Reward Function Design .....                                              | 84 |
| 6.5.1 Importance of Reward Engineering .....                                  | 84 |
| 6.5.2 Overall Reward Structure.....                                           | 85 |
| 6.5.3 Success Reward .....                                                    | 85 |
| 6.5.4 Distance-Based Reward .....                                             | 86 |
| 6.5.5 Heading Alignment Reward.....                                           | 86 |
| 6.5.6 Collision Penalty .....                                                 | 86 |
| 6.6 PPO Algorithm .....                                                       | 86 |
| 6.6.1 Motivation for PPO .....                                                | 86 |
| 6.6.2 Policy Optimization Objective .....                                     | 87 |
| 6.6.3 Actor-Critic Structure.....                                             | 87 |
| 6.6.4 Implementation and Reproducibility Details .....                        | 87 |
| 6.7 Neural Network Architecture .....                                         | 88 |
| 6.7.1 Input Layer.....                                                        | 88 |
| 6.7.2 Hidden Layers.....                                                      | 88 |
| 6.7.3 Output Layer .....                                                      | 88 |
| 6.8 Training Procedure .....                                                  | 88 |
| 6.8.1 Simulation-Only Training .....                                          | 88 |
| 6.8.2 Domain Randomization .....                                              | 88 |
| 6.8.3 Episode Reset Strategy .....                                            | 89 |
| 6.9 Zero-Shot Sim-to-Real Deployment.....                                     | 89 |
| 6.9.1 Deployment Objective .....                                              | 89 |
| 6.9.2 Real-Time Inference Pipeline.....                                       | 89 |
| 6.9.3 Importance of Zero-Shot Evaluation.....                                 | 89 |
| 6.10 Summary .....                                                            | 90 |
| Chapter 7 - Experimental Results and Discussion .....                         | 91 |
| 7.1 Introduction .....                                                        | 91 |
| 7.2 Digital-Twin Fidelity Validation, Deterministic Scripted Controller ..... | 91 |
| 7.3 Yaw-Angle Validation .....                                                | 92 |
| 7.3.1 Yaw Trajectory Analysis .....                                           | 92 |
| 7.3.2 Discussion of Yaw Consistency .....                                     | 93 |
| 7.4 Lateral Acceleration Validation .....                                     | 93 |



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



|                                                            |     |
|------------------------------------------------------------|-----|
| 7.4.1 Lateral Acceleration Analysis .....                  | 93  |
| 7.4.2 Sources of Dynamic Discrepancy .....                 | 94  |
| 7.5 Lateral Offset Validation .....                        | 95  |
| 7.5.1 Lateral Offset Analysis.....                         | 95  |
| 7.5.2 Geometric Interpretation .....                       | 95  |
| 7.6 Uniform vs Non-Uniform Sampling Analysis.....          | 95  |
| 7.6.1 Uniform Sampling Results.....                        | 95  |
| 7.6.2 Non-Uniform Sampling Results.....                    | 96  |
| 7.6.3 Interpretation.....                                  | 96  |
| 7.7 PPO Training Results .....                             | 96  |
| 7.7.1 Learning Curve Evolution.....                        | 96  |
| 7.7.2 Convergence Analysis.....                            | 97  |
| 7.7.3 Emergent Parking Behaviour .....                     | 97  |
| 7.8 Zero-Shot Sim-to-Real Deployment.....                  | 97  |
| 7.8.1 Deployment Procedure.....                            | 97  |
| 7.8.2 Real-World Parking Performance (PPO deployment)..... | 98  |
| 7.8.3 Performance Degradation Analysis .....               | 98  |
| 7.9 Failure Case Analysis .....                            | 98  |
| 7.9.1 Perception-Related Failures .....                    | 98  |
| 7.9.2 Dynamic Mismatch Failures .....                      | 99  |
| 7.10 Comparison with Existing Approaches .....             | 99  |
| 7.11 Discussion .....                                      | 99  |
| 7.12 Summary .....                                         | 100 |
| Chapter 8 Conclusion and Future Work.....                  | 100 |
| 8.1 Introduction .....                                     | 100 |
| 8.2 Summary of Contributions .....                         | 100 |
| 8.3 Answers to the Research Questions.....                 | 102 |
| 8.4 Limitations .....                                      | 103 |
| 8.5 Future Work.....                                       | 103 |
| 8.5.1 Improved Perception Systems.....                     | 103 |
| 8.5.2 Advanced Vehicle Dynamics Modelling .....            | 103 |
| 8.5.3 Broader and Dynamics-Oriented Randomization .....    | 104 |
| 8.5.4 End-to-End Learning .....                            | 104 |
| 8.5.5 Multi-Scenario Autonomous Driving.....               | 104 |
| 8.6 Final Remarks .....                                    | 104 |
| References .....                                           | 105 |



# Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



## Chapter 1 - Introduction

### 1.1 Background and Motivation

Autonomous Driving (AD) systems have emerged as one of the most active research areas in intelligent transportation, robotics, and artificial intelligence. Their primary objective is to enable vehicles to perceive the surrounding environment, make decisions, and execute driving manoeuvres without human intervention. In recent years, significant advances in sensing technologies, embedded computing, machine learning, and simulation frameworks have accelerated the development of autonomous systems for both industrial and research applications. Among the numerous functionalities required by autonomous vehicles, low-speed manoeuvring tasks such as automated parking remain particularly challenging due to the high precision demanded in confined spaces and the strong dependence on reliable short-range perception.

Automated parking systems traditionally rely on modular pipelines consisting of parking-slot detection, trajectory planning, and trajectory tracking. These systems often employ geometric methods and model-based controllers designed around explicit assumptions regarding vehicle kinematics, obstacle geometry, and environmental conditions. While such approaches can achieve high accuracy in controlled settings, their performance may degrade when subjected to uncertain environments, sensor noise, actuator nonlinearities, or dynamic mismatches between the assumed vehicle model and the real platform. Furthermore, the design and tuning of classical controllers can become increasingly complex as the environment grows more variable and unstructured.

For this reason, learning-based approaches have gained increasing attention within the autonomous-driving community. In particular, Deep Reinforcement Learning (DRL) has emerged as a promising paradigm for autonomous control because it allows an agent to learn optimal driving behaviours directly through interaction with the environment rather than through manually engineered rules. DRL methods have demonstrated strong performance in navigation, lane following, obstacle avoidance, and automated parking tasks. Algorithms such as Proximal Policy Optimization (PPO) are especially attractive due to their stability, sample efficiency, and compatibility with continuous control problems.

Despite these advances, training DRL agents directly on physical vehicles remains impractical in many scenarios. Reinforcement learning typically requires millions of interactions with the environment before converging to an effective policy. Performing such training entirely on hardware is associated with several major limitations. First, repeated experimentation can lead to mechanical wear and hardware degradation. Second, unsafe or exploratory actions may damage the vehicle or surrounding objects. Third, collecting large-scale physical data is time-consuming and expensive. These limitations make simulation an essential component of modern autonomous-driving research.

Simulation environments provide a controlled and repeatable framework for developing and validating autonomous systems. Among the available simulators, CARLA has become one of the most widely adopted platforms for autonomous-driving research. CARLA provides realistic urban environments, configurable sensors, traffic simulation, and physics-based vehicle dynamics while supporting integration with external machine-learning frameworks. By enabling large-scale data generation and safe controller testing, simulation dramatically accelerates development cycles while reducing experimental cost and risk.

However, the usefulness of simulation depends fundamentally on how accurately the simulated environment reproduces the behaviour of the real-world system. This challenge is commonly referred to as the sim-to-real gap or reality gap. Policies trained in simulation often exhibit substantial performance degradation when deployed on physical platforms because the simulator cannot perfectly



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



model all aspects of the real system. Differences may arise from inaccurate vehicle dynamics, actuator latency, wheel-ground interaction, sensor noise, environmental variability, or timing inconsistencies. Even small discrepancies can accumulate during closed-loop control tasks such as parking, where centimetre-level positioning accuracy and precise steering behaviour are required.

To address this issue, the concept of the digital twin has gained significant importance in autonomous systems research. A digital twin is a virtual representation of a physical system that remains closely synchronized with its real counterpart in terms of geometry, dynamics, sensing, and control interfaces. Unlike purely visual simulations, digital twins aim to reproduce behaviourally meaningful properties of the physical platform so that algorithms developed in simulation remain transferable to reality. In the context of autonomous driving, digital twins can serve as reliable environments for controller development, testing, validation, and sim-to-real transfer.

Existing digital-twin frameworks have demonstrated the potential of this approach for autonomous-vehicle validation and control. Nevertheless, many current implementations rely on expensive research platforms, custom-built hardware ecosystems, or highly specialized simulators that reduce accessibility and reproducibility. Furthermore, a significant portion of the literature focuses primarily on visual realism rather than on validating whether the simulated vehicle reproduces the actual dynamic behaviour of the physical platform under identical control inputs. In many cases, reinforcement-learning agents achieve high performance in simulation but fail to generalize to physical deployment because the learned policy exploits simulator-specific assumptions that do not exist in reality.

Scaled autonomous vehicles provide an attractive intermediate solution between purely simulated testing and full-scale vehicle experimentation. Small-scale platforms preserve essential characteristics of car-like motion and actuation while offering substantially lower cost, reduced safety risk, and easier experimental repeatability. Platforms such as F1TENTH and AutoDRIVE have demonstrated the effectiveness of scaled vehicles for autonomous-driving research and education. However, many of these systems either require dedicated hardware infrastructures or are tightly coupled to proprietary software ecosystems.

This thesis addresses these limitations through the development of a low-cost and fully reproducible sim-to-real framework based on a commercial 1:10-scale Traxxas Slash vehicle and a CARLA-based digital twin. The proposed system mirrors the physical platform inside the simulator while preserving a shared continuous action space and equivalent perception interfaces across both domains. The physical vehicle employs an NVIDIA Jetson Orin Nano for high-level autonomous control and an Arduino Due for low-level sensing and actuation. The digital twin reproduces the same steering constraints, throttle behaviour, sensor placement, and control-loop timing used on the real platform.

A key aspect of this work is that the digital twin is not assumed to be valid solely because it visually resembles the physical vehicle. Instead, the twin is quantitatively validated through direct comparison between simulation and reality using deterministic parking manoeuvres executed under identical control logic. By analysing yaw dynamics, lateral acceleration, and lateral offset using statistical similarity metrics, the thesis investigates where residual discrepancies appear and whether the simulated model reproduces the behaviour of the physical system with sufficient fidelity for control-policy transfer.

Once validated, the digital twin is used as a training environment for a PPO-based reinforcement-learning agent designed for autonomous reverse parking. The policy is trained entirely in simulation and deployed directly onto the real vehicle without additional fine-tuning, enabling evaluation of zero-shot sim-to-real transfer. This approach provides insight into whether signal-level fidelity in the digital twin translates into preserved behavioural performance at the task level.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



The motivation behind this work is therefore twofold. From a scientific perspective, the thesis seeks to investigate how accurately a low-cost CARLA-based digital twin can reproduce the behaviour of a physical autonomous vehicle and support reinforcement-learning transfer. From an engineering perspective, the work aims to provide a practical and reproducible framework that lowers the barrier for sim-to-real autonomous-driving research by combining commercially available hardware with open-source simulation tools.

Ultimately, improving sim-to-real transfer is essential for reducing development time, experimental cost, and deployment risk in autonomous systems. By bridging the gap between simulation and reality through validated digital twins, autonomous-driving research can move toward safer, more scalable, and more accessible methodologies for developing intelligent vehicle controllers.

### 1.2 Problem Statement

The development of autonomous-driving systems increasingly relies on simulation environments for controller design, testing, and validation. Simulation offers significant advantages in terms of safety, scalability, cost reduction, and experimental repeatability, particularly when machine-learning techniques such as Deep Reinforcement Learning (DRL) are employed. However, despite the rapid progress achieved in simulation-based autonomous driving, transferring learned behaviours from virtual environments to real vehicles remains a major unresolved challenge. This challenge is commonly referred to as the sim-to-real gap or reality gap.

The reality gap emerges because no simulator can perfectly reproduce all aspects of the physical world. Even advanced simulation frameworks such as CARLA simplify or approximate several phenomena involved in vehicle operation. Differences in tire-ground interaction, actuator latency, steering response, friction, sensor noise, vehicle mass distribution, environmental variability, and timing synchronization can lead to discrepancies between simulated and real-world behaviour. These discrepancies become especially problematic in closed-loop autonomous-driving tasks where the controller continuously reacts to sensor feedback and vehicle dynamics. Small modelling inaccuracies may accumulate over time and produce substantial trajectory deviations or task failures during real-world deployment.

This issue is particularly critical in automated parking applications. Parking manoeuvres are performed in constrained environments where centimetre-level accuracy is required to avoid collisions and achieve proper vehicle alignment. Unlike high-speed navigation tasks where minor positional errors may be tolerable, low-speed parking is highly sensitive to steering precision, short-range obstacle perception, and dynamic consistency between control commands and vehicle response. As a result, policies trained exclusively in simulation often fail to generalize effectively to physical platforms when deployed without adaptation.

Traditional automated parking systems typically address this problem through carefully engineered modular pipelines consisting of parking-slot detection, trajectory planning, and trajectory tracking. These approaches depend heavily on accurate mathematical models and manually tuned control parameters. Although they can achieve reliable performance under controlled assumptions, their effectiveness decreases when the environment becomes uncertain or when the vehicle behaviour deviates from the assumed model. Furthermore, designing and tuning such systems for every parking scenario can become increasingly complex and time-consuming.

Deep Reinforcement Learning offers an alternative paradigm by allowing autonomous agents to learn parking behaviour directly from interaction with the environment. DRL methods eliminate the need for manually designed trajectories and can potentially discover robust control strategies through experience. In simulation, DRL agents have demonstrated strong performance in various parking tasks,



## **Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle**



particularly when trained using algorithms such as Proximal Policy Optimization (PPO). Nevertheless, many existing studies remain limited to purely simulated evaluations and do not demonstrate successful deployment on physical vehicles. In several cases, reinforcement-learning policies rely on privileged observations such as bird's-eye-view images or perfect state information that are unavailable in realistic deployment scenarios. Consequently, the actual transferability of these approaches to real-world autonomous parking remains uncertain.

At the same time, research on digital twins has attempted to reduce the reality gap by constructing simulation models that closely replicate physical systems. While digital twins have shown promising results in robotics and autonomous driving, several important limitations remain in the existing literature. First, many digital-twin implementations prioritize visual realism rather than behavioural fidelity, meaning that the simulated vehicle may visually resemble the real platform while still failing to reproduce its actual dynamics under identical control inputs. Second, many proposed frameworks rely on expensive research-grade hardware or custom infrastructures that limit reproducibility and accessibility. Third, only a limited number of studies quantitatively validate the fidelity of the digital twin using direct signal-level comparisons between simulation and reality.

Another important limitation concerns the validation methodology itself. In many sim-to-real studies, the performance of the learned controller is used as indirect evidence of simulation fidelity. However, this approach makes it difficult to determine whether discrepancies originate from the controller design, the learning process, or the simulator itself. Without a deterministic and reproducible validation procedure, it becomes challenging to isolate and measure the actual simulation-reality mismatch. Therefore, there is a need for validation approaches that compare the simulated and physical systems under identical control logic while analysing measurable quantities such as vehicle orientation, acceleration, and trajectory evolution.

Additionally, although scaled autonomous-vehicle platforms have become increasingly popular for autonomous-driving research, many available solutions are based on specialized research platforms that require dedicated hardware ecosystems and extensive setup complexity. There is comparatively less research investigating whether low-cost, commercially available RC vehicles can serve as effective physical counterparts for validated CARLA-based digital twins while preserving the same control and sensing interfaces across simulation and reality.

The central problem addressed in this thesis can therefore be formulated as follows:

How can a low-cost CARLA-based digital twin of a commercial 1:10-scale autonomous vehicle be developed and validated such that reinforcement-learning policies trained entirely in simulation can be transferred effectively to the real platform with minimal performance degradation?

To address this problem, several subproblems must also be investigated:

1. How accurately can the simulated vehicle reproduce the physical platform dynamics during low-speed parking manoeuvres?
2. Which aspects of the vehicle behaviour contribute most significantly to the residual sim-to-real discrepancy?
3. Can a shared action and perception interface between simulation and hardware improve policy transferability?
4. Does signal-level agreement between simulation and reality translate into preserved task-level behaviour for a DRL parking controller?



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



5. To what extent do perception limitations, particularly ultrasonic sensing noise, affect the remaining performance gap after dynamic matching?

To answer these questions, this thesis proposes a complete sim-to-real framework based on a commercial 1:10-scale Traxxas Slash vehicle and a CARLA digital twin operating under a shared continuous action space. The work focuses specifically on autonomous reverse parking because it represents a highly sensitive and practically relevant low-speed manoeuvre where small modelling errors become immediately observable.

The proposed approach differs from many previous works in three key aspects. First, the physical and simulated platforms share the same control representation and equivalent perception interfaces, ensuring consistency between training and deployment conditions. Second, the digital twin is quantitatively validated using deterministic parking manoeuvres executed identically in both domains, allowing direct analysis of the remaining reality gap. Third, the validated twin is employed to train a PPO-based DRL policy that is deployed on the real platform without fine-tuning, enabling evaluation of true zero-shot sim-to-real transfer.

Ultimately, the problem addressed by this thesis lies at the intersection of autonomous driving, digital twins, reinforcement learning, and sim-to-real transfer. Solving this problem contributes toward the broader goal of developing autonomous systems that can be trained safely and efficiently in simulation while remaining reliable and transferable in real-world operation.

### 1.3 Research Objectives

The primary aim of this research is to develop and validate a high-fidelity digital twin of a scaled autonomous vehicle platform that enables effective sim-to-real transfer of reinforcement learning-based autonomous parking controllers. The study seeks to bridge the gap between simulation and reality by ensuring behavioural consistency between a virtual vehicle operating in the CARLA simulator and its physical counterpart.

To achieve this aim, the following research objectives are defined:

1. To design and develop a low-cost autonomous vehicle platform capable of performing autonomous parking manoeuvres using onboard sensing and embedded computation.
2. To create a CARLA-based digital twin that accurately reproduces the geometry, dynamics, sensor configuration, and timing characteristics of the physical vehicle.
3. To establish a quantitative validation framework for evaluating the fidelity of the digital twin through direct comparison of simulated and real-world vehicle behaviour.
4. To formulate and train a reinforcement learning controller for autonomous parking using the Proximal Policy Optimization (PPO) algorithm within the digital twin environment.
5. To investigate the effectiveness of zero-shot sim-to-real transfer by deploying a reinforcement learning policy trained entirely in simulation onto the physical vehicle without retraining or fine-tuning.
6. To identify the primary sources of discrepancy between simulation and reality and evaluate their impact on autonomous parking performance.

Through these objectives, the research aims to demonstrate that a validated digital twin can serve as a reliable platform for reinforcement learning development and significantly reduce the need for extensive real-world training.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 1.4 Research Questions

The research presented in this thesis is guided by the following research questions:

#### RQ1:

How accurately can a CARLA-based digital twin reproduce the dynamic behaviour of a physical scaled autonomous vehicle during autonomous parking manoeuvres?

#### RQ2:

Which vehicle-state characteristics provide meaningful indicators for evaluating digital twin fidelity between simulation and reality?

#### RQ3:

Can a reinforcement learning controller trained exclusively within the digital twin environment successfully perform autonomous parking tasks on the physical vehicle without additional adaptation?

#### RQ4:

What are the primary contributors to the remaining simulation-to-reality discrepancy after digital twin calibration and validation?

#### RQ5:

To what extent does maintaining consistency in sensing, vehicle dynamics, action spaces, and timing improve the transferability of reinforcement learning policies?

Answering these research questions provides insights into the practical feasibility of using digital twins as development and testing platforms for autonomous driving applications.

### 1.5 Contributions of the Thesis

This thesis makes several significant contributions to the fields of autonomous driving, digital twins, and sim-to-real reinforcement learning.

#### Contribution 1: Development of a Low-Cost Autonomous Vehicle Platform

A fully operational 1:10-scale autonomous vehicle platform was designed and implemented using commercially available hardware components. The platform integrates onboard computation, ultrasonic sensing, inertial measurement, and wireless communication while maintaining a low development cost and high reproducibility.

#### Contribution 2: Development of a CARLA-Based Digital Twin

A high-fidelity digital twin was developed within the CARLA simulator to replicate the geometry, steering behaviour, sensing architecture, and timing characteristics of the physical vehicle. The twin provides a realistic simulation environment suitable for controller development and testing.

#### Contribution 3: Quantitative Digital Twin Validation Framework

A signal-level validation methodology was proposed to evaluate digital twin fidelity using measurable vehicle-state variables. The framework employs statistical similarity metrics, including Pearson Correlation Coefficient (PCC) and Maximum Normalized Cross-Correlation (MNCC), to compare simulation and real-world behaviour.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### Contribution 4: Reinforcement Learning-Based Autonomous Parking Controller

A continuous-control autonomous parking controller based on the PPO algorithm was developed and trained entirely within the simulation environment. The controller learns parking behaviour through interaction with the digital twin without requiring manually designed trajectories.

### Contribution 5: Successful Zero-Shot Sim-to-Real Transfer

The trained reinforcement learning policy was successfully deployed onto the physical vehicle without retraining, fine-tuning, or domain adaptation. This demonstrates the effectiveness of the validated digital twin as a training environment for autonomous driving applications.

### Contribution 6: Analysis of the Remaining Reality Gap

The study identifies and analyses the major factors contributing to the remaining simulation-to-reality discrepancy, providing valuable insights for future digital twin and sim-to-real research.

Collectively, these contributions demonstrate a complete framework for developing, validating, and deploying reinforcement learning-based autonomous driving controllers using digital twin technology.

## 1.6 Thesis Structure

The remainder of this thesis is organized as follows:

**Chapter 2 – Literature Review** presents a comprehensive review of existing research related to autonomous parking systems, reinforcement learning, digital twins, simulation environments, and sim-to-real transfer methodologies. Research gaps addressed by this thesis are also identified.

**Chapter 3 – System Architecture** describes the overall architecture of the proposed system, including the physical vehicle platform, onboard hardware, sensing infrastructure, communication framework, and software components.

**Chapter 4 – CARLA Digital Twin Development** presents the design and implementation of the digital twin, including vehicle adaptation, sensor emulation, environment modelling, synchronization mechanisms, and simulation timing configuration.

**Chapter 5 – Digital Twin Validation Methodology** introduces the experimental methodology used to evaluate digital twin fidelity through deterministic parking experiments and quantitative signal comparison.

**Chapter 6 – Reinforcement Learning Controller** describes the formulation of the autonomous parking problem as a Markov Decision Process and presents the PPO-based learning framework, observation space, action space, reward design, and training process.

**Chapter 7 – Experimental Results and Discussion** presents the results obtained from digital twin validation and zero-shot sim-to-real deployment experiments. The chapter analyses controller performance, digital twin fidelity, and remaining sources of discrepancy.

**Chapter 8 – Conclusion and Future Work** summarizes the research findings, highlights the main contributions of the thesis, discusses limitations, and proposes directions for future research in digital twins, reinforcement learning, and autonomous driving systems.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### Chapter 2 – Literature Review

#### 2.1 Introduction

The rapid advancement of autonomous vehicle technologies has transformed intelligent transportation systems into one of the most active areas of research in robotics, artificial intelligence, computer vision, and control engineering. Autonomous vehicles promise significant improvements in road safety, traffic efficiency, energy consumption, and driving comfort by reducing human intervention and enabling intelligent decision-making. Recent developments in sensing technologies, high-performance embedded computing, machine learning, and wireless communication have accelerated the deployment of increasingly sophisticated autonomous driving systems capable of operating in complex urban environments. However, despite these technological advances, achieving reliable and safe autonomous operation under diverse real-world conditions remains a considerable challenge.

Among the various autonomous driving tasks, autonomous parking has received substantial attention from both academia and industry due to its practical importance and unique technical challenges. Parking manoeuvres require accurate perception of the surrounding environment, precise vehicle localization, low-speed trajectory planning, and smooth control within confined spaces. Unlike highway driving, where relatively large margins of error may be acceptable, autonomous parking demands centimeter-level positioning accuracy while simultaneously avoiding collisions with nearby obstacles, parked vehicles, and infrastructure. Consequently, autonomous parking serves as an excellent benchmark for evaluating perception, planning, and control algorithms because it combines geometric constraints, dynamic vehicle behaviour, and real-time decision-making within a highly constrained operating environment.

Conventional autonomous parking systems are generally implemented using modular architectures consisting of environment perception, parking-slot detection, trajectory generation, and trajectory tracking. These systems often rely on carefully designed mathematical models, geometric optimization techniques, and classical control algorithms such as Pure Pursuit, Stanley control, Model Predictive Control (MPC), or Proportional-Integral-Derivative (PID) control. While such approaches have demonstrated reliable performance in structured environments, they typically require extensive parameter tuning, accurate vehicle models, and significant engineering effort to maintain robustness under varying operating conditions. Furthermore, modular architectures often struggle to accommodate uncertainties arising from sensor noise, changing environmental conditions, or unmodeled vehicle dynamics.

Recent advances in machine learning, particularly Deep Reinforcement Learning (DRL), have introduced a fundamentally different paradigm for autonomous vehicle control. Instead of explicitly programming trajectories or handcrafted decision rules, reinforcement learning enables autonomous agents to learn optimal driving behaviours directly through interaction with an environment. By maximizing a cumulative reward function, an agent gradually discovers control policies capable of performing complex tasks without requiring manually designed trajectories. This learning-based approach has demonstrated promising results in numerous robotics applications, including manipulation, navigation, locomotion, and autonomous driving. For autonomous parking, reinforcement learning offers the potential to learn smooth and adaptive parking manoeuvres while naturally handling nonlinear vehicle behaviour and complex environmental interactions.

Despite these advantages, reinforcement learning introduces a significant practical challenge. Training deep reinforcement learning agents generally requires millions of interactions with the environment before satisfactory performance is achieved. Collecting such large quantities of experience using physical autonomous vehicles is often impractical due to the considerable time requirements, equipment



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



wear, operational costs, and safety risks associated with repeated experimentation. Consequently, simulation has become an indispensable component of reinforcement learning research for autonomous systems. Modern simulation environments allow researchers to accelerate training, repeat experiments under controlled conditions, and evaluate algorithms without exposing physical platforms to unnecessary risk.

Among the available simulation environments, CARLA has emerged as one of the most widely adopted open-source simulators for autonomous driving research [1]. Built on the Unreal Engine, CARLA provides realistic urban environments, configurable weather conditions, physics-based vehicle dynamics, and comprehensive sensor simulation capabilities [1]. Its flexible architecture and extensive Python application programming interface (API) make it particularly suitable for reinforcement learning research, autonomous vehicle development, and algorithm validation [1], [3]. Numerous studies have employed CARLA to investigate perception algorithms, path planning, control strategies, and reinforcement learning-based driving policies [3], [5], [6]. Nevertheless, controllers that achieve excellent performance within CARLA frequently experience performance degradation when deployed on real vehicles due to differences between simulated and physical environments [2], [9].

This discrepancy between simulation and reality, commonly referred to as the sim-to-real gap or reality gap, remains one of the principal obstacles to practical deployment of reinforcement learning in robotics and autonomous driving [9]. The reality gap originates from numerous factors, including inaccuracies in vehicle dynamics models, simplified environmental representations, imperfect sensor simulation, communication delays, actuator nonlinearities, and environmental uncertainties [2], [9]. . Even relatively small discrepancies between simulated and physical systems may accumulate over time, causing learned policies to behave unpredictably after deployment. Consequently, improving simulation fidelity has become an essential research direction for enabling reliable sim-to-real transfer of reinforcement learning policies.

One promising approach for reducing the reality gap is the development of digital twins [10], [11]. Unlike conventional simulation models that primarily focus on visual or functional representation, a digital twin is a continuously synchronized virtual counterpart of a physical system that accurately reproduces its geometry, behaviour, sensing characteristics, and operational dynamics [24], [25].

Although digital twins have gained widespread adoption in manufacturing, aerospace, and industrial automation, their application to reinforcement learning-based autonomous driving remains relatively limited [12], [13]. Existing studies frequently emphasize visual realism or task-level performance while providing limited quantitative evaluation of digital twin fidelity [12]. In many cases, reinforcement learning agents are trained using privileged simulator information that is unavailable during real-world operation [3], [5], [16].

As a result, policies developed under such assumptions often exhibit poor transferability when deployed outside the simulation environment. Furthermore, relatively few studies investigate low-cost scaled autonomous vehicle platforms that provide an accessible and reproducible alternative to full-sized autonomous vehicles for sim-to-real research.

To address these limitations, recent research has increasingly emphasized the importance of validating digital twins through quantitative comparisons between simulated and physical systems. Rather than relying solely on successful task completion as evidence of simulation fidelity, signal-level validation compares measurable vehicle responses such as yaw angle, lateral acceleration, steering behaviour, and trajectory evolution between simulation and reality. Statistical similarity metrics provide objective measures of agreement and help identify the remaining sources of discrepancy. Such validation



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



methodologies establish confidence that controllers trained within the digital twin experience vehicle dynamics representative of the physical platform, thereby improving the likelihood of successful sim-to-real transfer.

The present thesis is situated within this emerging research area by investigating the development of a high-fidelity CARLA-based digital twin for a 1:10-scale autonomous vehicle platform designed for reinforcement learning-based autonomous parking. The digital twin reproduces the geometry, steering characteristics, sensor placement, and dynamic behaviour of the physical platform while maintaining equivalent perception and control interfaces across both simulation and reality. A reinforcement learning controller based on the Proximal Policy Optimization (PPO) algorithm is trained entirely within the digital twin using only onboard-equivalent sensor observations. The learned policy is subsequently deployed directly onto the physical vehicle through a zero-shot sim-to-real transfer process without retraining or domain adaptation. Prior to deployment, the fidelity of the digital twin is quantitatively evaluated using deterministic parking experiments and statistical comparisons between simulated and real-world vehicle behaviour.

The remainder of this chapter reviews the existing body of knowledge relevant to this research. Section 2.2 presents an overview of autonomous vehicle systems and their fundamental architectural components. Section 2.3 reviews existing autonomous parking methodologies, including both conventional model-based approaches and recent learning-based techniques. Section 2.4 discusses the evolution of digital twin technology and its applications within autonomous driving and robotics. Section 2.5 examines major simulation platforms used for autonomous vehicle research, with particular emphasis on the CARLA simulator. Section 2.6 reviews the principles of deep reinforcement learning and compares commonly used continuous-control algorithms. Section 2.7 analyzes existing sim-to-real transfer methodologies and discusses the challenges associated with reducing the reality gap. Section 2.8 critically reviews recent literature on digital twin-based autonomous parking and reinforcement learning, highlighting the strengths and limitations of existing approaches. Finally, Section 2.9 identifies the research gaps that motivate the proposed methodology and establishes the contributions of this thesis within the broader context of autonomous driving research.

### 2.2 Autonomous Vehicle Systems

Autonomous vehicles (AVs), also referred to as self-driving or driverless vehicles, represent one of the most significant technological advancements in modern intelligent transportation systems. By integrating artificial intelligence, robotics, computer vision, embedded systems, and advanced control techniques, autonomous vehicles are capable of perceiving their surroundings, making driving decisions, and executing control actions with little or no human intervention. Over the past two decades, substantial progress in sensing technologies, high-performance computing, machine learning algorithms, and communication systems has accelerated the development of autonomous driving systems, transforming what was once considered a futuristic concept into a rapidly evolving reality.

The motivation behind autonomous vehicle development extends beyond technological innovation. Factors such as distracted driving, fatigue, impaired judgment, and delayed reaction times significantly contribute to road fatalities and economic losses. Autonomous driving technologies seek to address these challenges by replacing or assisting human decision-making with intelligent systems capable of continuously monitoring the surrounding environment and reacting more quickly and consistently than human drivers. In addition to improving road safety, autonomous vehicles are expected to enhance traffic efficiency, reduce fuel consumption, decrease emissions, improve mobility for elderly and disabled individuals, and enable new transportation services such as autonomous ride-sharing and last-mile delivery.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Despite these potential benefits, achieving full vehicle autonomy remains a highly complex engineering challenge. Unlike conventional automated systems that operate under controlled conditions, autonomous vehicles must function reliably within dynamic and unpredictable environments containing pedestrians, cyclists, moving vehicles, varying road conditions, and changing weather. Consequently, modern autonomous driving systems require the seamless integration of multiple sensing modalities, robust perception algorithms, intelligent planning strategies, and reliable control mechanisms capable of making real-time decisions under uncertainty.

### 2.2.1 Levels of Vehicle Automation

To standardize the classification of driving automation, the Society of Automotive Engineers (SAE) introduced the SAE J3016 standard, which defines six levels of driving automation ranging from Level 0 (no automation) to Level 5 (full automation) [17]. This framework has become the most widely accepted classification system within both academia and industry [17].

At **Level 0**, the human driver performs all aspects of the driving task, while assistance systems such as warning indicators may provide limited support.

**Level 1** introduces driver assistance features including adaptive cruise control or lane-keeping assistance, where automation supports either steering or longitudinal control but not both simultaneously.

**Level 2**, commonly referred to as partial automation, enables simultaneous steering and speed control under driver supervision. Commercial systems such as Tesla Autopilot, Mercedes-Benz Drive Pilot, and General Motors Super Cruise operate primarily within this level under specific operating conditions.

**Level 3**, known as conditional automation, allows the vehicle to perform all dynamic driving tasks under predefined conditions while expecting the human driver to intervene when requested.

**Level 4** extends this capability by enabling the vehicle to operate autonomously within a restricted Operational Design Domain (ODD), such as urban shuttle routes or designated geographic areas, without requiring immediate human intervention.

**Level 5** represents complete driving automation under all environmental and roadway conditions without the need for a human driver or manual controls. Although significant progress has been achieved toward higher levels of autonomy, fully autonomous Level 5 vehicles remain an active research challenge due to the enormous variability and unpredictability of real-world driving environments. Consequently, much of today's research focuses on solving specific autonomous driving tasks (including lane following, obstacle avoidance, and autonomous parking) that serve as essential building blocks toward fully autonomous transportation systems.

### 2.2.2 Architecture of Autonomous Vehicle Systems

Although implementations differ among manufacturers and research institutions, most autonomous vehicle systems follow a modular architecture consisting of four primary functional layers: perception, localization, planning, and control. These modules operate sequentially while continuously exchanging information to generate safe and reliable vehicle behaviour.

#### Perception Layer

The perception layer serves as the vehicle's interface with the external environment. Its primary responsibility is to collect raw data from onboard sensors and transform these measurements into a meaningful representation of the surrounding world. Accurate perception enables the autonomous



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



vehicle to detect obstacles, identify road boundaries, recognize traffic signs and signals, estimate free driving space, and track dynamic objects.

Modern autonomous vehicles typically employ multiple complementary sensing technologies to improve robustness and compensate for the limitations of individual sensors. Cameras provide high-resolution visual information useful for object recognition, lane detection, and traffic sign classification. LiDAR sensors generate dense three-dimensional point clouds that facilitate accurate distance estimation and environmental mapping. Radar systems offer reliable object detection under adverse weather conditions while providing direct velocity measurements. Ultrasonic sensors are commonly employed for short-range obstacle detection during low-speed manoeuvres such as parking. In addition, Global Navigation Satellite Systems (GNSS), Inertial Measurement Units (IMUs), and wheel encoders contribute to vehicle localization and state estimation.

Sensor fusion techniques combine information from multiple sensing modalities to generate a comprehensive understanding of the environment. Probabilistic approaches such as Kalman Filters, Extended Kalman Filters, Particle Filters, and modern deep learning-based fusion methods are frequently employed to improve perception accuracy while mitigating the effects of sensor noise and uncertainty.

### Localization Layer

Accurate localization is fundamental to autonomous driving because virtually all planning and control decisions depend upon knowledge of the vehicle's position and orientation. Localization estimates the vehicle's pose within either a global reference frame or a local coordinate system by integrating information from GPS receivers, inertial sensors, wheel odometry, cameras, and LiDAR measurements.

Several localization approaches have been proposed within the literature, including Simultaneous Localization and Mapping (SLAM), visual odometry, LiDAR-based localization, and map-matching techniques. While centimeter-level localization is often achievable under favorable conditions, performance may deteriorate in environments affected by poor GPS reception, dynamic obstacles, or adverse weather. Consequently, robust localization remains an active area of autonomous driving research.

For autonomous parking applications, localization requirements become even more stringent because relatively small positioning errors may prevent successful parking maneuvers. Nevertheless, many reinforcement learning approaches intentionally avoid reliance on highly accurate localization systems by utilizing only onboard sensor observations that are also available during real-world deployment. Such perception-driven approaches improve transferability between simulation and physical platforms.

### Planning Layer

The planning layer determines the sequence of actions required to move the vehicle safely toward its intended destination while satisfying traffic regulations, vehicle dynamics, and environmental constraints. Planning is commonly divided into three hierarchical levels.

The first level, known as **global planning**, computes a high-level route between the origin and destination using digital road maps and graph-search algorithms such as Dijkstra or A\*. The second level, **behaviour planning**, determines tactical driving decisions including lane changes, overtaking, intersection negotiation, and parking strategies. Finally, **local planning** generates dynamically feasible trajectories that avoid nearby obstacles while respecting vehicle kinematic and dynamic constraints.

Traditional planning algorithms frequently employ optimization methods, spline interpolation, clothoid curves, or graph-search techniques. More recently, machine learning and reinforcement learning have



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



been investigated as alternatives capable of directly learning driving policies through environmental interaction without explicitly constructing trajectories.

### Control Layer

The control layer converts planned trajectories or learned driving policies into executable steering, throttle, and braking commands. Its objective is to ensure that the vehicle accurately follows the desired trajectory while maintaining stability, passenger comfort, and safety.

Classical autonomous driving controllers include Proportional-Integral-Derivative (PID) control, Pure Pursuit, Stanley control, Linear Quadratic Regulators (LQR), and Model Predictive Control (MPC). These controllers rely on mathematical models describing vehicle behaviour and often require careful parameter tuning to achieve satisfactory performance.

Reinforcement learning introduces an alternative control paradigm by allowing the vehicle to learn optimal control policies directly from experience. Instead of explicitly computing trajectories, reinforcement learning algorithms generate control commands based on observations received from the environment, making them particularly attractive for highly nonlinear tasks such as autonomous parking. However, their successful deployment depends heavily on the availability of realistic training environments and effective sim-to-real transfer strategies.

### 2.2.3 Sensor Technologies in Autonomous Vehicles

The performance of an autonomous vehicle is strongly influenced by the capabilities of its sensing system. Each sensing technology possesses distinct advantages and limitations, motivating the use of heterogeneous sensor suites in practical implementations.

Camera systems provide rich semantic information and remain the primary sensor for traffic sign recognition, lane detection, and object classification. However, their performance is highly sensitive to lighting variations, shadows, fog, and adverse weather conditions.

LiDAR sensors generate highly accurate three-dimensional representations of the environment, enabling precise object localization and mapping. Despite their accuracy, LiDAR systems remain relatively expensive and computationally demanding, limiting their adoption in low-cost research platforms.

Radar sensors offer robust operation under challenging weather conditions and provide reliable range and relative velocity measurements. Nevertheless, their relatively low angular resolution reduces their effectiveness for detailed scene understanding.

Ultrasonic sensors are inexpensive, lightweight, and particularly suitable for detecting nearby obstacles during low-speed maneuvers. Although their measurement range is limited and angular resolution is relatively poor, ultrasonic sensors are widely employed in commercial parking assistance systems and scaled autonomous vehicle platforms due to their simplicity and cost-effectiveness. Consequently, they represent an appropriate sensing modality for reinforcement learning-based autonomous parking systems that seek to maintain consistency between simulation and physical deployment.

### 2.2.4 Challenges in Autonomous Vehicle Development

Despite remarkable technological progress, several challenges continue to limit the widespread deployment of autonomous vehicles.

One of the most significant challenges involves ensuring robust perception under varying environmental conditions. Sensor measurements are inherently affected by noise, occlusions, changing illumination,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



precipitation, and dynamic obstacles. Developing perception systems capable of maintaining reliable performance across diverse operating conditions remains an active area of research.

Another major challenge concerns computational requirements. Modern autonomous driving algorithms frequently rely on deep neural networks containing millions of trainable parameters. Real-time execution of these models requires powerful embedded hardware while simultaneously satisfying strict energy and thermal constraints.

Vehicle modeling also presents considerable difficulties. Tire-road interaction, actuator dynamics, suspension behaviour, and environmental disturbances introduce nonlinear effects that are difficult to model accurately. Consequently, controllers developed using simplified models often experience degraded performance during real-world deployment.

Perhaps the most critical challenge addressed in recent literature is the **simulation-to-reality (sim-to-real) gap**. While simulation provides a safe and efficient environment for algorithm development, discrepancies between simulated and physical systems frequently prevent direct deployment of learned controllers. These discrepancies originate from imperfect dynamics models, simplified sensor simulations, communication latency, actuator nonlinearities, and environmental uncertainty. Reducing this reality gap has therefore become a central objective of modern autonomous driving research, motivating the adoption of digital twin technologies and quantitative validation methodologies.

### 2.2.5 Autonomous Parking as a Representative Autonomous Driving Task

Among the numerous autonomous driving functions, autonomous parking occupies a unique position because it simultaneously requires accurate perception, precise vehicle control, intelligent decision-making, and reliable obstacle avoidance within highly constrained environments. Successful parking demands centimeter-level positioning accuracy while maintaining smooth vehicle motion and avoiding collisions throughout the maneuver.

These characteristics make autonomous parking an ideal benchmark for evaluating reinforcement learning algorithms and digital twin fidelity. Unlike highway driving, parking emphasizes low-speed vehicle dynamics where steering accuracy, sensor reliability, and geometric consistency dominate system performance. Consequently, improvements achieved in autonomous parking frequently translate to broader advances in autonomous vehicle control, simulation fidelity, and sim-to-real transfer methodologies.

The research presented in this thesis focuses specifically on autonomous reverse parking using a low-cost scaled autonomous vehicle platform and a high-fidelity CARLA-based digital twin. By combining realistic simulation, reinforcement learning, and quantitative validation, the proposed framework seeks to address one of the fundamental challenges in autonomous driving research: the reliable transfer of learned control policies from simulation to real-world operation.

### 2.3 Autonomous Parking Systems

Autonomous parking represents one of the earliest and most commercially successful applications of autonomous driving technology. Compared with highway driving or urban navigation, parking manoeuvres are performed at relatively low speeds; however, they require considerably higher positional accuracy, precise steering control, and continuous obstacle monitoring within confined spaces. Even small positioning errors may result in collisions with surrounding vehicles or infrastructure, making autonomous parking a challenging control problem despite its relatively low operating speed.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



From a research perspective, autonomous parking serves as an ideal benchmark for evaluating perception systems, motion planning algorithms, vehicle controllers, and decision-making strategies. The parking task combines nonlinear vehicle kinematics, geometric constraints, limited maneuvering space, and dynamic environmental interactions, requiring autonomous systems to perform accurate sensing and reliable control while satisfying strict safety requirements. Consequently, autonomous parking has become an active research area within robotics, intelligent transportation systems, and artificial intelligence.

Over the past three decades, autonomous parking methodologies have evolved from purely geometric and rule-based algorithms toward optimization-based planning techniques and, more recently, learning-based approaches driven by deep reinforcement learning. Each generation has addressed specific limitations of previous methods while introducing new challenges regarding computational complexity, robustness, and deployment in real-world environments.

### 2.3.1 Classical Autonomous Parking Approaches

Early autonomous parking systems were primarily based on geometric reasoning and analytical vehicle models. These methods assume that the vehicle follows well-defined kinematic constraints and generate collision-free trajectories using mathematical formulations. Their popularity stems from their interpretability, computational efficiency, and predictable behaviour under structured operating conditions.

One of the simplest approaches involves predefined parking trajectories constructed using circular arcs, spline curves, or polynomial functions [14]. Such methods determine steering commands directly from the geometry of the parking space and vehicle dimensions.

Because the resulting trajectories are computed analytically, these approaches require relatively little computational power and can be implemented on embedded automotive hardware. However, they generally assume ideal environmental conditions and are highly sensitive to localization errors and changes in vehicle dynamics.

Graph-search algorithms represent another widely adopted class of autonomous parking planners. Methods such as A\*, Dijkstra, Hybrid A\*, and Reeds–Shepp path planning compute feasible collision-free paths while explicitly considering the vehicle's kinematic constraints. Hybrid A\*, in particular, has become a standard solution for autonomous parking because it combines graph search with continuous vehicle motion primitives, producing smooth trajectories suitable for real vehicle operation. Nevertheless, these algorithms often require high-quality occupancy maps and can become computationally demanding in complex environments with numerous obstacles.

Optimization-based trajectory generation has also attracted significant attention. Techniques based on Model Predictive Control (MPC), Sequential Quadratic Programming (SQP), and nonlinear optimization formulate parking as a constrained optimization problem in which steering and velocity commands are optimized to minimize tracking errors while satisfying vehicle dynamics and collision constraints. Model Predictive Control has demonstrated excellent performance because it explicitly incorporates vehicle dynamics and actuator limitations while continuously updating control commands during execution. However, MPC requires accurate mathematical models of vehicle behaviour and often demands significant computational resources, limiting its applicability to low-cost embedded platforms.

Although classical planning methods remain widely employed in commercial parking assistance systems, they exhibit several important limitations. Their performance depends heavily on accurate



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



vehicle models, carefully tuned parameters, and reliable localization. Moreover, these methods often struggle to generalize across diverse parking scenarios because their decision-making processes are explicitly programmed rather than learned from experience.

### 2.3.2 Learning-Based Autonomous Parking

The rapid development of machine learning has introduced alternative approaches capable of learning parking behaviour directly from data. Instead of relying on handcrafted trajectory generation and manually designed control rules, learning-based systems automatically infer driving strategies through optimization or interaction with the environment.

Supervised learning approaches represent one of the earliest applications of artificial intelligence in autonomous parking [8]. In these systems, neural networks are trained using datasets collected from expert human drivers or high-performance controllers [8]. The trained model learns to predict steering angles or control commands based on sensor observations. While supervised learning simplifies controller development, its performance is fundamentally limited by the quality and diversity of the training dataset. Furthermore, collecting sufficiently representative driving data for every possible parking scenario is expensive and time-consuming.

Imitation learning extends supervised learning by attempting to replicate expert driving behaviour more efficiently. Behavioural Cloning directly maps observations to actions using demonstration data, whereas inverse reinforcement learning attempts to recover the underlying reward function governing expert behaviour. Although imitation learning can achieve high-quality parking performance under conditions similar to the training data, it often suffers from distribution shift when encountering unfamiliar environments or unexpected obstacles.

These limitations have motivated increasing interest in reinforcement learning as an alternative framework for autonomous parking.

### 2.3.3 Reinforcement Learning for Autonomous Parking

Reinforcement Learning (RL) formulates autonomous parking as a sequential decision-making problem in which an intelligent agent interacts with an environment to maximize cumulative reward. Unlike supervised learning, reinforcement learning does not require labelled datasets or manually generated trajectories. Instead, the agent gradually improves its control policy through repeated trial-and-error interactions.

The autonomous parking problem is naturally modelled as a Markov Decision Process (MDP). At each time step, the agent observes the current vehicle state and surrounding environment, selects an action consisting of steering, throttle, and braking commands, and receives a reward based on the consequences of its action. Positive rewards encourage successful parking behaviour, while penalties discourage collisions, excessive steering, unnecessary manoeuvring, and deviations from the desired parking position.

One of the primary advantages of reinforcement learning is its ability to learn complex nonlinear control policies without requiring explicit mathematical models of vehicle dynamics. This characteristic makes RL particularly attractive for autonomous parking, where vehicle behaviour may vary due to tire characteristics, actuator nonlinearities, surface conditions, and sensor uncertainty. Furthermore, reinforcement learning naturally accommodates continuous adaptation and can discover driving strategies that may be difficult to derive analytically.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Several reinforcement learning algorithms have been investigated for autonomous driving applications. Early studies employed Q-learning and Deep Q-Networks (DQN) for discrete control tasks [18]; however, parking requires continuous steering and speed control that cannot be easily represented using discrete action spaces. Consequently, continuous-control algorithms have become the preferred choice for autonomous parking research.

Deep Deterministic Policy Gradient (DDPG) introduced deterministic policy learning for continuous actions, enabling more realistic vehicle control. Twin Delayed Deep Deterministic Policy Gradient (TD3) subsequently improved training stability by reducing overestimation bias, while Soft Actor-Critic (SAC) incorporated entropy maximization to improve exploration and policy robustness.

Among these algorithms, Proximal Policy Optimization (PPO) has emerged as one of the most widely adopted methods due to its balance between training stability, implementation simplicity, and computational efficiency [4]. PPO employs clipped policy updates that prevent excessively large parameter changes during optimization, resulting in smoother convergence and improved robustness across a wide range of robotics applications [4].

Recent studies have demonstrated the effectiveness of PPO for autonomous parking tasks using both simulated and physical vehicle platforms [3], [5], [6], [16]. Nevertheless, many existing implementations rely on privileged simulator information such as precise global coordinates, occupancy grids, or perfect state estimation that are unavailable during real-world deployment [5], [16]. This dependence on unrealistic observations significantly limits the transferability of learned policies outside the simulation environment.

### 2.3.4 Perception for Autonomous Parking

Reliable perception is fundamental to autonomous parking because all control decisions depend upon accurate information regarding nearby obstacles and vehicle position. Commercial parking systems commonly integrate multiple sensing modalities, including ultrasonic sensors, cameras, radar, and LiDAR, to improve environmental awareness.

Ultrasonic sensors remain the most widely deployed sensing technology for parking assistance due to their low cost, simplicity, and effectiveness in short-range obstacle detection [15]. These sensors measure the distance to nearby objects by transmitting high-frequency acoustic pulses and estimating the time required for the reflected signal to return [15].

Although ultrasonic sensors provide relatively limited angular resolution and operating range, they are particularly well suited for low-speed parking applications where precise short-distance measurements are more important than long-range environmental perception.

Vision-based parking systems employ monocular or stereo cameras to detect parking spaces, estimate free driving space, and recognize surrounding objects. Deep learning techniques have significantly improved camera-based perception by enabling robust object detection and semantic scene understanding. However, camera performance remains highly dependent on illumination conditions, shadows, and weather.

LiDAR-based perception offers highly accurate three-dimensional environmental reconstruction but typically requires expensive hardware and substantial computational resources [14]. Consequently, LiDAR is more commonly employed in full-scale autonomous vehicles than in low-cost research platforms.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Scaled autonomous vehicle platforms frequently rely on simplified sensing configurations consisting of ultrasonic sensors and inertial measurements. While these systems provide less environmental information than full-scale sensor suites, they more closely resemble practical low-cost autonomous systems and simplify sim-to-real transfer by maintaining equivalent sensing interfaces between simulation and physical deployment.

### 2.3.5 Challenges in Reinforcement Learning-Based Parking

Despite considerable progress, reinforcement learning-based autonomous parking remains associated with several practical challenges.

The first challenge concerns **sample efficiency**. Deep reinforcement learning algorithms typically require millions of interactions before converging to satisfactory control policies. Training directly on physical vehicles is therefore impractical due to equipment wear, battery consumption, operational cost, and safety considerations.

A second challenge involves **reward function design**. Poorly designed reward functions may encourage undesirable behaviours such as oscillatory steering, excessive reversing, or collision-prone trajectories. Developing reward structures that simultaneously promote safety, efficiency, and smooth vehicle motion remains an active research topic.

Another important issue is **generalization**. Policies trained within limited environments frequently fail when encountering previously unseen parking geometries, obstacle configurations, or environmental conditions. Domain randomization and curriculum learning have been proposed to improve robustness, although complete generalization remains difficult.

Perhaps the most significant challenge is the **simulation-to-reality gap**. Reinforcement learning agents trained in simulation often exploit characteristics specific to the virtual environment that are absent in physical systems. Differences in vehicle dynamics, sensor noise, communication latency, actuator response, and environmental variability may substantially degrade performance following deployment. Consequently, the effectiveness of reinforcement learning-based parking depends strongly upon the fidelity of the simulation environment and the consistency of perception and control interfaces across simulation and reality.

### 2.3.6 Research Trends and Remaining Challenges

The literature demonstrates a clear transition from model-based autonomous parking toward data-driven and learning-based methodologies. Classical approaches continue to dominate commercial parking systems because of their reliability and interpretability, whereas reinforcement learning offers greater adaptability and reduced dependence on explicit vehicle models. Nevertheless, most reinforcement learning studies remain confined to simulation environments and provide limited evidence regarding successful deployment on physical platforms.

Furthermore, existing research frequently evaluates parking success using only task-completion rates while neglecting quantitative validation of the underlying simulation model. Few studies perform detailed comparisons between simulated and physical vehicle behaviour using measurable dynamic variables such as yaw angle, lateral acceleration, or trajectory deviation. As a result, it is often difficult to determine whether successful deployment results from robust learning or merely favourable experimental conditions.

These limitations motivate the development of high-fidelity digital twins capable of accurately reproducing vehicle behaviour and supporting reliable zero-shot sim-to-real transfer. By combining quantitative digital twin validation with reinforcement learning, researchers can significantly improve



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



confidence that policies trained in simulation will exhibit comparable performance when deployed on real autonomous vehicles.

### 2.4 Digital Twin Technology

Digital Twin (DT) technology has emerged as one of the most influential concepts in the fields of Industry 4.0, cyber-physical systems, robotics, and intelligent transportation systems. By creating a virtual representation that accurately reflects the characteristics and behaviour of a physical asset, digital twins enable engineers and researchers to monitor, analyse, optimize, and predict the performance of real-world systems in a safe and cost-effective manner. Unlike conventional computer simulations, which often represent static models or isolated scenarios, digital twins maintain a close relationship with their physical counterparts throughout the system lifecycle, allowing continuous synchronization between the physical and virtual domains.

Originally developed for industrial applications, digital twins have rapidly expanded into numerous engineering disciplines, including aerospace, healthcare, smart manufacturing, energy systems, construction, and autonomous vehicles. Their ability to reduce development costs, accelerate testing, improve predictive maintenance, and support virtual experimentation has made them an essential component of modern engineering design. More recently, digital twins have gained increasing attention within robotics and autonomous driving, where they provide high-fidelity simulation environments capable of supporting algorithm development, controller validation, and sim-to-real reinforcement learning.

This section reviews the evolution of digital twin technology, examines its fundamental architecture, discusses its applications within autonomous vehicle research, and analyses current challenges associated with developing high-fidelity digital twins for reinforcement learning.

#### 2.4.1 Evolution of Digital Twin Technology

Although the term *Digital Twin* became widely recognized only during the last decade, the underlying concept originated much earlier. During the Apollo space missions in the 1960s, NASA developed highly accurate ground-based replicas of spacecraft to reproduce system behaviour and evaluate potential solutions for unexpected failures. These physical replicas functioned as early forms of digital twin methodology by enabling engineers to test corrective actions before applying them to operational spacecraft.

The formal concept of the digital twin was introduced by Michael Grieves in 2002 within the context of Product Lifecycle Management (PLM) [24]. Grieves proposed a framework in which every physical product is associated with a corresponding virtual representation that evolves throughout the product's lifecycle [24]. This virtual model continuously exchanges information with its physical counterpart, allowing engineers to monitor system behaviour, predict failures, evaluate design modifications, and optimize operational performance [24]. It was in later work co-authored with NASA's John Vickers that the term "Digital Twin" itself was formally attached to this concept [25].

The widespread adoption of Industry 4.0 further accelerated digital twin research by integrating advanced sensing technologies, Internet of Things (IoT) devices, cloud computing, artificial intelligence, and big data analytics. These technologies enabled continuous data exchange between physical and virtual systems, transforming digital twins from static engineering models into dynamic cyber-physical systems capable of real-time monitoring and decision support.

Today, digital twins are recognized as one of the enabling technologies of smart manufacturing and intelligent systems. Their application has expanded well beyond industrial production to include smart



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



cities, healthcare systems, energy management, transportation networks, and autonomous robotic platforms.

### 2.4.2 Definition and Characteristics of Digital Twins

Despite their widespread adoption, no universally accepted definition of a digital twin exists [25]. However, most definitions emphasize three fundamental components: a physical entity, its virtual representation, and a communication mechanism enabling information exchange between the two domains [25].

Unlike traditional simulations, digital twins are characterized by continuous correspondence with their physical counterparts. They reproduce not only the geometry of the physical system but also its dynamic behaviour, sensing characteristics, operational state, and environmental interactions. As the physical system evolves, the digital twin is updated accordingly, ensuring that both representations remain consistent throughout the system lifecycle.

Several characteristics distinguish digital twins from conventional simulation models:

- **Behavioural fidelity**, which ensures that the virtual model reproduces the dynamic response of the physical system rather than merely its visual appearance.
- **Bidirectional information exchange**, allowing information to flow between physical and virtual environments for monitoring, prediction, and optimization.
- **Lifecycle integration**, enabling continuous system analysis from design and manufacturing through deployment, maintenance, and retirement.
- **Predictive capability**, whereby the virtual model can evaluate future operating conditions or potential failures before they occur in reality.

These characteristics make digital twins particularly attractive for autonomous vehicle development, where controller performance depends strongly upon the realism of the simulated environment.

For the purposes of this thesis, a **digital twin** is defined as a simulation-based virtual representation of a physical system that (i) is calibrated against measured data from its physical counterpart to reproduce that system's dynamic and sensing behaviour with quantitatively validated fidelity, and (ii) is used as a substitute for the physical system during a specific downstream task, in this case, training and evaluating a reinforcement learning controller. This definition emphasizes **calibration and validated behavioural fidelity** as the defining requirement, rather than continuous real-time synchronization between the physical and virtual domains, which is not present in the system developed here (see 4.9/3.1 for the specific scope adopted).

### 2.4.3 Digital Twin Architecture

Although digital twin implementations vary considerably depending on the application, most architectures follow a layered cyber-physical framework consisting of four primary components: the physical system, the data acquisition layer, the virtual model, and the application layer.

The **physical layer** contains the real-world asset, including its mechanical structure, sensors, actuators, and embedded computing hardware. In autonomous vehicle applications, this layer consists of the vehicle platform together with its perception and control systems.

The **data acquisition layer** collects information describing the operational state of the physical system. Sensors continuously measure variables such as position, velocity, acceleration, steering angle, battery



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



voltage, obstacle distance, and environmental conditions. Communication technologies transfer these measurements to the virtual environment for analysis and synchronization.

The **virtual layer** represents the core of the digital twin. It contains mathematical models, physics simulations, environmental representations, and virtual sensors that collectively reproduce the behaviour of the physical system. High-fidelity digital twins emphasize behavioural consistency by accurately replicating system dynamics, actuator response, sensing characteristics, and timing behaviour.

Finally, the **application layer** utilizes information generated by the digital twin for monitoring, optimization, predictive maintenance, controller development, fault diagnosis, and decision support.

The effectiveness of a digital twin depends largely on the accuracy with which information flows between these layers. Poor synchronization or simplified modeling may significantly reduce simulation fidelity and limit the practical value of the digital twin.

### 2.4.4 Digital Twins in Robotics

Robotics has become one of the fastest-growing application domains for digital twin technology. Developing robotic systems directly on physical hardware is often expensive, time-consuming, and potentially hazardous, particularly when machine learning algorithms require extensive experimentation. Digital twins provide an attractive alternative by allowing researchers to perform algorithm development, controller tuning, and system validation within virtual environments before deployment.

Industrial robotics has benefited extensively from digital twins for production planning, robot programming, collision detection, and predictive maintenance. Virtual commissioning enables manufacturing engineers to validate robotic workcells before physical installation, significantly reducing commissioning time and operational cost.

Mobile robotics has similarly adopted digital twins for navigation, mapping, multi-robot coordination, and reinforcement learning. Digital twins allow researchers to evaluate navigation algorithms under controlled conditions while avoiding damage to expensive robotic hardware.

More recently, reinforcement learning has further increased the importance of robotic digital twins because learning algorithms require millions of interactions that cannot realistically be collected using physical robots alone.

### 2.4.5 Digital Twins for Autonomous Vehicles

Autonomous vehicles represent one of the most demanding applications of digital twin technology due to the complexity of their sensing, planning, and control systems. Modern autonomous vehicles operate within dynamic environments characterized by moving objects, uncertain weather conditions, varying road surfaces, and complex traffic interactions. Developing and validating autonomous driving algorithms directly on physical vehicles is therefore both expensive and potentially unsafe.

Digital twins provide an effective solution by enabling virtual experimentation under highly controlled conditions while maintaining close correspondence with the physical vehicle. Researchers can investigate perception algorithms, path planning strategies, obstacle avoidance techniques, and reinforcement learning controllers before deployment onto real platforms.

Unlike conventional vehicle simulations, digital twins emphasize consistency between simulation and physical operation. Vehicle geometry, wheelbase, steering constraints, tire behaviour, actuator response,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



sensor placement, and communication timing must closely match the physical platform to minimize discrepancies during deployment.

Several recent studies have investigated digital twins for vehicle diagnostics, fleet management, predictive maintenance, traffic optimization, and autonomous driving. However, many existing implementations primarily emphasize visual realism while providing limited quantitative validation of dynamic vehicle behaviour. Consequently, successful simulation performance does not necessarily guarantee successful deployment on physical vehicles.

### 2.4.6 Digital Twins for Reinforcement Learning

The rapid growth of deep reinforcement learning has significantly expanded interest in digital twins because reinforcement learning algorithms require large quantities of interaction data during training. Simulation environments provide an efficient means of generating this data without exposing physical systems to unnecessary wear or safety risks.

Within reinforcement learning, digital twins perform two critical functions. First, they provide realistic environments in which agents can explore complex behaviours through repeated interaction. Second, they reduce the simulation-to-reality gap by accurately reproducing the sensing, dynamics, and control interfaces of physical systems.

An effective reinforcement learning digital twin should satisfy several important requirements:

- Accurate vehicle dynamics.
- Realistic sensor behaviour.
- Consistent action and observation spaces.
- Stable timing characteristics.
- Environmental realism sufficient for policy generalization.

Failure to satisfy these requirements frequently results in learned policies exploiting unrealistic characteristics of the simulator, thereby reducing transferability to physical systems.

### 2.4.7 Challenges in Digital Twin Development

Despite considerable progress, developing high-fidelity digital twins remains a challenging research problem.

One major challenge involves accurately modelling nonlinear vehicle dynamics. Tire-road interaction, steering mechanisms, suspension behaviour, actuator delay, and surface friction introduce nonlinear effects that are difficult to reproduce within simulation environments.

Sensor modelling presents another significant challenge. Real sensors are affected by noise, measurement uncertainty, limited resolution, communication latency, and environmental interference. Simplified virtual sensors often neglect these effects, resulting in overly optimistic reinforcement learning performance.

Synchronization between physical and virtual systems also remains difficult. Differences in execution frequency, communication delay, operating system scheduling, and embedded hardware performance may introduce timing discrepancies that accumulate during long-duration experiments.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Perhaps the most important challenge concerns **digital twin validation**. Many studies assume that visually similar simulations accurately represent physical systems without performing quantitative comparisons between simulated and measured vehicle behaviour. However, visual similarity alone provides little evidence regarding behavioural fidelity. Consequently, recent research increasingly emphasizes statistical validation methodologies that compare measurable signals such as trajectory evolution, steering response, yaw angle, and acceleration between simulation and reality.

### 2.4.8 Research Gaps

The literature reveals several important limitations that motivate the work presented in this thesis.

First, numerous digital twin implementations focus primarily on graphical realism rather than behavioural accuracy. Although visually realistic simulations improve user interaction, reinforcement learning performance depends far more strongly on accurate system dynamics and sensing characteristics.

Second, relatively few studies provide quantitative validation of digital twin fidelity using measurable vehicle-state variables and statistical similarity metrics. Successful task completion alone cannot demonstrate that simulation accurately reproduces physical behaviour.

Third, existing autonomous driving research frequently relies on privileged simulator information that is unavailable during real-world deployment. Such approaches limit the practical applicability of reinforcement learning because the learned policies cannot directly operate using onboard sensors alone.

Finally, only a limited number of studies investigate low-cost scaled autonomous vehicle platforms combined with high-fidelity digital twins. These platforms offer highly reproducible research environments while significantly reducing development cost, making them particularly attractive for academic research.

These observations highlight the need for digital twins that emphasize behavioural fidelity, quantitative validation, and consistent perception interfaces. Such characteristics form the foundation of the framework proposed in this thesis.

### 2.5 Simulation Platforms for Autonomous Vehicle Research

Simulation has become an indispensable component of autonomous vehicle research, enabling the development, testing, and validation of perception, planning, and control algorithms under safe, repeatable, and cost-effective conditions. Developing autonomous driving systems exclusively on physical vehicles is often impractical due to the significant financial costs, lengthy experimental procedures, equipment wear, and safety risks associated with real-world testing. These challenges are particularly pronounced in reinforcement learning applications, where agents typically require millions of interactions with their environment before converging to an effective control policy. Consequently, simulation environments have evolved from simple vehicle dynamics models into sophisticated virtual ecosystems capable of accurately reproducing complex urban traffic scenarios, environmental conditions, sensor behaviour, and vehicle dynamics.

Modern autonomous vehicle simulators provide researchers with configurable environments that facilitate algorithm development, parameter tuning, performance evaluation, and large-scale experimentation. By allowing experiments to be repeated under identical conditions, simulation significantly improves the reproducibility of scientific research while accelerating system development. Furthermore, simulation enables researchers to investigate dangerous or rare traffic situations that would be difficult or unsafe to reproduce using physical vehicles.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



For reinforcement learning applications, simulation plays an even more critical role because it enables continuous interaction between the learning agent and its environment without risking physical damage to hardware. However, the effectiveness of simulation-based learning depends strongly on the realism of the simulation environment. Simplified or inaccurate simulation models may introduce unrealistic behaviours that degrade policy performance following deployment on physical systems. Therefore, selecting an appropriate simulation platform is one of the most important decisions in autonomous vehicle research.

Over the past decade, several simulation platforms have been developed to support autonomous driving research. Although many provide similar functionality, they differ considerably in their graphics engines, vehicle dynamics models, sensor capabilities, extensibility, and suitability for reinforcement learning. The following sections review the most widely adopted simulators used within the autonomous driving community.

### 2.5.1 CARLA Simulator

CARLA (Car Learning to Act) is currently one of the most widely adopted open-source simulation platforms for autonomous driving research [1]. Originally developed by the Computer Vision Center (CVC) at the Universitat Autònoma de Barcelona in collaboration with Intel, CARLA was specifically designed to facilitate research in autonomous driving, machine learning, and intelligent transportation systems [1].

Built upon the Unreal Engine, CARLA combines high-quality photorealistic rendering with a physics engine capable of modelling realistic vehicle dynamics and environmental interactions [1]. The simulator includes detailed urban environments featuring roads, intersections, traffic signs, pedestrians, traffic lights, and dynamic vehicles. In addition, researchers can configure numerous environmental parameters including weather conditions, lighting, road friction, and traffic density, enabling systematic evaluation under diverse operating conditions. One of CARLA's greatest strengths is its comprehensive sensor simulation framework [1].

The simulator supports a wide variety of virtual sensors, including RGB cameras, depth cameras, semantic segmentation cameras, LiDAR, radar, GNSS receivers, inertial measurement units (IMUs), collision sensors, lane invasion detectors, and custom user-defined sensors. These virtual sensors generate data formats that closely resemble those produced by corresponding physical devices, making CARLA particularly suitable for perception algorithm development and digital twin applications.

CARLA also provides an extensive Python Application Programming Interface (API), enabling researchers to control vehicles, configure simulation parameters, collect sensor data, generate training datasets, and integrate reinforcement learning algorithms. Compatibility with machine learning frameworks such as PyTorch, TensorFlow, and Stable-Baselines3 has further contributed to CARLA's widespread adoption within the reinforcement learning community.

From a digital twin perspective, CARLA offers several important advantages. Vehicle dimensions, wheelbase, steering constraints, sensor placement, and environmental geometry can be customized to closely match physical platforms, allowing researchers to construct high-fidelity virtual counterparts of real vehicles. Moreover, CARLA supports synchronous simulation, ensuring deterministic execution and precise synchronization between reinforcement learning algorithms and simulator updates.

Despite these advantages, CARLA also exhibits certain limitations. High-fidelity rendering and physics simulation require significant computational resources, particularly when multiple sensors are simulated simultaneously. Furthermore, although vehicle dynamics are considerably more realistic than



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



those of many alternative simulators, discrepancies between simulated and physical behaviour still exist, making quantitative digital twin validation essential before deploying learned controllers onto real vehicles.

### 2.5.2 Microsoft AirSim

Microsoft AirSim is another widely recognized open-source simulation platform developed primarily for autonomous drones and ground vehicles [21]. Built using Unreal Engine, AirSim provides realistic environmental rendering and supports multiple sensing modalities including cameras, LiDAR, GPS, IMUs, and barometers [21].

AirSim has been extensively used for computer vision, simultaneous localization and mapping (SLAM), and reinforcement learning research due to its flexibility and relatively simple integration with external software [21]. Researchers frequently employ AirSim for developing navigation algorithms, visual perception systems, and drone control strategies.

However, compared with CARLA, AirSim provides fewer built-in urban traffic scenarios and less comprehensive traffic infrastructure. Vehicle behaviour is generally less detailed, and the simulator lacks many autonomous driving features such as traffic management, pedestrian interaction, and configurable road networks. Consequently, although AirSim remains highly effective for aerial robotics and general autonomous navigation, it is less suitable for advanced autonomous driving research involving complex urban environments and parking scenarios.

### 2.5.3 SVL Simulator

SVL Simulator, formerly known as LGSVL Simulator, was developed to support autonomous driving research through high-fidelity simulation of urban environments. The simulator integrates closely with the Robot Operating System (ROS) and autonomous driving software stacks such as Autoware and Apollo, making it particularly attractive for full-scale autonomous vehicle development.

SVL provides realistic sensor simulation, configurable traffic participants, and detailed urban environments similar to those available in CARLA. It also supports high-definition map integration and complex scenario generation, enabling researchers to evaluate autonomous driving systems under realistic operating conditions.

Although SVL has demonstrated strong performance in industrial autonomous driving research, its installation and configuration are generally more complex than CARLA. Furthermore, community adoption has declined following changes in project development, reducing the availability of tutorials, community support, and third-party resources. As a result, CARLA has become the preferred simulator for many reinforcement learning researchers.

### 2.5.4 Gazebo

Gazebo is one of the most widely used robotic simulation environments and has become a standard tool within the Robot Operating System (ROS) ecosystem [22]. Gazebo provides realistic rigid-body physics, sensor simulation, and flexible robot modelling through the Unified Robot Description Format (URDF) [22].

Researchers frequently employ Gazebo for mobile robot navigation, manipulator control, multi-robot coordination, and reinforcement learning.

Despite its versatility, Gazebo offers relatively limited support for realistic urban driving environments. Creating detailed road networks, traffic systems, and high-quality visual scenes typically requires



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



substantial manual development. Consequently, Gazebo is generally more appropriate for robotic research than for autonomous vehicle applications involving complex traffic interactions.

### 2.5.5 Webots

Webots is another popular open-source robotics simulator supporting educational, industrial, and research applications [23]. It includes extensive libraries of robotic components, sensors, actuators, and programmable environments, making it suitable for rapid prototyping of robotic algorithms [23].

Webots supports multiple programming languages and provides straightforward integration with ROS and machine learning libraries. However, similar to Gazebo, its emphasis lies primarily in robotics rather than intelligent transportation systems. Although autonomous vehicle experiments can be implemented within Webots, its environmental realism and autonomous driving capabilities remain less sophisticated than those provided by CARLA.

### 2.5.6 Comparative Analysis of Autonomous Driving Simulators

Each simulation platform offers distinct advantages depending on the target application. General robotics simulators such as Gazebo and Webots provide excellent flexibility for robotic system development but lack realistic urban driving environments. AirSim offers high-quality rendering and strong support for aerial robotics but provides fewer autonomous driving features. SVL Simulator delivers excellent integration with industrial autonomous driving software stacks but has experienced reduced community support in recent years.

CARLA distinguishes itself through its combination of realistic vehicle dynamics, extensive sensor simulation, configurable urban environments, active research community, comprehensive documentation, and strong compatibility with reinforcement learning frameworks. These characteristics explain its widespread adoption within the autonomous driving research community.

Table 2.2 summarizes the characteristics of the most widely used autonomous vehicle simulation platforms.

| Simulator            | Primary Application  | Physics Engine | Sensor Simulation | Urban Traffic | Reinforcement Learning Support | Open Source |
|----------------------|----------------------|----------------|-------------------|---------------|--------------------------------|-------------|
| <b>CARLA</b>         | Autonomous Driving   | Unreal Engine  | Excellent         | Excellent     | Excellent                      | Yes         |
| <b>AirSim</b>        | Robotics & Drones    | Unreal Engine  | Excellent         | Moderate      | Good                           | Yes         |
| <b>SVL Simulator</b> | Autonomous Driving   | Unity          | Excellent         | Excellent     | Good                           | Yes         |
| <b>Gazebo</b>        | Robotics             | ODE/DART       | Good              | Limited       | Good                           | Yes         |
| <b>Webots</b>        | Robotics & Education | ODE            | Good              | Limited       | Moderate                       | Yes         |

Although all five platforms provide valuable research capabilities, CARLA offers the most balanced combination of realism, flexibility, extensibility, and community support for digital twin development and reinforcement learning-based autonomous driving.

### 2.5.7 Justification for Selecting CARLA

The selection of an appropriate simulation platform must align closely with the objectives of the proposed research. Since this thesis focuses on developing a high-fidelity digital twin for reinforcement learning-based autonomous parking, the simulator must accurately reproduce vehicle dynamics, support



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



realistic sensor simulation, allow customization of vehicle geometry, and provide interfaces compatible with machine learning frameworks.

CARLA satisfies these requirements more effectively than the alternative simulators reviewed in this chapter [1]. Its synchronous simulation mode ensures deterministic execution during reinforcement learning, while its Python API facilitates seamless integration with Stable-Baselines3 and custom control software [1].

The simulator also allows the dimensions, wheelbase, steering characteristics, and sensor locations of the virtual vehicle to be configured to match the physical 1:10-scale platform developed in this research.

Another important consideration is reproducibility. CARLA is actively maintained by a large international research community, making experimental configurations, software versions, and datasets readily reproducible by other researchers. This characteristic aligns closely with the scientific objectives of this thesis, where quantitative validation and reproducible experimentation are central contributions.

For these reasons, CARLA was selected as the simulation environment used to construct the digital twin, train the reinforcement learning agent, and evaluate the sim-to-real transfer process presented in this work.

### 2.6 Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) has emerged as one of the most influential artificial intelligence paradigms for solving sequential decision-making problems in robotics and autonomous systems. Unlike supervised learning, which relies on labeled datasets, or unsupervised learning, which seeks to identify hidden patterns within data, reinforcement learning enables an intelligent agent to learn optimal behaviour through continuous interaction with its environment. By repeatedly observing environmental states, selecting actions, and receiving feedback in the form of rewards or penalties, the agent gradually improves its decision-making policy to maximize long-term cumulative reward.

The integration of deep neural networks with classical reinforcement learning algorithms has significantly expanded the capability of autonomous systems to solve complex control problems characterized by high-dimensional state spaces and nonlinear dynamics. Deep reinforcement learning has achieved remarkable success in robotics, game playing, manipulation, autonomous navigation, and autonomous driving. In particular, autonomous parking has become an attractive application because the task requires continuous control, nonlinear vehicle dynamics, obstacle avoidance, and sequential decision-making under uncertainty.

This section reviews the fundamental principles of reinforcement learning, examines the evolution of deep reinforcement learning algorithms, compares the most widely used continuous-control methods, and discusses the reasons for selecting Proximal Policy Optimization (PPO) as the learning algorithm employed in this thesis.

#### 2.6.1 Fundamentals of Reinforcement Learning

Reinforcement learning describes a learning framework in which an autonomous agent interacts with an environment to achieve a specified objective. During each interaction, the agent observes the current state of the environment, selects an action according to its current policy, receives a numerical reward, and transitions to a new state. Through repeated interaction, the agent seeks to maximize the cumulative discounted reward over time.

Most reinforcement learning problems are formulated as **Markov Decision Processes (MDPs)**, which are defined by the tuple:



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



$[ (S, A, P, R, \gamma) ]$

where:

- (S) represents the set of possible states,
- (A) denotes the available actions,
- (P) describes the state transition probabilities,
- (R) represents the reward function,
- ( $\gamma$ ) is the discount factor that determines the importance of future rewards.

The agent's objective is to learn an optimal policy that maximizes the expected cumulative return. Rather than optimizing only immediate rewards, reinforcement learning considers the long-term consequences of each decision, making it particularly suitable for autonomous driving tasks where current actions influence future vehicle behaviour.

In autonomous parking applications, the state typically includes sensor observations, vehicle orientation, velocity, and obstacle distances. The action space generally consists of continuous steering, throttle, and braking commands, while the reward function encourages successful parking, collision avoidance, smooth steering, and efficient manoeuvre execution.

### 2.6.2 Deep Reinforcement Learning

Classical reinforcement learning algorithms perform well in environments containing relatively small discrete state spaces. However, autonomous driving problems involve high-dimensional observations generated by cameras, LiDAR, ultrasonic sensors, and inertial measurement units. Representing every possible environmental state explicitly becomes computationally infeasible.

Deep Reinforcement Learning addresses this limitation by replacing traditional value tables with deep neural networks capable of approximating value functions or policies directly from raw observations. Instead of storing separate values for every possible state, neural networks learn compact representations that generalize across similar situations.

The introduction of Deep Q-Networks (DQN) demonstrated that convolutional neural networks could successfully approximate optimal action-value functions, enabling reinforcement learning to solve previously intractable problems. Since then, numerous deep reinforcement learning algorithms have been developed for continuous control tasks encountered in robotics and autonomous driving.

The combination of deep learning and reinforcement learning enables autonomous vehicles to learn complex nonlinear control policies while adapting to changing environments without requiring explicit mathematical models of vehicle behaviour.

### 2.6.3 Value-Based Reinforcement Learning

Value-based methods estimate the expected future reward associated with individual actions. The agent selects the action corresponding to the highest predicted value.

#### Deep Q-Network (DQN)

Deep Q-Network (DQN) represents one of the earliest successful deep reinforcement learning algorithms [18]. It employs a neural network to approximate the Q-function, which estimates the expected cumulative reward associated with each action [18].



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



DQN introduced several important innovations, including experience replay and target networks, which significantly improved training stability [18].

Experience replay stores previous interactions within a replay buffer, allowing the network to learn from randomly sampled experiences instead of highly correlated sequential observations.

Although DQN demonstrated impressive performance in discrete control problems, it is unsuitable for autonomous parking because steering angles and throttle commands are continuous variables. Discretizing continuous control actions often produces jerky vehicle motion and poor manoeuvre precision.

Consequently, continuous-control reinforcement learning algorithms have become the preferred choice for autonomous driving applications.

### 2.6.4 Policy Gradient Methods

Unlike value-based methods, policy gradient algorithms learn the control policy directly rather than estimating action values [19]. These methods optimize the probability distribution governing action selection through gradient-based optimization [19].

Policy gradient methods naturally support continuous action spaces, making them well suited for robotic manipulation and autonomous vehicle control. Furthermore, policy optimization often produces smoother control behaviour compared with discretized value-based approaches.

However, early policy gradient algorithms frequently suffered from unstable training due to large policy updates that could significantly degrade previously learned behaviour. Several advanced algorithms have subsequently been developed to improve stability and sample efficiency.

### 2.6.5 Continuous-Control Reinforcement Learning Algorithms

#### Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) extends deterministic policy gradient methods using deep neural networks. It combines actor and critic networks to learn deterministic control policies for continuous action spaces.

DDPG demonstrated promising performance in robotic control and autonomous driving because it generates continuous steering and acceleration commands. Nevertheless, the algorithm is often sensitive to hyperparameter selection and may exhibit unstable convergence due to overestimation bias within the critic network.

#### Twin Delayed Deep Deterministic Policy Gradient (TD3)

Twin Delayed Deep Deterministic Policy Gradient (TD3) was developed to address the instability observed in DDPG. The algorithm introduces three primary improvements:

- Twin critic networks to reduce value overestimation,
- Delayed policy updates,
- Target policy smoothing.

These modifications significantly improve training stability while producing more reliable continuous control policies. TD3 has demonstrated excellent performance in numerous robotics applications, although careful hyperparameter tuning remains necessary.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### Soft Actor-Critic (SAC)

Soft Actor-Critic (SAC) introduces entropy regularization into reinforcement learning, encouraging greater exploration during training. Rather than maximizing only cumulative reward, SAC also maximizes policy entropy, producing more robust and diverse control strategies.

SAC generally exhibits excellent sample efficiency and strong performance across many continuous-control benchmarks. However, the algorithm requires optimization of multiple neural networks simultaneously, increasing implementation complexity and computational requirements.

Although SAC often converges faster than alternative methods, its stochastic policy representation may complicate deployment within safety-critical autonomous driving applications requiring predictable control behaviour.

### 2.6.6 Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO), introduced by OpenAI, has become one of the most widely adopted reinforcement learning algorithms for robotics and autonomous driving [4]. PPO belongs to the family of policy optimization algorithms and was specifically designed to improve training stability while maintaining computational simplicity [4]. The central innovation of PPO is the introduction of a clipped surrogate objective function that restricts excessively large policy updates during optimization [4].

Rather than allowing unrestricted changes to the policy after each learning iteration, PPO constrains parameter updates within a predefined trust region, preventing catastrophic degradation of previously learned behaviours.

This mechanism offers several important advantages.

First, PPO provides highly stable training compared with earlier policy gradient methods. Second, implementation is relatively straightforward because the algorithm avoids the computational complexity associated with second-order optimization techniques such as Trust Region Policy Optimization (TRPO). Third, PPO demonstrates strong performance across a wide range of continuous-control tasks, including robotic locomotion, drone navigation, autonomous driving, and robotic manipulation.

In autonomous parking applications, PPO naturally generates continuous steering, throttle, and braking commands while maintaining smooth vehicle behaviour. The algorithm also exhibits strong robustness when trained using noisy sensor observations, making it particularly suitable for sim-to-real reinforcement learning.

Because of these advantages, PPO has become one of the most frequently adopted reinforcement learning algorithms within the autonomous driving research community.

### 2.6.7 Reinforcement Learning for Autonomous Parking

Autonomous parking can be naturally formulated as a sequential decision-making problem. During each simulation step, the reinforcement learning agent receives observations describing the surrounding environment and selects steering and throttle commands that maximize long-term parking performance.

Unlike traditional path-planning approaches, reinforcement learning does not explicitly compute geometric trajectories. Instead, the agent gradually discovers effective parking strategies through interaction with the environment.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Several studies have demonstrated that reinforcement learning can successfully perform reverse parking, parallel parking, and obstacle avoidance using both simulated and physical vehicle platforms [3], [5], [6], [7], [16]. However, many existing implementations rely upon privileged simulator information, including precise global coordinates or occupancy maps unavailable during real-world operation [5], [16].

Such assumptions simplify the learning problem but significantly reduce transferability to physical autonomous vehicles.

The present research addresses this limitation by utilizing onboard-equivalent ultrasonic sensor observations in both simulation and reality, thereby maintaining consistency between training and deployment environments.

### 2.6.8 Challenges of Deep Reinforcement Learning

Despite remarkable progress, several challenges continue to limit the widespread adoption of reinforcement learning within autonomous driving.

One of the most significant challenges is **sample inefficiency**. Training modern reinforcement learning algorithms often requires millions of interactions, making direct physical experimentation impractical.

Another challenge concerns **reward engineering**. Designing reward functions that simultaneously encourage safe, efficient, and comfortable driving remains difficult. Poor reward design may produce undesirable behaviours despite successful optimization.

Generalization also remains problematic. Policies trained under limited environmental conditions frequently perform poorly when encountering previously unseen road geometries, obstacle configurations, or weather conditions.

Finally, reinforcement learning algorithms remain highly dependent on simulation fidelity. Differences between simulated and physical vehicle dynamics, sensing characteristics, and actuator behaviour frequently prevent successful deployment following training. Consequently, high-fidelity digital twins have become essential components of modern reinforcement learning research.

### 2.6.9 Comparison of Deep Reinforcement Learning Algorithms

Table 2.4 summarizes the principal reinforcement learning algorithms employed for autonomous driving applications.

| Algorithm   | Action Space | Training Stability | Sample Efficiency | Complexity | Suitability for Autonomous Parking |
|-------------|--------------|--------------------|-------------------|------------|------------------------------------|
| <b>DQN</b>  | Discrete     | High               | Moderate          | Low        | Poor                               |
| <b>DDPG</b> | Continuous   | Moderate           | Good              | Moderate   | Good                               |
| <b>TD3</b>  | Continuous   | High               | Good              | Moderate   | Very Good                          |
| <b>SAC</b>  | Continuous   | Very High          | Excellent         | High       | Excellent                          |
| <b>PPO</b>  | Continuous   | Very High          | Good              | Moderate   | Excellent                          |

Although SAC and TD3 have demonstrated excellent performance in continuous-control benchmarks, PPO provides an attractive balance between training stability, implementation simplicity, computational efficiency, and reproducibility. These characteristics make PPO particularly suitable for reinforcement learning-based autonomous parking using digital twin environments.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 2.6.10 Justification for Selecting PPO

The primary objective of this thesis is to develop a reinforcement learning controller capable of reliable zero-shot transfer from a CARLA-based digital twin to a physical 1:10-scale autonomous vehicle. Achieving this objective requires an algorithm that produces stable learning behaviour, continuous control outputs, and robust convergence under noisy sensor observations.

PPO satisfies these requirements more effectively than the alternative algorithms reviewed in this chapter 4.

Its clipped policy optimization strategy minimizes unstable parameter updates while maintaining efficient learning performance. Furthermore, PPO integrates seamlessly with the Stable-Baselines3 framework and has been extensively validated within robotics and autonomous driving research.

For these reasons, PPO was selected as the reinforcement learning algorithm employed throughout this thesis.

### 2.7 Simulation-to-Real Transfer in Autonomous Driving

One of the principal motivations for using simulation in autonomous driving research is the ability to develop and evaluate algorithms safely, efficiently, and economically before deploying them on physical vehicles. Simulation allows researchers to perform millions of experiments under controlled conditions without risking hardware damage, human safety, or excessive operational costs. This capability is particularly important for reinforcement learning, where agents require extensive interaction with the environment before converging to an effective control policy.

Despite these advantages, a persistent challenge remains: controllers that perform exceptionally well in simulation frequently experience significant performance degradation when deployed in the real world. This phenomenon, commonly known as the **simulation-to-reality (sim-to-real) gap** or **reality gap**, has become one of the most important research topics in robotics and autonomous driving. The reality gap arises because simulation environments can only approximate the behaviour of physical systems, resulting in discrepancies between virtual and real-world operation.

Simulation-to-real transfer refers to the process of deploying controllers, perception systems, or decision-making policies developed entirely within simulation onto physical platforms while maintaining comparable performance. Ideally, an autonomous vehicle trained exclusively in simulation should operate safely and effectively in the real world without requiring additional training or manual modifications. Achieving this objective would substantially reduce development time, improve safety during controller design, and lower the cost of autonomous vehicle research.

However, realizing reliable sim-to-real transfer remains challenging because even minor differences between simulation and reality may accumulate over time, causing learned policies to fail in situations that were successfully handled during training. Consequently, reducing the reality gap has become a fundamental objective of modern reinforcement learning and digital twin research.

#### 2.7.1 Sources of the Reality Gap

The reality gap originates from numerous factors associated with both the physical system and the simulation environment [2], [9]. One important source of discrepancy is vehicle dynamics. Simulated vehicles typically rely on mathematical models that approximate steering behaviour, tire-road interaction, suspension dynamics, actuator response, and friction characteristics. Physical vehicles, however, exhibit nonlinear behaviours influenced by manufacturing tolerances, component wear, surface irregularities, and environmental conditions. Even relatively small modelling errors may significantly alter vehicle trajectories during complex manoeuvres such as autonomous parking.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



A second source of discrepancy involves **sensor modelling**. Real sensors are affected by measurement noise, limited resolution, latency, environmental interference, calibration errors, and occasional measurement failures. Simulated sensors often neglect many of these imperfections, providing cleaner observations than those available during physical deployment. Consequently, reinforcement learning agents may learn policies that rely upon unrealistically precise sensor information.

**Actuator dynamics** also contribute to the reality gap. Steering motors, electronic speed controllers, braking systems, and communication interfaces introduce delays and nonlinearities that are difficult to reproduce accurately within simulation. Differences in control frequency or actuator response may accumulate throughout a parking manoeuvre, resulting in noticeable deviations between simulated and physical trajectories.

Environmental factors constitute another important source of mismatch. Lighting conditions, road surface characteristics, wheel slip, vibrations, weather, and unexpected obstacles introduce variability that cannot be perfectly represented within virtual environments. Although many simulators provide configurable weather and road conditions, complete replication of real-world uncertainty remains impossible.

Finally, software and hardware execution timing influence controller performance. Differences in processor speed, operating system scheduling, communication latency, and synchronization frequency may alter the sequence of observations and control commands, particularly in reinforcement learning systems operating in closed-loop configurations.

Because these discrepancies collectively influence autonomous vehicle behaviour, successful simulation performance alone cannot guarantee reliable physical deployment.

### 2.7.2 Sim-to-Real Transfer Strategies

Over the past decade, researchers have proposed numerous strategies to improve the transferability of reinforcement learning policies from simulation to physical systems. Although these approaches differ considerably in implementation, they share the common objective of reducing the discrepancy between training and deployment environments.

#### Domain Randomization

Domain randomization is one of the most widely adopted sim-to-real techniques [20]. Rather than attempting to create a perfectly accurate simulation, domain randomization intentionally introduces variability into the training environment by randomly modifying parameters such as lighting, textures, friction coefficients, sensor noise, object positions, and vehicle dynamics [20].

Domain randomization has demonstrated considerable success in robotic manipulation, drone navigation, and autonomous driving [20]. However, excessive randomization may significantly increase training complexity and slow convergence [20].

#### Domain Adaptation

Domain adaptation seeks to reduce differences between simulated and real observations by transforming data from one domain into another. Image translation techniques based on Generative Adversarial Networks (GANs), CycleGAN, and feature adaptation methods have been widely investigated for camera-based autonomous driving.

Rather than modifying vehicle dynamics, domain adaptation focuses primarily on improving perception by making simulated images appear more realistic or learning domain-invariant feature representations.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Although effective for computer vision tasks, domain adaptation is generally less suitable for autonomous parking systems relying primarily on ultrasonic sensors and vehicle dynamics, where discrepancies originate largely from physical behaviour rather than visual appearance.

### System Identification

System identification aims to improve simulation fidelity by estimating physical system parameters directly from experimental measurements. Instead of relying solely on theoretical vehicle models, researchers measure steering characteristics, acceleration response, tire behaviour, actuator dynamics, and other vehicle properties before incorporating these parameters into the simulator.

Accurate system identification significantly improves dynamic consistency between simulation and reality. However, identifying all relevant system parameters remains challenging because many vehicle characteristics vary with operating conditions and environmental factors.

### Fine-Tuning on Physical Systems

Another commonly adopted strategy involves training reinforcement learning policies initially within simulation before performing additional learning using physical hardware. Fine-tuning allows the policy to adapt to discrepancies encountered during deployment while requiring substantially fewer real-world interactions than training entirely on the physical platform.

Although effective, fine-tuning partially sacrifices one of the principal advantages of simulation because physical experimentation remains necessary. Furthermore, repeated online learning may introduce safety concerns when operating autonomous vehicles within real environments.

### 2.7.3 Digital Twins for Sim-to-Real Transfer

Recent research increasingly recognizes digital twins as a promising solution for improving sim-to-real transfer [10], [13]. Unlike conventional simulation environments, digital twins emphasize accurate reproduction of physical system behaviour through close correspondence between virtual and real platforms [10].

Within reinforcement learning, digital twins provide two important advantages. First, they allow extensive policy training under realistic operating conditions without exposing physical hardware to unnecessary risk. Second, they reduce discrepancies between simulation and reality by maintaining consistency in vehicle geometry, dynamics, sensor placement, control interfaces, and environmental representation.

A high-fidelity digital twin minimizes the likelihood that the reinforcement learning agent will exploit unrealistic simulator characteristics. Instead, the agent learns behaviours that remain applicable after deployment because the observations and control responses closely resemble those encountered by the physical vehicle.

However, constructing a digital twin alone does not guarantee successful sim-to-real transfer. The digital twin must first be quantitatively validated to demonstrate that simulated vehicle behaviour accurately reproduces physical system dynamics.

### 2.7.4 Validation of Sim-to-Real Transfer

One of the major shortcomings identified within recent literature is the lack of rigorous validation procedures for evaluating digital twin fidelity [12]. Many studies simply demonstrate successful task completion in both simulation and reality without providing quantitative evidence that the underlying



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



vehicle dynamics are equivalent [12].

Behavioural validation requires comparison of measurable variables collected simultaneously from both simulated and physical systems. Common validation variables include vehicle trajectory, steering angle, yaw angle, velocity, acceleration, obstacle distance, and parking completion time.

Statistical analysis provides objective measures of similarity between simulation and reality. Metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Pearson correlation coefficient, confidence intervals, and standard deviation quantify the degree of agreement between corresponding signals.

Trajectory comparison is particularly valuable because it directly illustrates the consistency of vehicle motion throughout an entire parking manoeuvre. Closely overlapping trajectories indicate that the digital twin accurately reproduces vehicle behaviour, thereby increasing confidence that reinforcement learning policies trained in simulation will remain effective following deployment.

The use of statistical validation distinguishes high-fidelity digital twins from visually realistic simulations by providing measurable evidence of behavioural correspondence rather than relying solely on qualitative observations.

### 2.7.5 Zero-Shot Sim-to-Real Transfer

Zero-shot sim-to-real transfer represents one of the most challenging objectives in reinforcement learning. In this approach, a policy trained entirely within simulation is deployed directly onto the physical system without additional training, parameter tuning, or domain adaptation.

Successful zero-shot transfer demonstrates that the simulation environment accurately represents the physical system and that the learned policy has generalized beyond simulator-specific characteristics.

Although several recent studies have reported successful zero-shot transfer for robotic manipulation and mobile robot navigation [9], relatively few investigations have demonstrated comparable performance for autonomous parking using scaled vehicle platforms.

Furthermore, many existing implementations rely on privileged simulator observations unavailable during physical deployment, reducing the practical significance of their reported results.

The present thesis specifically addresses this challenge by developing a validated CARLA-based digital twin whose geometry, vehicle dynamics, sensor configuration, and control interfaces closely match those of the physical platform. The reinforcement learning policy is trained entirely within the digital twin using only onboard-equivalent ultrasonic sensor observations before being deployed directly onto the real vehicle without retraining.

### 2.7.6 Research Gaps

The literature reviewed in this section highlights several important limitations that motivate the proposed research.

First, many reinforcement learning studies emphasize policy performance while providing only limited discussion of the fidelity of the simulation environment used during training. Consequently, it remains difficult to determine whether successful deployment results from robust learning or simply favourable experimental conditions.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Second, relatively few studies quantitatively validate digital twin behaviour using statistical comparisons between simulated and physical vehicle dynamics. Most evaluations rely primarily on qualitative visual inspection or task completion rates.

Third, numerous autonomous driving studies employ privileged information such as global coordinates, occupancy maps, or perfect localization that cannot be obtained during real-world operation. These assumptions reduce the practical applicability of learned controllers.

Finally, only a limited number of investigations have combined high-fidelity digital twins, reinforcement learning, onboard-equivalent sensing, and zero-shot deployment on low-cost scaled autonomous vehicle platforms. Addressing these limitations represents the principal contribution of the present thesis.

### 2.8 Review of Related Work

The previous sections reviewed the theoretical foundations of autonomous vehicle systems, autonomous parking, digital twin technology, simulation environments, reinforcement learning, and simulation-to-reality transfer. This section critically examines recent research that combines these technologies for autonomous driving applications. Rather than reviewing individual studies independently, the literature is organized according to major research themes in order to identify current trends, common methodologies, existing limitations, and unresolved research challenges.

Recent research demonstrates increasing interest in integrating reinforcement learning with realistic simulation environments to develop autonomous driving controllers. At the same time, digital twin technology has emerged as a promising approach for improving simulation fidelity and facilitating the transfer of learned policies to physical systems. Despite these advances, several important challenges remain unresolved, particularly regarding quantitative digital twin validation, reproducibility, and zero-shot sim-to-real transfer.

#### 2.8.1 Conventional Autonomous Parking Systems

Early autonomous parking research focused primarily on deterministic planning and control algorithms. These methods typically decomposed the parking problem into perception, path planning, trajectory generation, and trajectory tracking modules.

Hybrid A\*, Reeds–Shepp curves, spline interpolation, and clothoid-based trajectory generation became widely adopted because they generate smooth, collision-free parking paths while respecting vehicle kinematic constraints [14]. Combined with controllers such as Pure Pursuit or Model Predictive Control (MPC), these approaches achieved high positioning accuracy in structured environments [7].

Although these methods demonstrated reliable performance, they depend heavily on accurate mathematical models, precise localization, and carefully tuned controller parameters. Their modular architectures also make adaptation to new environments difficult because each subsystem must be redesigned independently.

Recent optimization-based approaches have attempted to improve flexibility by solving parking as a nonlinear constrained optimization problem. While these techniques produce high-quality trajectories, their computational requirements increase significantly as environmental complexity grows. Furthermore, optimization methods often require detailed environmental maps and exact vehicle models that may not always be available.

These limitations have motivated increasing interest in learning-based autonomous parking methods capable of directly learning control policies from interaction with the environment.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 2.8.2 Reinforcement Learning-Based Autonomous Parking

The application of reinforcement learning to autonomous parking has expanded rapidly during the past decade. Instead of explicitly generating trajectories, reinforcement learning algorithms enable vehicles to learn parking strategies through repeated interaction with simulated environments.

Early reinforcement learning implementations employed discrete-action algorithms such as Q-learning and Deep Q-Networks (DQN). Although successful for simple navigation tasks, these approaches were unsuitable for parking because discrete steering actions often produced discontinuous vehicle motion.

The introduction of continuous-control algorithms, including DDPG, TD3, SAC, and PPO, significantly improved parking performance by generating smooth steering and throttle commands [4]. Among these algorithms, PPO has become one of the most frequently adopted due to its stable policy optimization, relatively straightforward implementation, and robust convergence characteristics [4].

Several recent studies have demonstrated successful autonomous parking using PPO within simulation environments such as CARLA, AirSim, and custom robotic simulators [3], [5], [6], [7], [8], [16]. Reported improvements include smoother trajectories, reduced collision rates, and higher parking success rates compared with traditional control strategies [3], [16]. Nevertheless, careful examination of the literature reveals several recurring limitations.

First, many reinforcement learning studies rely upon privileged simulator information, including perfect localization, occupancy grids, global coordinates, or complete environmental maps. Such information is generally unavailable during practical deployment, reducing the realism of the learning process.

Second, experimental evaluation frequently focuses only on parking success rates while providing limited analysis of vehicle dynamics, trajectory consistency, or controller robustness.

Finally, relatively few investigations demonstrate successful deployment on physical vehicles, leaving uncertainty regarding the practical applicability of many proposed methods.

### 2.8.3 Digital Twins for Autonomous Vehicle Research

Digital twin technology has recently become an active area of research within autonomous driving due to its potential to improve simulation fidelity and reduce the simulation-to-reality gap.

Numerous studies have developed virtual counterparts of autonomous vehicles capable of reproducing environmental perception, vehicle dynamics, and control interfaces [10], [11], [13]. These digital twins have been employed for virtual testing, predictive maintenance, traffic management, and controller development [11], [12], [13].

Several researchers have emphasized the importance of maintaining geometric consistency between physical and simulated vehicles by matching dimensions, wheelbase, steering geometry, and sensor placement. Such correspondence reduces discrepancies encountered during controller deployment.

More recent investigations have extended digital twin functionality by integrating real-time communication between physical and virtual systems through Internet of Things (IoT) technologies and cloud computing infrastructures [12].

These systems enable continuous synchronization between physical vehicles and their virtual counterparts for monitoring and predictive analysis.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Although these developments represent important progress, many published digital twins emphasize graphical realism while providing limited quantitative evaluation of dynamic fidelity. In numerous studies, successful visual correspondence is assumed to indicate accurate behavioural reproduction without comparing measurable vehicle responses between simulation and physical operation.

Consequently, uncertainty remains regarding whether reinforcement learning agents trained within these digital twins actually experience vehicle behaviour representative of the real system.

### 2.8.4 Simulation Platforms for Reinforcement Learning

Simulation environments have become essential tools for reinforcement learning because they enable rapid generation of training data under safe and repeatable conditions.

Among available simulation platforms, CARLA has emerged as the dominant environment for autonomous driving research due to its realistic urban environments, configurable weather conditions, comprehensive sensor simulation, and extensive Python API [1].

Several recent reinforcement learning studies have successfully employed CARLA to investigate lane following, autonomous navigation, intersection management, and autonomous parking. Researchers have demonstrated that CARLA provides sufficiently realistic environments for developing sophisticated perception and control algorithms while significantly reducing development costs. AirSim, Gazebo, and SVL Simulator have also been employed within autonomous driving research [21], [22].

However, their adoption for reinforcement learning-based parking remains considerably lower than CARLA due to differences in environmental realism, available driving scenarios, and community support.

Despite widespread adoption of CARLA, many published studies continue to rely upon default vehicle configurations rather than constructing digital twins that accurately replicate physical research platforms. Consequently, discrepancies between simulated and physical vehicle dynamics frequently remain unaddressed.

### 2.8.5 Sim-to-Real Reinforcement Learning

The simulation-to-reality gap has become one of the principal obstacles limiting practical deployment of reinforcement learning in autonomous systems.

To improve policy transferability, researchers have investigated domain randomization, domain adaptation, system identification, curriculum learning, and online fine-tuning.

Domain randomization improves policy robustness by exposing reinforcement learning agents to randomized environmental conditions during training [20]. While effective in many applications, excessive randomization often increases training complexity and may reduce convergence speed [20], relying primarily on distance measurements rather than visual information.

System identification improves transferability by incorporating experimentally measured vehicle parameters into simulation models. Accurate identification of steering dynamics, tire characteristics, and actuator response substantially reduces discrepancies between simulation and physical behaviour. However, many vehicle properties remain difficult to identify accurately due to nonlinear operating conditions.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Several investigations have reported successful zero-shot transfer of reinforcement learning policies from simulation to physical robotic platforms. Nevertheless, relatively few studies have demonstrated comparable success for autonomous parking using scaled autonomous vehicles while simultaneously validating digital twin fidelity through quantitative statistical analysis.

### 2.8.6 Critical Analysis of Existing Literature

Although considerable progress has been achieved in reinforcement learning, digital twins, and autonomous driving, several common limitations appear consistently throughout the literature.

The first limitation concerns **simulation fidelity**. Numerous studies employ simulation environments for reinforcement learning without evaluating whether simulated vehicle behaviour accurately matches the physical platform. Consequently, successful controller performance within simulation cannot necessarily be interpreted as evidence of successful real-world deployment.

A second limitation involves **insufficient quantitative validation**. Many researchers evaluate autonomous parking using qualitative demonstrations or parking success rates while neglecting detailed analysis of trajectory evolution, steering behaviour, yaw response, or statistical similarity between simulation and reality.

The third limitation concerns **sensor assumptions**. A large proportion of reinforcement learning studies rely upon idealized observations including perfect localization, occupancy maps, or global coordinates unavailable during practical deployment. Such assumptions reduce the realism of the learning problem and limit the practical applicability of reported results.

Another recurring issue is **reproducibility**. Published studies frequently omit important implementation details, including reinforcement learning hyperparameters, reward functions, neural network architectures, vehicle configurations, and simulation settings. This lack of documentation makes independent verification and comparison difficult.

Finally, many investigations focus primarily on algorithmic improvements while treating the simulation environment as a secondary consideration. However, reinforcement learning performance depends strongly upon simulation fidelity, making digital twin validation equally important as algorithm selection.

### 2.8.7 Research Gap

The literature reviewed throughout this chapter demonstrates substantial progress in autonomous parking, reinforcement learning, and digital twin technology. Nevertheless, several important research gaps remain.

First, few studies develop **high-fidelity digital twins** whose vehicle geometry, steering characteristics, sensing configuration, and dynamic behaviour are systematically matched to corresponding physical platforms.

Second, quantitative validation of digital twin fidelity remains relatively uncommon. Most existing research relies upon visual comparison or successful task completion rather than statistical evaluation of vehicle dynamics.

Third, reinforcement learning policies are frequently trained using privileged information unavailable during real-world operation, reducing their transferability to physical autonomous vehicles.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Fourth, only a limited number of investigations demonstrate **zero-shot sim-to-real transfer** for reinforcement learning-based autonomous parking using low-cost scaled autonomous vehicle platforms.

Finally, comprehensive frameworks integrating validated digital twins, realistic onboard sensing, reinforcement learning, and quantitative experimental evaluation remain scarce within the existing literature.

### 2.8.8 Positioning of the Present Research

The research presented in this thesis directly addresses the limitations identified above.

A high-fidelity digital twin of a 1:10-scale autonomous vehicle is developed within the CARLA simulator by replicating the physical vehicle geometry, steering behaviour, sensor placement, and control interfaces. Unlike many previous reinforcement learning studies, the proposed framework employs only onboard-equivalent ultrasonic sensor observations during both training and deployment, eliminating dependence on privileged simulator information.

The digital twin is quantitatively validated through deterministic parking experiments that compare simulated and physical vehicle trajectories and dynamic responses using statistical performance metrics. Following validation, a PPO-based reinforcement learning controller is trained entirely within the digital twin and subsequently deployed directly onto the physical platform through zero-shot sim-to-real transfer without additional retraining.

By combining validated digital twin development, realistic sensing, reinforcement learning, and quantitative experimental analysis, this thesis contributes a comprehensive framework that advances the practical deployment of reinforcement learning within autonomous parking systems.

### 2.9 Research Gaps and Motivation

The literature reviewed throughout this chapter demonstrates that considerable progress has been made in autonomous driving, autonomous parking, digital twin technology, reinforcement learning, and simulation-to-real transfer. Advances in sensing technologies, simulation platforms, and deep reinforcement learning algorithms have enabled autonomous vehicles to perform increasingly complex driving tasks under controlled conditions. Similarly, the emergence of digital twins has provided researchers with powerful tools for virtual testing, controller development, and system optimization. Despite these achievements, several important research challenges remain unresolved, limiting the practical deployment of reinforcement learning-based autonomous parking systems.

The first major research gap concerns the **fidelity of simulation environments** used for reinforcement learning. Many studies employ advanced simulators such as CARLA, AirSim, or SVL to train autonomous driving agents; however, relatively little attention is given to ensuring that the simulated vehicle accurately reproduces the behaviour of its physical counterpart. In numerous publications, vehicle models are adopted directly from the simulator without systematic calibration or validation against experimental measurements. Consequently, reinforcement learning agents may learn control policies that exploit simulator-specific characteristics rather than behaviours that are transferable to real vehicles. This limitation significantly reduces the reliability of simulation-based controller development.

A second limitation identified throughout the literature is the **lack of quantitative validation of digital twins**. Although many researchers describe their simulation environments as digital twins, only a small proportion provide objective evidence demonstrating that the virtual model accurately reproduces physical vehicle behaviour. In many cases, validation is restricted to qualitative visual comparisons or



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



successful task completion within both simulation and reality. Such evaluations provide only limited evidence regarding behavioural fidelity because visually similar motion does not necessarily imply identical vehicle dynamics. Quantitative comparisons involving vehicle trajectories, steering response, yaw angle, velocity, acceleration, and statistical similarity metrics remain relatively uncommon. As a result, confidence in the accuracy of many published digital twins remains limited.

Another important research gap involves the **use of privileged information during reinforcement learning**. A large proportion of existing studies train reinforcement learning agents using observations that are unavailable during practical deployment, including precise global coordinates, occupancy maps, ideal localization systems, or complete environmental information. While these assumptions simplify the learning problem and accelerate convergence, they reduce the realism of the training environment and limit the applicability of the resulting control policies. Autonomous vehicles operating in real environments must rely exclusively on onboard sensors whose measurements are inherently affected by uncertainty, latency, and noise. Therefore, reinforcement learning frameworks should ideally employ observation spaces that remain consistent between simulation and physical deployment.

The literature also reveals limited research concerning **low-cost scaled autonomous vehicle platforms**. Most published investigations focus either on purely simulated environments or expensive full-scale autonomous vehicles requiring sophisticated sensing systems such as LiDAR, high-precision GPS, and multiple cameras. Although these platforms provide valuable research insights, they are often financially inaccessible to many academic laboratories and educational institutions. Scaled autonomous vehicles offer an attractive alternative by providing reproducible, cost-effective experimental platforms while preserving the essential characteristics of autonomous driving systems. Nevertheless, relatively few studies integrate scaled platforms with validated digital twins and reinforcement learning.

A further limitation concerns the **simulation-to-reality transfer process**. Existing approaches frequently employ domain randomization, domain adaptation, system identification, or additional fine-tuning on the physical platform to improve policy transferability. Although these techniques have demonstrated promising results, they often require extensive parameter tuning, additional computational resources, or further physical experimentation after simulation-based training. Comparatively fewer studies investigate zero-shot sim-to-real transfer, where a reinforcement learning policy trained entirely within simulation is deployed directly onto the physical vehicle without retraining. Successfully achieving zero-shot deployment would significantly reduce development time while demonstrating a high level of digital twin fidelity.

The issue of **experimental reproducibility** also remains insufficiently addressed in many published studies. Important implementation details including neural network architectures, reinforcement learning hyperparameters, reward functions, simulator configurations, training procedures, and experimental protocols are often omitted or only briefly described. This lack of transparency makes independent replication difficult and limits meaningful comparison between different reinforcement learning approaches. Improving reproducibility is therefore essential for advancing autonomous driving research and establishing reliable benchmarks for future investigations.

Another recurring observation is that many existing studies evaluate autonomous parking primarily using overall task completion rates. While parking success is undoubtedly an important performance indicator, it provides only a limited understanding of controller behaviour. Comprehensive evaluation should include additional performance metrics such as trajectory accuracy, collision rate, steering smoothness, parking time, control stability, statistical variability, and confidence intervals. Such quantitative analyses provide deeper insight into system performance and facilitate objective comparison with alternative approaches.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



The present thesis has been designed specifically to address these research gaps through the development of a comprehensive reinforcement learning framework centered on a validated digital twin. A high-fidelity virtual model of a 1:10-scale autonomous vehicle is constructed within the CARLA simulator by carefully reproducing the physical vehicle's geometry, steering characteristics, sensor placement, and control interfaces. The digital twin is quantitatively validated using deterministic parking experiments and statistical comparisons between simulated and physical vehicle behaviour. Unlike many previous reinforcement learning studies, the proposed framework utilizes only onboard-equivalent ultrasonic sensor observations during both training and deployment, thereby eliminating dependence on privileged simulator information. A PPO-based reinforcement learning controller is trained entirely within the validated digital twin and deployed directly onto the physical platform through zero-shot simulation-to-reality transfer without additional retraining.

By integrating validated digital twin development, realistic sensor modelling, reinforcement learning, and comprehensive experimental evaluation within a single framework, this research contributes toward improving the reliability, reproducibility, and practical applicability of simulation-based autonomous parking systems. The proposed methodology therefore represents a meaningful advancement beyond many existing approaches reported within the current literature.

### 2.10 Chapter Summary

This chapter presented a comprehensive review of the literature relevant to autonomous parking, digital twin technology, reinforcement learning, simulation environments, and simulation-to-reality transfer. The review began by examining the fundamental architecture of autonomous vehicle systems and discussing the perception, localization, planning, and control modules that enable autonomous driving. Existing autonomous parking approaches were then analysed, highlighting the evolution from classical model-based planning algorithms toward modern learning-based techniques capable of acquiring driving behaviour through interaction with the environment.

The chapter subsequently examined the development of digital twin technology and its growing importance within autonomous vehicle research. It was shown that digital twins provide significantly greater value than conventional simulations by accurately reproducing the geometry, dynamics, sensing characteristics, and operational behaviour of physical systems. The discussion also emphasized the increasing role of digital twins in reinforcement learning, where realistic virtual environments enable safe and efficient controller development while reducing the simulation-to-reality gap.

A review of contemporary autonomous driving simulators demonstrated that CARLA provides one of the most comprehensive research platforms due to its realistic urban environments, advanced sensor simulation, flexible programming interface, and strong support for reinforcement learning. The theoretical foundations of deep reinforcement learning were then presented, including value-based methods, policy-gradient approaches, and actor-critic algorithms. Among the available continuous-control algorithms, Proximal Policy Optimization (PPO) was identified as particularly well suited for autonomous parking because of its stable policy optimization, continuous action representation, and strong empirical performance.

The chapter further examined the simulation-to-reality transfer problem, identifying discrepancies in vehicle dynamics, sensing systems, actuator behaviour, and environmental conditions as the primary sources of the reality gap. Existing transfer strategies (including domain randomization, domain adaptation, system identification, and fine-tuning) were critically reviewed, demonstrating that no single approach completely eliminates the challenges associated with transferring reinforcement learning policies from simulation to physical platforms. High-fidelity digital twins combined with rigorous



## **Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle**



quantitative validation were identified as one of the most promising directions for improving sim-to-real transfer.

Finally, a critical review of recent research highlighted several important limitations within the existing literature, including insufficient validation of digital twin fidelity, dependence on privileged simulator information, limited investigation of low-cost scaled autonomous vehicle platforms, inadequate experimental reproducibility, and the absence of comprehensive quantitative performance evaluation. These observations established the research gap that motivates the work presented in this thesis.

Based on these findings, the next chapter describes the methodology adopted to address the identified limitations. It presents the design and implementation of the proposed CARLA-based digital twin, the architecture of the physical 1:10-scale autonomous vehicle, the reinforcement learning framework based on PPO, the sensor integration strategy, the experimental procedures, and the quantitative validation methodology employed to evaluate the fidelity of the digital twin and the effectiveness of zero-shot simulation-to-reality transfer.

## Chapter 3 - System Architecture

### 3.1 Overview of the Proposed Framework

The proposed system architecture is designed to establish a complete and reproducible sim-to-real pipeline for autonomous parking using a low-cost 1:10-scale vehicle and a CARLA-based digital twin. The architecture integrates hardware, embedded computing, sensing, communication, simulation, and control components into a unified framework that preserves consistency between the simulated and physical domains. The primary design objective is to ensure that both environments expose the same control interface, sensing structure, and timing behaviour so that policies developed in simulation can be transferred directly to the real vehicle with minimal adaptation.

The complete framework consists of two tightly connected subsystems:

1. the physical autonomous vehicle platform, and
2. the CARLA-based digital twin.

The physical platform is responsible for executing autonomous parking manoeuvres in the real world using onboard sensing and computation. The digital twin reproduces the geometry, dynamics, sensor configuration, and control interface of the physical vehicle inside the simulation environment. Both systems operate under the same continuous action space and equivalent perception architecture, enabling seamless portability of both scripted and learned controllers.

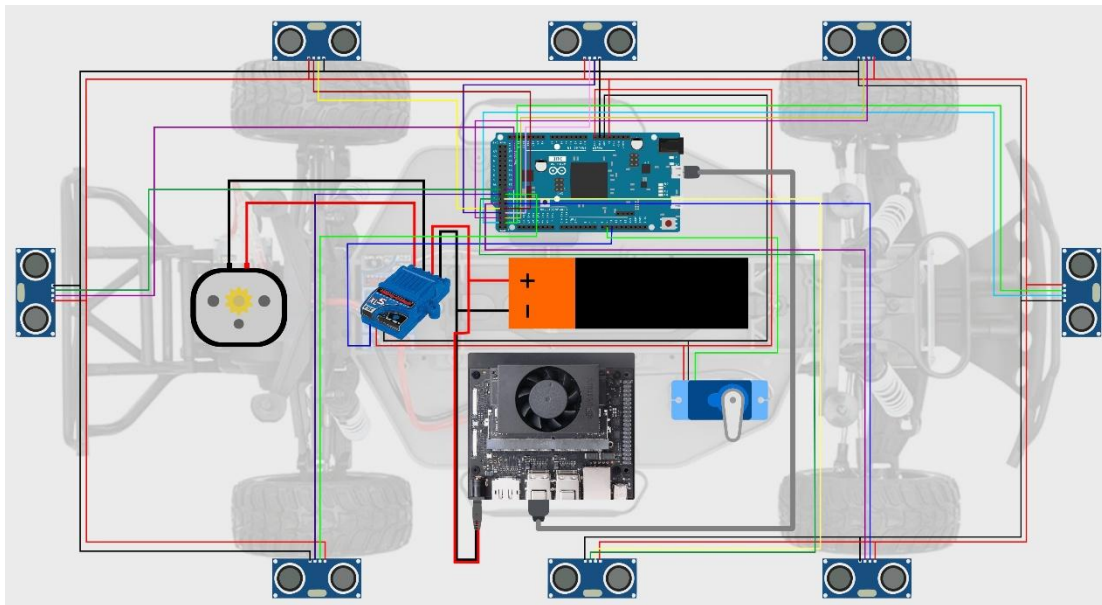


Figure 1, High-level architecture of the proposed framework.

The physical vehicle is based on a commercial 1:10-scale Traxxas Slash RC platform modified for autonomous operation. The original radio-control interface is removed and replaced by an onboard embedded architecture composed of an NVIDIA Jetson Orin Nano and an Arduino Due. The Jetson Orin Nano performs all high-level autonomous-driving tasks, including sensor processing, decision making, control inference, and communication with the simulation framework. The Arduino Due acts as a low-level interface responsible for sensor acquisition and PWM-based actuator control.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



The sensing layer consists primarily of eight HC-SR04 ultrasonic range sensors distributed around the vehicle body to provide short-range obstacle detection during parking manoeuvres. These sensors emulate the functionality of ultrasonic parking-assistance systems commonly found in commercial passenger vehicles. Additional vehicle-state information, such as yaw and acceleration, is obtained from onboard inertial measurements and integrated into the control loop.

The communication architecture separates high-level autonomy from low-level hardware control. The Jetson and Arduino communicate through a bidirectional USB serial interface. High-level control commands generated by the autonomous controller are transmitted from the Jetson to the Arduino, which converts them into actuator-level PWM signals for the steering servo and Electronic Speed Controller (ESC). Sensor measurements acquired by the Arduino are transmitted back to the Jetson for processing and control inference.

The digital twin reproduces this same architecture inside CARLA. The simulated vehicle is configured to match the dimensions, steering limits, mass properties, and throttle characteristics of the physical platform. Similarly, the ultrasonic sensors are emulated through raycasting operations positioned identically to the real sensors. The same continuous control commands used on the physical vehicle are directly applied to the simulated platform without modification.

A key feature of the proposed architecture is the preservation of a shared action space between simulation and reality. Both the physical and simulated vehicles interpret control commands identically in terms of steering, throttle, and braking behaviour. This consistency minimizes interface discrepancies and ensures that reinforcement-learning policies trained in simulation remain compatible with the real vehicle.

The architecture also emphasizes deterministic timing synchronization. The control frequency, physics timestep, and command-update rate are aligned between CARLA and the real platform to reduce temporal inconsistencies that could otherwise contribute to the reality gap. By maintaining equivalent timing behaviour across domains, the framework improves the fidelity of sim-to-real transfer.

From a systems-engineering perspective, the proposed framework follows a layered architecture composed of:

- a perception layer,
- a low-level hardware abstraction layer,
- a high-level control layer,
- and a simulation-validation layer.

This modular structure improves maintainability, reproducibility, and extensibility while allowing future integration of additional sensors, controllers, or learning algorithms.

It should be noted that the digital twin developed in this thesis implements a **calibrate-then-validate-then-deploy** workflow rather than a continuously synchronized bidirectional twin. Real-world measurements are used offline to calibrate the CARLA vehicle model (Chapter 4) and to validate its fidelity (Chapter 5, Chapter 7); once validated, the twin is used to train the reinforcement learning policy without further real-world data exchange, and the resulting policy is deployed to the physical vehicle in a single zero-shot transfer. There is no runtime feedback loop from the physical vehicle back into the simulator during either training or deployment. This scope is consistent with the calibration/validation definition adopted in 2.4.2, but narrower than architectures that maintain live synchronization (2.4.3) throughout the system's operational lifecycle; this distinction is discussed further as a limitation in 8.4.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



The remainder of this chapter describes each subsystem in detail, including the physical vehicle platform, sensing architecture, embedded computing infrastructure, software framework, communication pipeline, and continuous action-space implementation.

### 3.2 Physical Vehicle Platform

#### 3.2.1 Traxxas Slash Platform

The physical platform selected for this work is a commercially available 1:10-scale Traxxas Slash RC vehicle. The platform was chosen because it provides a cost-effective, mechanically robust, and easily modifiable base for autonomous-driving experimentation while preserving essential characteristics of full-scale car-like vehicles.

Scaled autonomous vehicles represent an attractive compromise between simulation-only approaches and full-scale autonomous-car experimentation. Full-scale autonomous platforms involve significant financial cost, safety constraints, infrastructure requirements, and operational complexity. In contrast, small-scale platforms enable safe indoor experimentation, rapid iteration, and reproducible testing conditions while still maintaining realistic steering and longitudinal vehicle dynamics.

The Traxxas Slash platform provides several advantages for sim-to-real validation:

- Ackermann-style steering geometry,
- servo-controlled front-wheel steering,
- ESC-based longitudinal control,
- sufficient payload capacity for onboard computation,
- mechanical durability,
- and availability of replacement components.

The vehicle chassis was modified to support autonomous operation by integrating onboard computing hardware, sensor mounts, communication electronics, and power-distribution circuitry. The original radio receiver was removed to eliminate manual intervention and allow direct software-based control of all actuators.

The resulting platform preserves the original mechanical structure of the RC vehicle while enabling complete autonomous operation under external computational control.

#### 3.2.2 Steering System

The steering subsystem is based on a servo-actuated front-wheel steering mechanism typical of RC vehicles. Steering commands are generated digitally by the high-level controller and converted into PWM signals that drive the steering servo.

The steering angle is constrained to an experimentally calibrated range of approximately:

$$\delta \in [-30^\circ, +30^\circ]$$

This limitation is imposed to avoid excessive mechanical stress and nonlinear steering behaviour near the physical limits of the servo mechanism. The same steering constraints are enforced inside the CARLA digital twin to maintain behavioural consistency between simulation and reality.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Steering commands are represented as normalized continuous values:

$$u_{steer} \in [-1, +1]$$

which are linearly mapped to the physical steering range. This normalized representation simplifies controller design and allows identical control policies to operate in both domains.

Because parking manoeuvres require high steering precision at low velocities, steering consistency plays a critical role in minimizing the sim-to-real discrepancy.

### 3.2.3 Electronic Speed Controller and Longitudinal Motion

Longitudinal vehicle motion is controlled using an Electronic Speed Controller (ESC) connected to the DC motor of the Traxxas platform. The ESC regulates both forward and reverse motion through PWM-based throttle commands.

Unlike conventional automotive braking systems, the RC platform does not include an independent braking actuator. Instead, braking is emulated electronically by applying a throttle signal of opposite polarity relative to the current direction of motion. This behaviour reproduces the electronic braking mechanism implemented in CARLA.

Throttle commands are represented through a normalized continuous variable:

$$u_{throttle} \in [-1, +1]$$

where:

- positive values correspond to forward motion,
- negative values correspond to reverse motion,
- and zero represents idle operation.

To ensure consistent low-speed parking behaviour, the minimum throttle required to overcome static friction and drivetrain resistance was experimentally identified for both forward and reverse motion. Equivalent longitudinal dynamics were then reproduced inside CARLA by adjusting the simulated torque characteristics.

Low-speed consistency is particularly important because parking manoeuvres are highly sensitive to small velocity variations and actuator nonlinearities.

### 3.2.4 Power Distribution and Electrical Integration

The autonomous platform integrates multiple electronic subsystems requiring stable and isolated power delivery. The primary battery supplies power to:

- the propulsion motor,
- the steering servo,
- the ESC,
- the NVIDIA Jetson Orin Nano,
- the Arduino Due,
- and the sensor array.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Voltage regulation circuitry is employed to ensure stable operating conditions for the embedded computing components and sensors. Electrical isolation between high-current motor circuitry and low-power sensing electronics helps reduce electromagnetic interference and communication instability.

Careful power-distribution design is essential because unstable voltage conditions can introduce sensor noise, actuator jitter, and timing inconsistencies that negatively affect sim-to-real reproducibility.

### 3.3 Sensor Suite

#### 3.3.1 Ultrasonic Sensors

The perception system primarily relies on eight HC-SR04 ultrasonic range sensors distributed around the vehicle body. Ultrasonic sensing was selected because parking manoeuvres occur at low speed and require reliable short-range obstacle detection rather than long-range environmental perception.

Ultrasonic sensors provide several advantages:

- low cost,
- lightweight integration,
- simple interfacing,
- low computational requirements,
- and suitability for close-range detection.

Each sensor measures obstacle distance by emitting an ultrasonic pulse and measuring the echo return time.

The measured distance is computed as:

$$d = \frac{v \cdot t}{2}$$

where:

- $d$  is the measured distance,
- $v$  is the speed of sound,
- and  $t$  is the round-trip echo time.

#### 3.3.2 Sensor Placement

The sensors are distributed to maximize obstacle coverage during parking manoeuvres. The configuration consists of:

- three sensors on the left side,
- three sensors on the right side,
- one front-facing sensor,
- and one rear-facing sensor.

This arrangement enables the vehicle to estimate distances relative to parked vehicles, roadside boundaries, and nearby obstacles during slot detection and reverse parking.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



The sensor layout was replicated identically in the digital twin through simulated raycasts positioned at the same geometric locations.

### 3.3.3 Sensor Limitations

Although ultrasonic sensors are inexpensive and effective for close-range detection, they exhibit several limitations:

- limited angular resolution,
- measurement noise,
- surface-dependent reflections,
- interference between sensors,
- and reduced accuracy near corners and edges.

These limitations become particularly significant during the final alignment phase of parking manoeuvres, where small perception errors can lead to incorrect steering decisions.

As later experimental results demonstrate, perception uncertainty constitutes one of the primary remaining contributors to the residual sim-to-real gap.

## 3.4 Embedded Computing Architecture

### 3.4.1 NVIDIA Jetson Orin Nano

The NVIDIA Jetson Orin Nano serves as the high-level computational unit of the autonomous platform. It executes:

- autonomous control logic,
- reinforcement-learning inference,
- sensor processing,
- data logging,
- and communication management.

The Jetson platform was selected because it combines GPU acceleration with compact size and low power consumption, making it well suited for onboard AI applications.

Both the deterministic parking controller and the PPO reinforcement-learning policy execute directly on the Jetson during real-world operation.

### 3.4.2 Arduino Due

The Arduino Due functions as the low-level hardware abstraction layer between the Jetson and the physical actuators/sensors.

Its responsibilities include:

- ultrasonic sensor acquisition,
- PWM signal generation,
- actuator interfacing,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- and serial communication.

The Arduino continuously receives control commands from the Jetson and maintains the latest command values until updated instructions arrive.

This architecture separates computationally intensive AI processing from time-sensitive hardware control operations.

### 3.4.3 Serial Communication Interface

Communication between the Jetson and Arduino is established through a bidirectional USB serial interface.

The Jetson transmits:

- steering commands,
- throttle commands,
- and braking commands.

The Arduino transmits:

- ultrasonic measurements,
- timing information,
- and low-level telemetry.

The communication pipeline is designed to operate deterministically within the fixed control loop period used in both simulation and reality.

## 3.5 Software Architecture

### 3.5.1 High-Level Control Layer

The high-level software layer is implemented primarily in Python and manages:

- sensor preprocessing,
- control inference,
- reinforcement-learning integration,
- and CARLA communication.

This layer provides a unified software interface shared by both the simulation and physical platforms.

### 3.5.2 CARLA Integration Layer

The CARLA integration layer interfaces directly with the simulator API to:

- retrieve sensor data,
- update vehicle states,
- apply control commands,
- and synchronize simulation timing.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



This layer enables the same controller implementation to operate transparently in both domains.

### 3.5.3 Real-Time Control Loop

Both the physical and simulated systems operate under equivalent control-loop timing constraints.

The control cycle consists of:

1. sensor acquisition,
2. state construction,
3. policy inference,
4. action generation,
5. actuator command transmission.

Maintaining deterministic timing is essential for minimizing temporal discrepancies during sim-to-real transfer.

### 3.6 Continuous Action Space Design

A central contribution of the proposed architecture is the implementation of a shared continuous control interface between the digital twin and the physical platform.

The action space is defined as:

$$a_t = [u_{throttle}, u_{steer}, u_{brake}]$$

where:

- $u_{throttle}$  controls longitudinal motion,
- $u_{steer}$  controls steering angle,
- and  $u_{brake}$  controls braking behaviour.

The continuous representation enables smooth vehicle control and is particularly suitable for reinforcement-learning algorithms such as PPO.

The preservation of an identical action representation across simulation and reality eliminates the need for controller adaptation during deployment and significantly contributes to reducing the sim-to-real gap.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### Chapter 4 - CARLA Digital Twin Development

#### 4.1 Introduction

The effectiveness of a sim-to-real pipeline depends fundamentally on the quality of the simulation environment used during controller development and validation. In reinforcement-learning applications, the simulator becomes the primary source of interaction data from which the autonomous agent learns its behaviour. Consequently, if the simulated dynamics, sensing characteristics, or timing behaviour differ significantly from those of the physical platform, the learned policy may exploit simulator-specific artifacts that do not exist in reality, resulting in poor transferability after deployment.

To reduce this problem, this thesis develops a CARLA-based digital twin of the physical autonomous vehicle platform described in Chapter 3. The objective of the digital twin is not merely to visually replicate the physical vehicle, but rather to reproduce the dynamic and perceptual behaviour of the real system under identical control inputs. The proposed twin therefore emphasizes behavioural fidelity over graphical realism.

The digital twin reproduces the geometry, steering constraints, longitudinal response, sensor placement, control-loop timing, and environmental conditions of the physical platform. By preserving consistency between the simulated and real systems, the twin enables direct portability of both deterministic and reinforcement-learning-based controllers.

This chapter describes the complete development of the CARLA digital twin, including vehicle adaptation, sensor modelling, environment configuration, synchronization strategies, and simulation timing design.

#### 4.2 CARLA Simulator Overview

CARLA is an open-source autonomous-driving simulator designed for research and development in perception, planning, and control. Built on top of Unreal Engine, CARLA provides physically simulated urban environments, configurable traffic scenarios, multiple sensor modalities, and programmable APIs for autonomous-driving experimentation.



Figure 2, CARLA simulator environment



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Several features make CARLA particularly suitable for sim-to-real research:

- physics-based vehicle simulation,
- configurable environmental conditions,
- deterministic synchronous execution,
- sensor abstraction support,
- Python API integration,
- and compatibility with machine-learning frameworks.

Unlike lightweight simulators focused primarily on reinforcement-learning benchmarking, CARLA allows direct manipulation of vehicle dynamics and sensor configurations, enabling the construction of simulation environments closely aligned with physical platforms.

In this work, CARLA serves as the foundation for:

- digital twin development,
- deterministic validation experiments,
- and PPO training.

The simulator is configured to operate in synchronous mode with fixed-timestep physics execution to ensure deterministic behaviour and reproducible experiments.

### 4.3 Digital Twin Design Objectives

The primary objective of the digital twin is to minimize the sim-to-real discrepancy while maintaining low implementation complexity and high reproducibility.

To achieve this goal, several design requirements were established:

#### 4.3.1 Dynamic Consistency

The simulated vehicle must reproduce the physical platform dynamics as closely as possible, particularly during low-speed parking manoeuvres where small steering or acceleration errors may accumulate into significant trajectory deviations.

Key dynamic properties requiring alignment include:

- wheelbase,
- steering limits,
- tire radius,
- vehicle mass,
- throttle response,
- and turning behaviour.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 4.3.2 Perception Consistency

The perception interface exposed to the controller in simulation must replicate the sensing structure available on the real vehicle.

This includes:

- identical sensor count,
- identical sensor placement,
- equivalent sensing range,
- and equivalent observation-space structure.

The objective is to ensure that policies trained in simulation receive information comparable to that available during real-world deployment.

### 4.3.3 Control Interface Consistency

The digital twin must preserve the same continuous action representation used by the physical vehicle.

The action vector is defined as:

$$a_t = [u_{throttle}, u_{steer}, u_{brake}]$$

where:

- $u_{throttle}$  controls longitudinal motion,
- $u_{steer}$  controls steering,
- and  $u_{brake}$  controls braking behaviour.

By maintaining a shared action space across domains, the same controller implementation can operate transparently in both simulation and reality.

### 4.3.4 Temporal Consistency

Timing discrepancies represent a major source of sim-to-real mismatch in autonomous systems. Differences in sensor-update frequency, control-loop timing, or physics execution may alter closed-loop behaviour even when the underlying controller remains unchanged.

To reduce temporal inconsistencies, the digital twin replicates:

- the physical control frequency,
- command-update intervals,
- and sensor-acquisition timing.

## 4.4 Vehicle Model Adaptation

### 4.4.1 Base Vehicle Selection

The Traxxas Slash platform is not available as a native CARLA vehicle asset. Therefore, the digital twin was constructed by modifying an existing CARLA vehicle blueprint whose dimensions and steering configuration most closely resembled the physical platform.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



The selected vehicle blueprint served as the initial approximation from which all dynamic and geometric properties were progressively adapted.

The adaptation process focused on preserving:

- geometric proportions,
- turning behaviour,
- steering response,
- and low-speed manoeuvrability.

The real-world trajectories used to calibrate the CARLA digital-twin parameters (steering range, throttle-torque response, wheelbase, and tire-ground interaction) [are entirely separate from the 20 real-world trajectories used for the validation experiments reported in Chapter 5 and Chapter 7.

### 4.4.2 Wheelbase Adjustment

The wheelbase is one of the most critical parameters affecting vehicle turning dynamics.

The simulated wheelbase was adjusted to match the physical Traxxas platform dimensions:

$$L_{sim} \approx L_{real}$$

where:

- $L_{sim}$  is the simulated wheelbase,
- and  $L_{real}$  is the measured physical wheelbase.

Matching the wheelbase is essential because turning radius depends strongly on the relationship:

$$R = \frac{L}{\tan(\delta)}$$

where:

- $R$  is the turning radius,
- $L$  is the wheelbase,
- and  $\delta$  is the steering angle.

Incorrect wheelbase scaling would therefore produce unrealistic parking trajectories.

### 4.4.3 Steering System Calibration

The steering characteristics of the CARLA vehicle were modified to replicate the servo-controlled steering system of the physical platform.

The steering range was constrained to:

$$\delta \in [-30^\circ, +30^\circ]$$

matching the experimentally calibrated steering limits of the real vehicle.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



The steering curve inside CARLA was also adjusted to improve low-speed steering fidelity. This calibration was necessary because the default CARLA steering response is optimized for full-scale automotive vehicles rather than lightweight RC platforms.

The steering mapping preserves linearity between the normalized control signal and wheel deflection angle.

### 4.4.4 Vehicle Mass and Inertia

Vehicle mass significantly affects acceleration response, turning stability, and momentum during manoeuvring.

The simulated vehicle mass was modified to approximate the measured mass of the physical platform, including:

- onboard computation hardware,
- sensor mounts,
- batteries,
- and additional structural modifications.

Although CARLA does not expose all inertial properties directly, approximating the overall mass distribution improves consistency during transient steering and acceleration phases.

### 4.4.5 Tire Radius and Ground Interaction

The tire radius was rescaled to match the dimensions of the Traxxas wheels.

Tire dimensions influence:

- effective velocity,
- wheel torque transfer,
- and turning behaviour.

Because low-speed parking manoeuvres involve frequent steering reversals and transient lateral motion, accurate tire scaling contributes significantly to dynamic consistency.

However, it is important to note that CARLA cannot perfectly reproduce all tire-ground interaction phenomena present in the real world, particularly:

- surface irregularities,
- local friction variations,
- and micro-sliding effects.

These limitations contribute partially to the remaining sim-to-real discrepancy observed in later experiments.

### 4.4.6 Throttle and Torque Calibration

Longitudinal behaviour was calibrated by adjusting the engine torque curve and throttle response inside CARLA.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Special attention was given to reproducing:

- low-speed acceleration,
- reverse motion behaviour,
- and ESC response characteristics.

Experimentally measured throttle thresholds required to overcome static resistance on the physical vehicle were replicated in simulation.

This calibration ensured that:

- low throttle values produced realistic motion initiation,
- and reverse parking behaviour remained dynamically consistent across domains.

### 4.5 Ultrasonic Sensor Emulation

#### 4.5.1 Motivation

The physical platform relies heavily on ultrasonic sensing during parking manoeuvres. Therefore, preserving equivalent perception behaviour inside the simulator is essential for successful sim-to-real transfer.

Instead of using privileged simulator information such as:

- bird's-eye-view images,
- perfect obstacle coordinates,
- or global occupancy maps,

the digital twin exposes only locally observable distance measurements analogous to those available on the real vehicle.

This design choice increases realism and improves transferability.

#### 4.5.2 Sensor Modelling Approach

Each HC-SR04 ultrasonic sensor is emulated inside CARLA using a single horizontal raycast.

The raycast:

- originates from the sensor mounting location,
- follows the sensor orientation,
- and returns the distance to the first obstacle intersection.

The maximum sensing range is limited to approximately 1 meter to reproduce the operational characteristics of the real sensors.

The simulated measurement is computed as:

$$d_{sensor} = \min(d_{intersection}, d_{max})$$

where:

- $d_{intersection}$  is the detected obstacle distance,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- and  $d_{max}$  is the maximum sensor range.

### 4.5.3 Sensor Placement Replication

The raycast sensors were positioned inside CARLA to match the geometric locations of the physical HC-SR04 sensors.

The configuration includes:

- three left-side sensors,
- three right-side sensors,
- one front sensor,
- and one rear sensor.

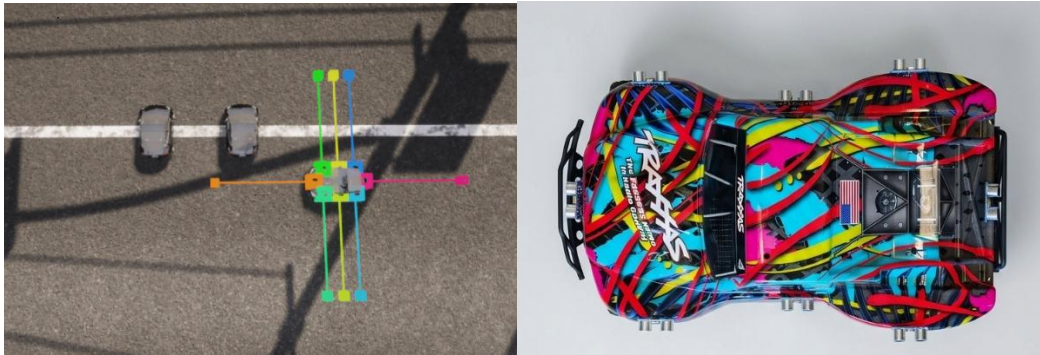


Figure 3, simulated digital twin in the CARLA simulator environment and the physical model.

Preserving sensor placement consistency is critical because parking policies depend heavily on relative obstacle geometry.

### 4.5.4 Sensor Limitations in Simulation

Although the raycasting approach provides computational simplicity and deterministic behaviour, it cannot fully reproduce all physical ultrasonic phenomena.

The simulation does not model:

- multipath reflections,
- sensor interference,
- material-dependent reflections,
- or acoustic noise.

Consequently, the simulated sensor measurements are generally cleaner and more stable than real-world ultrasonic data.

As later experimental results demonstrate, this perception discrepancy constitutes one of the primary remaining contributors to the residual reality gap.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 4.6 Parking Environment Modelling

#### 4.6.1 Parking Scenario Design

The parking environment was designed to reproduce a constrained reverse perpendicular parking manoeuvre.

The environment includes:

- the ego vehicle,
- two parked non-player vehicles (NPVs),
- and a static rear boundary.

The parking slot dimensions were selected to allow feasible but nontrivial parking trajectories.

#### 4.6.2 Non-Player Vehicles

The parked vehicles serve as obstacles defining the parking slot geometry.

Their positions remain fixed across all validation experiments to ensure:

- repeatability,
- deterministic comparisons,
- and trajectory consistency.



Figure 4, Non-player vehicles position in the simulation

#### 4.6.3 Environment Determinism

To minimize variability during validation and reinforcement-learning experiments:

- weather conditions were fixed,
- traffic was disabled,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- pedestrian actors were removed,
- and environmental randomness was minimized.

This deterministic configuration ensures that observed discrepancies originate primarily from the simulation-reality mismatch rather than uncontrolled environmental variability.

### 4.7 Synchronization and Timing Design

#### 4.7.1 Synchronous Simulation Mode

CARLA was configured to operate in synchronous mode.

In synchronous execution:

- the simulator advances only when explicitly triggered,
- ensuring deterministic physics updates and reproducible trajectories.

This configuration is particularly important for reinforcement learning and sim-to-real validation because asynchronous timing may introduce nondeterministic behaviour.

#### 4.7.2 Physics Timestep

The physics engine was configured with a fixed timestep of:

$$\Delta t_{physics} = 10 \text{ ms}$$

A fixed timestep improves:

- numerical stability,
- reproducibility,
- and temporal consistency.

#### 4.7.3 Control Frequency Matching

The high-level controller operates at the same frequency in both simulation and reality.

Control updates occur every:

$$\Delta t_{control} = 50 \text{ ms}$$

corresponding to a control frequency of:

$$f_{control} = 20 \text{ Hz}$$

The simulator therefore executes five physics updates between consecutive control actions.

#### 4.7.4 Importance of Timing Consistency

Timing consistency is critical in autonomous control systems because even small latency mismatches may alter:

- steering trajectories,
- acceleration response,
- and sensor-feedback timing.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



By synchronizing:

- sensor updates,
- control execution,
- and physics integration,

the proposed framework minimizes temporal contributions to the reality gap.

### 4.8 Summary

This chapter presented the development of the CARLA-based digital twin used throughout this thesis. The twin was designed to reproduce the geometry, dynamics, sensing structure, and timing behaviour of the physical 1:10-scale autonomous vehicle platform while maintaining a shared continuous action space across simulation and reality.

The vehicle blueprint was adapted through modifications to:

- wheelbase,
- steering limits,
- mass properties,
- tire dimensions,
- and throttle response.

The perception layer was reproduced using raycast-based ultrasonic sensor emulation positioned identically to the real sensor configuration. Additionally, the simulator was configured for deterministic synchronous execution with timing parameters aligned to the physical platform.

The resulting digital twin provides the foundation for the validation methodology presented in the following chapter, where its fidelity is quantitatively evaluated through direct comparison between simulated and real parking manoeuvres.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### Chapter 5 - Digital Twin Validation Methodology

#### 5.1 Introduction

The usefulness of a digital twin for autonomous-driving research depends not only on its construction, but also on its ability to reproduce the behaviour of the physical platform under identical operating conditions. A simulator may visually resemble the real environment while still failing to reproduce the dynamic response, sensor behaviour, or timing characteristics required for reliable controller transfer. Consequently, before employing the proposed CARLA-based twin for reinforcement-learning training, it is necessary to quantitatively evaluate its fidelity with respect to the real vehicle.

Validation represents a critical stage in sim-to-real research because it determines whether the simulator captures the aspects of system behaviour that are relevant for autonomous control. In the context of automated parking, small discrepancies in steering dynamics, acceleration response, or obstacle perception may accumulate over time and lead to significant trajectory deviations. Therefore, evaluating the digital twin requires more than observing task completion; it requires detailed signal-level analysis of the vehicle behaviour during manoeuvre execution.

A common limitation in many sim-to-real studies is that validation is performed indirectly through controller performance alone. In such approaches, if a controller trained in simulation performs well in reality, the simulator is assumed to be accurate. However, this methodology does not isolate the actual source of discrepancies because performance differences may originate from the controller architecture, learning instability, exploration strategy, or sensor uncertainty rather than from the simulator itself.

To address this issue, the proposed framework adopts a deterministic validation methodology in which the same scripted parking controller is executed in both simulation and reality. By fixing the decision logic and exposing identical control commands to both systems, any observed discrepancy can be attributed directly to the remaining simulation-reality mismatch rather than to policy variability.

The validation process focuses specifically on low-speed reverse perpendicular parking because this manoeuvre is highly sensitive to:

- steering consistency,
- acceleration response,
- and short-range obstacle perception.

Three measurable vehicle-state channels are analysed:

- yaw angle,
- lateral acceleration,
- and lateral offset.

These signals are compared between the physical vehicle and the digital twin using quantitative similarity metrics designed to evaluate both signal shape and temporal alignment.

This chapter presents the complete validation methodology employed to assess the fidelity of the proposed digital twin. The chapter describes:

- the deterministic reference controller,
- the experimental setup,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- the data-acquisition procedure,
- the signal-processing pipeline,
- and the statistical metrics used for quantitative comparison.

### 5.2 Validation Objectives

The primary objective of the validation process is to determine whether the CARLA-based digital twin reproduces the behaviour of the physical vehicle with sufficient fidelity to support sim-to-real controller transfer.

More specifically, the validation seeks to answer the following questions:

1. Does the simulated vehicle reproduce the steering behaviour of the physical platform during parking manoeuvres?
2. Does the simulator accurately capture transient lateral dynamics during steering reversals?
3. Are the simulated and real trajectories geometrically consistent?
4. Which aspects of the vehicle behaviour contribute most significantly to the residual reality gap?
5. Is the digital twin sufficiently accurate for reinforcement-learning policy deployment?

To answer these questions, the validation methodology emphasizes:

- deterministic experimentation,
- repeatable trajectories,
- synchronized data collection,
- and quantitative signal comparison.

Rather than evaluating only the final parking success, the methodology analyses the evolution of the manoeuvre over time, allowing identification of localized discrepancies during specific driving phases.

### 5.3 Deterministic Reference Policy

#### 5.3.1 Motivation for a Scripted Controller

A deterministic reference controller is employed to isolate the simulation-reality discrepancy from learning-related variability.

Using a reinforcement-learning policy during validation would introduce stochastic behaviour and potentially obscure whether observed differences originate from:

- the simulator,
- the policy itself,
- or the learning process.

Therefore, the validation employs a scripted state-machine controller that generates identical action sequences under equivalent sensor conditions.

This approach ensures that both domains execute the same control logic under equivalent sensor conditions. It is important to note, however, that this does not fully isolate plant (dynamics) mismatch



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



from perception mismatch: because state transitions in the scripted controller (5.3.2) are triggered by ultrasonic sensor readings, any discrepancy between simulated and real ultrasonic measurements can cause the controller to transition between parking phases at different times in each domain. The two vehicles would then be executing the same **control logic** but not necessarily the same **timestamped input sequence**, so a portion of the observed trajectory discrepancy may originate from sensing mismatch rather than from vehicle-dynamics mismatch alone. This validation therefore constitutes a **closed-loop system validation** (5.3.1a): it evaluates the combined behavioural similarity of sensing, control logic, and dynamics together, rather than isolating plant dynamics under identical inputs. This distinction and its implications are discussed further below.

### 5.3.2 Open-Loop vs. Closed-Loop Validation

Two complementary validation designs can be distinguished:

1. **Open-loop plant validation:** the exact same timestamped steering and throttle command sequence, recorded once, is replayed open-loop in both simulation and reality, with no dependency on live sensor feedback in either domain. This isolates dynamic (plant) fidelity under strictly identical inputs, since any resulting trajectory difference can only be attributed to mismatch in vehicle dynamics, actuation, or ground interaction.
2. **Closed-loop system validation:** the same control logic (e.g., the scripted state machine of 5.3.2) is executed independently in each domain, using each domain's own live sensor readings to drive state transitions. This evaluates the behavioural similarity of the complete closed-loop system but a trajectory discrepancy here can arise from any of these three sources, not from dynamics alone.

The validation presented in this thesis follows the **closed-loop** design: the scripted controller (5.3.2) reacts to each domain's own ultrasonic measurements. This is a deliberate and useful evaluation, since it reflects the same sensing/control coupling the deployed RL policy will experience, but it should not be interpreted as isolating plant dynamics.

### 5.3.3 State-Machine Architecture

The reference controller is implemented as a finite-state machine operating over the shared continuous action space introduced previously.

The control vector is defined as:

$$a_t = [u_{throttle}, u_{steer}, u_{brake}]$$

where:

- $u_{throttle}$  controls longitudinal motion,
- $u_{steer}$  controls steering,
- and  $u_{brake}$  controls braking behaviour.

State transitions are triggered by ultrasonic sensor measurements and predefined geometric conditions.

The controller consists of three sequential parking phases:

1. slot-search,
2. approach-alignment,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



3. reverse parking.

### 5.3.4 Slot-Search Phase

During the slot-search phase, the ego vehicle moves forward at low constant speed while monitoring the ultrasonic sensors positioned on the parking-side flank.

The objective of this phase is to detect the presence of a sufficiently large parking slot between two parked vehicles.

The vehicle maintains approximately straight-line motion using near-zero steering commands:

$$u_{steer} \approx 0$$

while applying low forward throttle.

This phase is important for evaluating:

- straight-line dynamic consistency,
- steering stability,
- and low-speed longitudinal behaviour.

### 5.3.5 Approach-Alignment Phase

Once the parking slot is detected, the controller initiates an alignment manoeuvre designed to position the vehicle appropriately for reverse entry into the slot.

The vehicle steers slightly away from the parking space while continuing forward motion. This creates sufficient geometric clearance for the subsequent reverse trajectory.

This phase introduces moderate steering dynamics and transient lateral acceleration, making it useful for evaluating:

- steering response,
- yaw evolution,
- and lateral trajectory consistency.

### 5.3.6 Reverse Parking Phase

The reverse-parking phase constitutes the most dynamically demanding portion of the manoeuvre.

The vehicle:

- switches to reverse motion,
- applies steering toward the parking slot,
- and progressively aligns itself within the parking boundaries.

This phase contains:

- steering reversals,
- transient lateral motion,
- and close-range obstacle interactions.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Consequently, it represents the most sensitive stage for sim-to-real discrepancies.

Any mismatch in:

- steering dynamics,
- acceleration response,
- or ultrasonic perception

becomes particularly observable during this phase.

### 5.4 Experimental Setup

#### 5.4.1 Parking Environment

The validation experiments are conducted using equivalent parking scenarios in simulation and reality.

The environment consists of:

- the ego vehicle,
- two parked obstacle vehicles,
- and a rear boundary representing the roadside limit.

The parking slot dimensions are approximately:

$$L_{slot} \approx 1.6L_{vehicle}$$

ensuring that the manoeuvre remains feasible while still requiring precise control.

#### 5.4.2 Initial Conditions

To ensure reproducibility, all experimental runs begin from identical initial conditions.

Both the simulated and physical vehicles are initialized with:

- the same starting pose,
- identical orientation,
- equivalent slot geometry,
- and fixed obstacle positions.

This consistency minimizes external variability and ensures that observed differences originate primarily from the sim-to-real discrepancy.

#### 5.4.3 Number of Experimental Runs

A total of:

- 20 simulation runs,
- and 20 real-world runs

were recorded for validation.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Each simulation run is paired with a corresponding real-world run using equivalent initialization conditions.

The resulting dataset contains:

- 40 parking trajectories,
- synchronized state measurements,
- and temporally aligned sensor data.

### 5.4.4 Data Acquisition

Vehicle-state data are recorded throughout the complete parking manoeuvre.

The collected signals include:

- yaw angle,
- lateral acceleration,
- lateral offset,
- steering commands,
- throttle commands,
- and ultrasonic measurements.

Data are sampled at:

$$T_s = 0.05 \text{ s}$$

corresponding to a sampling frequency of:

$$f_s = 20 \text{ Hz}$$

The complete observation horizon consists of:

- 213 temporal samples,
- corresponding to approximately 10.65 seconds of manoeuvre execution.

## 5.5 Validation Metrics

### 5.5.1 Motivation for Statistical Metrics

Evaluating digital-twin fidelity requires quantitative metrics capable of comparing temporal signals generated in simulation and reality.

Simple trajectory visualization alone is insufficient because:

- visually similar trajectories may contain temporal misalignment,
- signal offsets,
- or amplitude discrepancies.

Therefore, the validation employs two complementary statistical similarity measures:



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



1. Pearson Correlation Coefficient (PCC),
2. Maximum Normalized Cross-Correlation (MNCC).

Together, these metrics evaluate:

- signal shape similarity,
- temporal alignment,
- and amplitude consistency.

### 5.5.2 Pearson Correlation Coefficient (PCC)

The PCC evaluates the linear relationship between two temporal signals.

It is defined as:

$$\rho(X, Y) = \frac{E[(X - \mu_x)(Y - \mu_y)]}{\sigma_x \sigma_y}$$

where  $X$  and  $Y$  are the compared signals,  $\mu_x$  and  $\mu_y$  their means, and  $\sigma_x$  and  $\sigma_y$  their standard deviations.

The PCC ranges between **-1 and +1**: a value of +1 indicates perfect positive linear correlation, -1 indicates perfect negative linear correlation, and 0 indicates no linear correlation. In this thesis, PCC values are consistently positive and close to 1, indicating that the simulated and real signals evolve with similar temporal shape.

It is important to note that **PCC measures the similarity of temporal shape, not numerical agreement**. Because PCC is invariant to affine transformations of either signal, two signals can achieve  $PCC = 1$  while differing substantially in scale (amplitude) or offset (bias). A high PCC therefore confirms that the simulated and real signals rise and fall together, but does not by itself confirm that their magnitudes match. Amplitude agreement is assessed separately, as described in 5.5.3 and 5.7.

### 5.5.3 Maximum Normalized Cross-Correlation (MNCC)

To complement PCC, this thesis also computes the Maximum Normalized Cross-Correlation, which additionally captures temporal (lag) misalignment between the two signals. For two discrete signals  $X$  and  $Y$  of length  $n$ , the cross-correlation at integer lag  $\tau$  is defined as:

$$C(\tau) = \sum_t X(t) \cdot Y(t + \tau)$$

The normalized cross-correlation at lag  $\tau$  is then:

$$\beta(\tau) = C(\tau) / \sqrt{(C_{XX}(0) \cdot C_{YY}(0))}$$

where  $C_{XX}(0)$  and  $C_{YY}(0)$  are the zero-lag autocorrelations (i.e., the energy) of  $X$  and  $Y$  respectively. The MNCC is defined as the maximum of  $\beta(\tau)$  over a bounded search range of lags  $\tau \in [-\tau_{\max}, +\tau_{\max}]$ :

$$MNCC = \max_{\tau} \beta(\tau), \tau_{\text{opt}} = \operatorname{argmax}_{\tau} \beta(\tau)$$

where  $\tau_{\max}$  = samples at the 20 Hz sampling rate used in this study. The optimal lag  $\tau_{\text{opt}}$  at which the maximum occurs is recorded and reported alongside the MNCC value for every channel and every run, since an unreported lag could otherwise conceal a genuine timing mismatch between simulation and reality rather than measuring it. Because MNCC is normalized by the two signals' own energies



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



rather than by their standard deviations about the mean, it remains sensitive to scale and offset differences that PCC does not detect, provided it is computed on signals that have not already been normalized to remove those differences (see 5.7).

### 5.5.4 Evaluated Signals

The validation focuses on three key vehicle-state channels:

#### Yaw Angle

Yaw describes the vehicle orientation during the manoeuvre.

It is particularly important during reverse parking because:

- steering-induced heading changes dominate the trajectory evolution.

High yaw consistency indicates accurate steering reproduction.

#### Lateral Acceleration

Lateral acceleration reflects transient steering dynamics and tire-ground interaction.

It is highly sensitive to:

- steering reversals,
- inertia mismatch,
- and dynamic modelling errors.

Consequently, lateral acceleration constitutes one of the most difficult channels to reproduce accurately.

#### Lateral Offset

Lateral offset evaluates geometric trajectory consistency relative to the parking slot.

It measures whether the simulated and physical vehicles follow comparable spatial paths during manoeuvre execution.

## 5.6 Uniform and Non-Uniform Sampling

### 5.6.1 Motivation

The distribution of evaluation samples along the trajectory may influence the measured similarity metrics.

Parking manoeuvres contain:

- dynamically inactive intervals,
- and transient steering events.

Uniform sampling treats all temporal regions equally, whereas non-uniform sampling emphasizes transient phases.

### 5.6.2 Uniform Sampling

Under uniform sampling, all trajectory samples are distributed evenly across the observation horizon.

This configuration evaluates global trajectory consistency.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 5.6.3 Non-Uniform Sampling

The non-uniform configuration concentrates evaluation points near the beginning of the trajectory using dyadic spacing.

This strategy emphasizes:

- transient steering behaviour,
- rapid dynamic changes,
- and initial manoeuvre evolution.

The approach follows methodologies proposed in previous digital-twin validation literature.

### 5.7 Signal Alignment and Processing

Before metric computation, the simulation and real-world signals are temporally aligned.

Alignment begins at the first control instant satisfying:

$$u_{throttle} > 0$$

ensuring synchronization at the beginning of active vehicle motion.

Before metric computation, the simulation and real-world signals are temporally aligned by identifying the first control instant satisfying  $u_{throttle} > 0$  in both domains, ensuring synchronization at the onset of active vehicle motion.

Two versions of each signal pair are then prepared for the two metrics, since normalization and amplitude assessment are mutually exclusive on the same data:

- For **PCC**, signals are used as recorded (no amplitude normalization), since PCC is already scale- and offset-invariant by construction; this preserves the interpretation of PCC as a measure of shape similarity only.
- For **MNCC**, signals are likewise used in their raw (unnormalized) amplitude form, since normalizing amplitude before computing MNCC would remove exactly the scale differences MNCC is intended to detect. The signals are only time-segmented according to the parking phases (5.8), not rescaled.

This preprocessing ensures that the amplitude-sensitivity of MNCC is preserved and that neither metric's stated purpose is undermined by the preprocessing applied to it.

### 5.8 Importance of Phase-Based Validation

Rather than evaluating the manoeuvre globally, the validation analyses each parking phase independently.

This approach allows identification of localized discrepancies that may otherwise remain hidden in aggregate metrics.

For example:

- straight-line motion may exhibit near-perfect agreement,
- while reverse steering may reveal dynamic mismatch.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Phase-based analysis therefore provides deeper insight into:

- the strengths of the digital twin,
- and the primary contributors to the residual reality gap.

### 5.9 Summary

This chapter presented the methodology used to validate the proposed CARLA-based digital twin against the physical autonomous vehicle platform.

The validation framework employs:

- a deterministic state-machine parking controller,
- synchronized experimental conditions,
- quantitative statistical metrics,
- and phase-based signal analysis.

The methodology evaluates the similarity between simulation and reality through comparison of:

- yaw angle,
- lateral acceleration,
- and lateral offset

using PCC and MNCC metrics under both uniform and non-uniform sampling configurations.

By isolating the simulator from controller-learning variability, the proposed validation process provides a rigorous framework for identifying the remaining sim-to-real discrepancy prior to reinforcement-learning deployment.

The next chapter presents the reinforcement-learning framework used to train and deploy a PPO-based autonomous parking controller using the validated digital twin.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### Chapter 6 - Reinforcement Learning Controller

#### 6.1 Introduction

After validating the fidelity of the CARLA-based digital twin, the next stage of the proposed framework focuses on the development of an autonomous parking controller based on Deep Reinforcement Learning (DRL). The primary objective of this controller is to learn a reverse perpendicular parking manoeuvre entirely in simulation and subsequently transfer the learned behaviour directly to the physical vehicle without additional fine-tuning.

Traditional autonomous parking systems typically rely on modular architectures composed of:

- parking-slot detection,
- trajectory planning,
- and trajectory tracking.

Although these methods can achieve reliable performance in structured environments, they often require:

- extensive model tuning,
- handcrafted heuristics,
- and explicit geometric formulations.

Moreover, such pipelines may become increasingly complex when operating under uncertain sensing conditions or dynamically varying vehicle behaviour.

Reinforcement learning offers an alternative paradigm in which the autonomous agent learns the control policy directly through interaction with the environment. Instead of explicitly defining a parking trajectory, the agent progressively discovers an optimal sequence of steering and throttle actions that maximize a cumulative reward function.

Among available DRL algorithms, Proximal Policy Optimization (PPO) has emerged as one of the most widely adopted methods for continuous-control autonomous-driving applications due to:

- training stability,
- robustness,
- sample efficiency,
- and implementation simplicity.

In this thesis, PPO is employed to learn a continuous parking-control policy operating over the shared action space introduced in previous chapters. The policy receives observations derived exclusively from onboard-equivalent sensing and generates continuous steering and throttle commands that directly control the vehicle.

A key characteristic of the proposed approach is that the reinforcement-learning agent is intentionally restricted to realistic perception inputs. Unlike many simulation-only parking studies that rely on privileged simulator information such as:

- global coordinates,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- occupancy maps,
- or bird's-eye-view images,

the proposed framework exposes only locally observable measurements analogous to those available on the physical platform. This design improves transferability and prevents the policy from exploiting simulator-specific information unavailable during real-world deployment.

This chapter presents the complete reinforcement-learning framework used in this thesis, including:

- the Markov Decision Process (MDP) formulation,
- observation-space design,
- continuous action-space representation,
- reward-function engineering,
- PPO configuration,
- training methodology,
- and zero-shot deployment strategy.

### 6.2 Reinforcement Learning Formulation

#### 6.2.1 Markov Decision Process Representation

The autonomous parking problem is formulated as a Markov Decision Process (MDP).

An MDP is defined by the tuple:

$$\mathcal{M} = (S, A, P, R, \gamma)$$

where:

- $S$  represents the state space,
- $A$  represents the action space,
- $P$  denotes the state-transition probability distribution,
- $R$  denotes the reward function,
- and  $\gamma$  is the discount factor.

At each timestep  $t$ , the agent:

1. observes the current state  $s_t$ ,
2. selects an action  $a_t$ ,
3. receives reward  $r_t$ ,
4. and transitions to the next state  $s_{t+1}$ .

The objective of the learning process is to determine an optimal policy:

$$\pi^*(a | s)$$



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



that maximizes the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right]$$

The parking task constitutes a continuous-control problem because both steering and throttle commands vary continuously over time.

### 6.2.2 Episodic Learning Structure

Training is performed episodically.

Each episode begins with:

- randomized initial vehicle placement,
- randomized parking-slot geometry,
- and reset environmental conditions.

An episode terminates when one of the following conditions occurs:

- successful parking completion,
- collision,
- excessive deviation,
- or maximum timestep exhaustion.

The episodic structure enables repeated exploration of parking trajectories under varying conditions, improving policy generalization.

### 6.3 Observation Space Design

#### 6.3.1 Design Philosophy

The observation space was designed according to two primary objectives:

1. preserving compatibility between simulation and reality,
2. minimizing dependence on privileged simulator information.

Rather than relying on perfect state estimation or externally tracked coordinates, the agent receives only information obtainable from onboard sensing and internal vehicle-state estimation.

This design choice improves realism and increases the likelihood of successful sim-to-real transfer.

#### 6.3.2 Ultrasonic Sensor Observations

The primary perception inputs are the eight ultrasonic distance measurements.

The observation vector includes:

$$o_{ultra} = [d_1, d_2, \dots, d_8, \psi, ax, ay]$$

where:

- $d_i$  represents the distance measured by sensor  $i$ .



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- $\psi$ , represents the Yaw,
- $a_x$ , represents the acceleration in x-axis,
- And  $a_y$ , represents the acceleration in y-axis.

These measurements provide local geometric awareness regarding:

- nearby parked vehicles,
- roadside boundaries,
- and slot occupancy.

The sensor values are normalized prior to policy input to improve training stability.

### 6.3.3 Vehicle Orientation

The observation vector also includes vehicle heading information represented through the yaw angle:

$$\psi$$

Yaw information is critical because successful parking requires accurate orientation alignment relative to the parking slot.

The heading representation allows the policy to:

- estimate rotational deviation,
- coordinate steering corrections,
- and stabilize final alignment.

### 6.3.4 Lateral Acceleration

To improve awareness of transient dynamic behaviour, the observation space includes lateral acceleration measurements:

$$a_y$$

Lateral acceleration provides indirect information regarding:

- steering aggressiveness,
- turning intensity,
- and transient dynamic response.

Including dynamic-state information helps the policy adapt its steering behaviour during reverse manoeuvring.

### 6.3.5 Final Observation Vector

The complete observation vector is therefore defined as:

$$o_t = [d_1, d_2, \dots, d_8, \psi, a_y]$$

This representation preserves compatibility between:

- the CARLA digital twin,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- and the physical autonomous vehicle.

### 6.4 Continuous Action Space

#### 6.4.1 Motivation for Continuous Control

Parking manoeuvres require smooth and precise low-speed control. Discrete-action formulations often produce abrupt steering changes and oscillatory behaviour that are undesirable in physical systems.

Therefore, the proposed framework adopts a continuous action space.

The action vector is defined as:

$$a_t = [u_{throttle}, u_{steer}, u_{brake}]$$

where:

- $u_{throttle} \in [-1, 1]$ ,
- $u_{steer} \in [-1, 1]$ ,
- and  $u_{brake} \in [0, 1]$ .

#### 6.4.2 Steering Control

The steering action controls the front-wheel steering angle through linear mapping:

$$\delta = u_{steer} \cdot \delta_{max}$$

where:

- $\delta_{max}$  represents the maximum steering angle.

Continuous steering improves:

- trajectory smoothness,
- parking precision,
- and transferability to physical hardware.

#### 6.4.3 Longitudinal Control

Throttle commands regulate forward motion, reverse motion, and stopping behaviour through the signed value  $u_{throttle} \in [-1, 1]$ , where positive values command forward motion and negative values command reverse motion. The separate braking command  $u_{brake} \in [0, 1]$  applies additional deceleration independent of throttle direction.

Because these two commands are generated independently by the policy's output distribution, a mapping rule is required to resolve physically contradictory combinations. This rule is applied identically in both the CARLA digital twin and the physical vehicle's low-level Arduino controller (3.4.2), ensuring that the action-space semantics remain consistent across domains.

### 6.5 Reward Function Design

#### 6.5.1 Importance of Reward Engineering

The reward function plays a central role in reinforcement learning because it defines the optimization objective guiding policy behaviour.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Poorly designed rewards may lead to:

- unstable learning,
- unintended behaviours,
- local minima,
- or reward exploitation.

The parking reward function was therefore designed to:

- encourage successful parking completion,
- minimize collisions,
- improve alignment quality,
- and promote stable trajectories.

### 6.5.2 Overall Reward Structure

The reward function is composed of multiple weighted components:

$$r = r_{success} + r_{distance} + r_{heading} - P_{collision}$$

where:

- $r_{success}$  rewards successful parking,
- $r_{distance}$  encourages positional improvement,
- $r_{heading}$  rewards orientation alignment,
- and  $P_{collision}$  penalizes collisions.

It is important to distinguish between information available to the policy and information available for reward computation. The observation space (6.3) deliberately excludes privileged simulator state to preserve compatibility with real-world deployment, where such information is unavailable. The reward function, by contrast, is computed only during simulation-based training and is never evaluated at deployment time; it is therefore permitted to use privileged simulator state purely as training-time supervision signals. This is a standard and valid design pattern in sim-to-real reinforcement learning: privileged state may shape the reward without being observed by the policy, since the reward is not needed once the policy is deployed. The distance and heading reward terms in this thesis follow exactly this pattern.

### 6.5.3 Success Reward

A large terminal reward is applied when the vehicle successfully parks within the designated slot boundaries.

Successful parking requires:

- acceptable positional alignment,
- acceptable orientation error,
- and collision-free execution.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



The terminal success reward is defined as:

$$r_{success} = R_s$$

where  $R_s$  is a large positive constant.

This reward encourages the policy to prioritize manoeuvre completion.

### 6.5.4 Distance-Based Reward

The distance component encourages the vehicle to reduce positional error relative to the target parking pose.

The reward is proportional to the reduction in Euclidean distance:

$$r_{distance} = -k_d \| p_t - p_{goal} \|$$

where:

- $p_t$  is the current position,
- $p_{goal}$  is the target parking pose,
- and  $k_d$  is a weighting coefficient.

This term guides exploration toward the parking slot.

### 6.5.5 Heading Alignment Reward

Correct orientation is critical for realistic parking behaviour.

The heading reward penalizes angular deviation:

$$r_{heading} = -k_h |\psi - \psi_{goal}|$$

where:

- $\psi$  is the current yaw angle,
- and  $\psi_{goal}$  is the desired parking orientation.

This component encourages smooth alignment during reverse entry.

### 6.5.6 Collision Penalty

Collisions terminate the episode immediately and generate a large negative reward.

The penalty is defined as:

$$P_{collision} = C$$

where  $C$  is a large positive constant.

Strong collision penalties are necessary because parking occurs in highly constrained environments.

## 6.6 PPO Algorithm

### 6.6.1 Motivation for PPO

Proximal Policy Optimization (PPO) was selected because it provides:



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- stable policy updates,
- strong continuous-control performance,
- implementation simplicity,
- and robustness across robotics applications.

Unlike value-based reinforcement-learning methods, PPO directly optimizes the policy distribution.

### 6.6.2 Policy Optimization Objective

PPO maximizes the clipped surrogate objective:

$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)]$$

where:

- $r_t(\theta)$  is the probability ratio,
- $\hat{A}_t$  is the estimated advantage,
- and  $\epsilon$  is the clipping threshold.

The clipping mechanism prevents excessively large policy updates and improves training stability.

### 6.6.3 Actor-Critic Structure

The PPO implementation employs an actor-critic architecture consisting of:

- a policy network (actor),
- and a value-estimation network (critic).

The actor outputs continuous action distributions, while the critic estimates the expected future return.

### 6.6.4 Implementation and Reproducibility Details

The PPO implementation is based on Stable-Baselines3, using default algorithm settings with the following configuration:

Table 6.1: PPO Hyperparameters

| Parameter                         | value         |
|-----------------------------------|---------------|
| Learning rate                     | [ 3.0e-4 ]    |
| Discount factor ( $\gamma$ )      | [ 0.99 ]      |
| GAE $\lambda$                     | [ 0.95 ]      |
| Clipping threshold ( $\epsilon$ ) | [ 0.20 ]      |
| Batch size                        | [ 64 ]        |
| Rollout length (steps per update) | [ 2048 ]      |
| Total training timesteps          | [ 2,000,000 ] |
| Maximum steps per episode         | [ 512 ]       |

The observation space is an 11-dimensional vector concatenating the eight normalized ultrasonic ranges, the yaw angle, and the two-axis (longitudinal and lateral) acceleration:  $\mathbf{o}_t = [d_1, \dots, d_8, \psi, a_x,$



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



$a_y$ ]. The action space is unchanged from the scripted reference policy (6.4): throttle  $\in [-1, 1]$ , steering  $\in [-1, 1]$ , brake  $\in \{0, 1\}$ .

Due to the computational cost of training with a physics-based simulator, results are reported for a single training run; variance across seeds was not evaluated.

### 6.7 Neural Network Architecture

#### 6.7.1 Input Layer

The neural network receives the observation vector:

$$o_t = [d_1, d_2, \dots, d_8, \psi, a_y]$$

as input.

#### 6.7.2 Hidden Layers

The policy network consists of fully connected layers with nonlinear activation functions.

The architecture was intentionally kept lightweight to:

- reduce onboard inference latency,
- improve deployment efficiency,
- and maintain real-time execution capability on the Jetson Orin Nano.

#### 6.7.3 Output Layer

The output layer predicts:

- steering commands,
- throttle values,
- and braking actions.

Continuous Gaussian action distributions are used during training to encourage exploration.

### 6.8 Training Procedure

#### 6.8.1 Simulation-Only Training

All PPO training is performed exclusively inside the CARLA digital twin.

No real-world data are used during policy optimization.

This setup evaluates whether the validated simulator alone is sufficient for transferable policy learning.

#### 6.8.2 Domain Randomization

To improve robustness, several environmental properties are randomized during training, including:

- initial vehicle pose,
- parking-slot geometry,
- and minor sensor noise perturbations.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



Domain randomization prevents overfitting to a single deterministic trajectory.

### 6.8.3 Episode Reset Strategy

Episodes are reset whenever:

- the vehicle collides,
- leaves the valid operating area,
- or exceeds the maximum timestep horizon.

This strategy improves exploration efficiency and stabilizes training.

## 6.9 Zero-Shot Sim-to-Real Deployment

### 6.9.1 Deployment Objective

After training convergence, the learned PPO policy is deployed directly onto the physical vehicle without:

- retraining,
- fine-tuning,
- or additional adaptation.

This procedure is referred to as zero-shot sim-to-real transfer.

### 6.9.2 Real-Time Inference Pipeline

During deployment:

1. sensor data are collected onboard,
2. observations are constructed,
3. the PPO network performs inference,
4. and continuous actions are transmitted to the actuators.

The policy executes entirely onboard the NVIDIA Jetson Orin Nano.

### 6.9.3 Importance of Zero-Shot Evaluation

Zero-shot deployment constitutes one of the strongest evaluations of digital-twin fidelity because the controller is never exposed to real-world adaptation data.

Successful transfer therefore indicates that:

- the simulated dynamics,
- sensing behaviour,
- and timing structure

are sufficiently representative of the physical platform.



## **Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle**



### **6.10 Summary**

This chapter presented the reinforcement-learning framework developed for autonomous parking using the validated CARLA digital twin.

The parking problem was formulated as a continuous-control Markov Decision Process using realistic onboard-equivalent observations derived from:

- ultrasonic sensing,
- yaw estimation,
- and lateral acceleration measurements.

A PPO-based controller was designed using a shared continuous action space compatible with both simulation and reality. The reward function was engineered to encourage:

- successful parking,
- positional accuracy,
- orientation alignment,
- and collision avoidance.

Training was performed entirely in simulation using the CARLA digital twin, after which the learned policy was deployed directly onto the physical platform without fine-tuning.

The following chapter presents the experimental results obtained from:

- digital-twin validation,
- PPO training,
- and zero-shot sim-to-real deployment.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### Chapter 7 - Experimental Results and Discussion

#### 7.1 Introduction

This chapter presents the experimental evaluation of the proposed digital twin framework and the PPO-based autonomous parking system. The objective of the experiments is twofold. First, the fidelity of the developed CARLA digital twin is quantitatively validated by comparing its behaviour with that of the physical 1:10-scale autonomous vehicle under identical parking scenarios. Second, the performance of the reinforcement learning controller is evaluated in both simulation and real-world environments to assess the effectiveness of zero-shot simulation-to-reality transfer.

For the purposes of this thesis, a parking attempt is classified as **successful** if the vehicle comes to rest with a lateral offset below 0.05 m and a heading error below  $10^\circ$  relative to the target parking pose, without collision, within 7.02 seconds of manoeuvre initiation. This criterion is applied consistently to both the digital-twin fidelity evaluation (7.2) and the reinforcement-learning deployment evaluation (7.8).

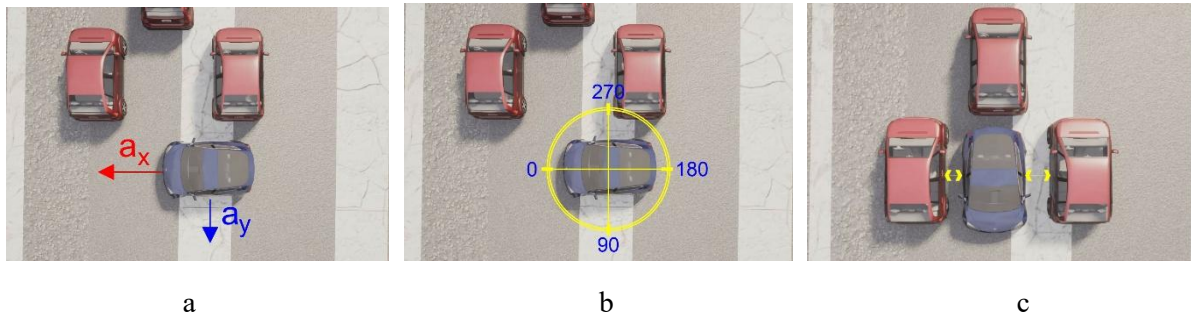
Unlike many previous studies that evaluate autonomous parking primarily through qualitative demonstrations, this work employs a comprehensive set of quantitative performance metrics, including trajectory error, parking success rate, parking completion time, steering smoothness, collision rate, and statistical measures such as mean, standard deviation, root mean square error (RMSE), and confidence intervals. These metrics provide objective evidence regarding the accuracy of the digital twin and the robustness of the learned parking policy.

The experimental results presented in this chapter demonstrate that the proposed digital twin accurately reproduces the behaviour of the physical vehicle and provides an effective training environment for reinforcement learning-based autonomous parking.

#### 7.2 Digital-Twin Fidelity Validation, Deterministic Scripted Controller

This section reports agreement between simulation and reality when the same scripted state-machine controller (5.3) is executed in both domains. The controller consists of three phases:

- (a) slot-search
- (b) approach-alignment
- (c) and reverse-parking



Each triggered by ultrasonic threshold crossings. Twenty runs were recorded on each platform (40 total), yielding trajectories of 213 samples at a 0.05 s sampling period (10.65 s observation horizon), aligned at the first control step where throttle exceeds zero. Fidelity was evaluated using PCC and MNCC under both uniform sampling and non-uniform (dyadic, front-loaded) sampling, since concentrating



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



evaluation points during transient phases can reveal timing- and shape-mismatch that uniform sampling may average out. This evaluates digital-twin fidelity, not reinforcement-learning policy performance; PPO-based results are reported separately in 7.8.

### 7.3 Yaw-Angle Validation

#### 7.3.1 Yaw Trajectory Analysis

Yaw angle represents one of the most important indicators of steering consistency during parking manoeuvres. Figure 5 illustrates the yaw trajectories obtained in simulation and reality during reverse parking execution.

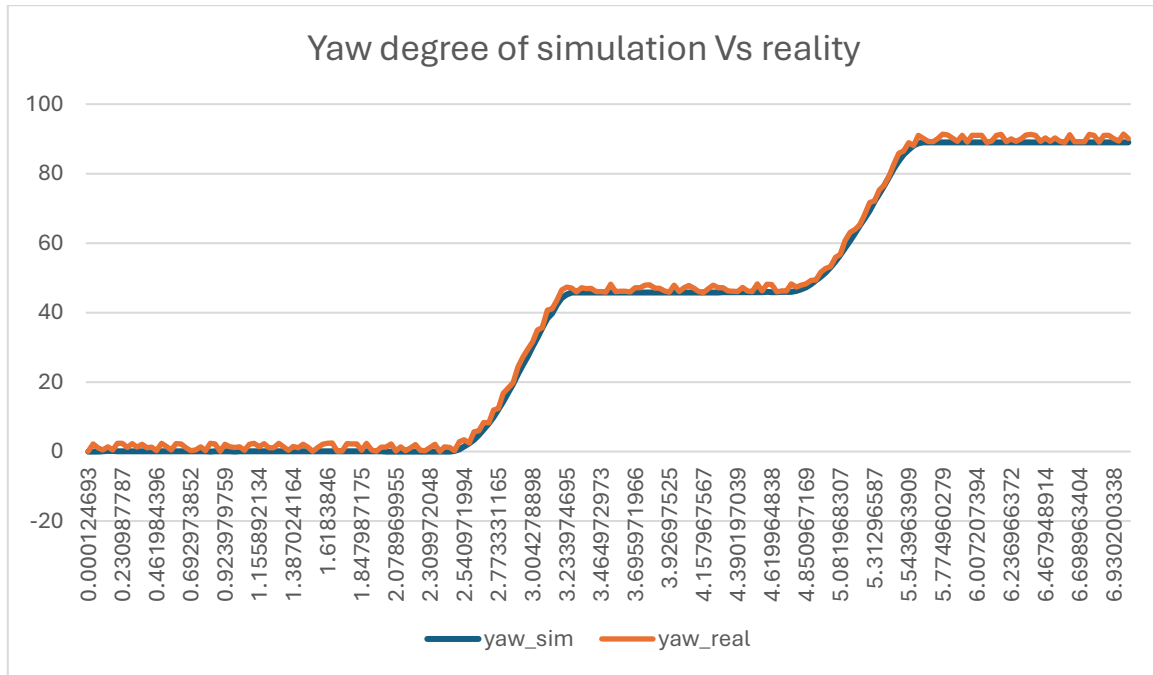


Figure 5, Yaw degree comparison of the simulation Vs physical model

Table 7.1: Reality-Gap Metrics for Yaw ( $\Psi$ )

| Phase              | PCC (Uni.) | PCC (NoU.) | MNCC (Uni.) | MNCC (NoU.) |
|--------------------|------------|------------|-------------|-------------|
| Slot-search        | 0.99       | 0.99       | 0.99        | 1.00        |
| Approach-alignment | 0.99       | 0.99       | 0.99        | 1.00        |
| Reverse-parking    | 0.99       | 0.99       | 0.99        | 1.00        |

The yaw channel shows near-perfect agreement between simulation and reality across all three phases, with consistently high correlation under both sampling strategies. The improvement observed under non-uniform sampling ( where evaluation points are concentrated during transient phases ) indicates that the digital twin accurately captures not only the shape of the yaw dynamics but also their magnitude and timing. This consistency, including during reverse parking where yaw dominates vehicle behaviour, provides strong validation of the digital twin's steering fidelity.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 7.3.2 Discussion of Yaw Consistency

The high yaw similarity suggests that:

- steering calibration,
- wheelbase scaling,
- and turning-radius adaptation

were successfully reproduced inside the digital twin.

Minor deviations appear primarily during:

- rapid steering reversals,
- and the final alignment stage.

These discrepancies are attributed mainly to:

- servo nonlinearities,
- wheel-ground friction variation,
- and actuator latency.

Nevertheless, the results demonstrate that the digital twin accurately captures the dominant steering behaviour required for parking-policy transfer.

### 7.4 Lateral Acceleration Validation

#### 7.4.1 Lateral Acceleration Analysis

Table 7.2 (Lateral Acceleration)

| Phase              | PCC (Uni.) | PCC (NoU.) | MNCC (Uni.) | MNCC (NoU.) |
|--------------------|------------|------------|-------------|-------------|
| Slot-search        | 0.93       | 0.92       | 0.93        | 0.93        |
| Approach-alignment | 0.94       | 0.95       | 0.94        | 0.95        |
| Reverse-parking    | 0.83       | 0.85       | 0.85        | 0.86        |

Lateral acceleration is the most sensitive channel to driving conditions. While agreement remains strong during slot-search and approach-alignment, performance degrades during reverse-parking, suggesting a shape mismatch rather than a systematic bias, likely caused by complex lateral coupling under steering reversal, which is more difficult for the digital twin's tire and suspension model to reproduce exactly than the lower-dynamics phases.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle

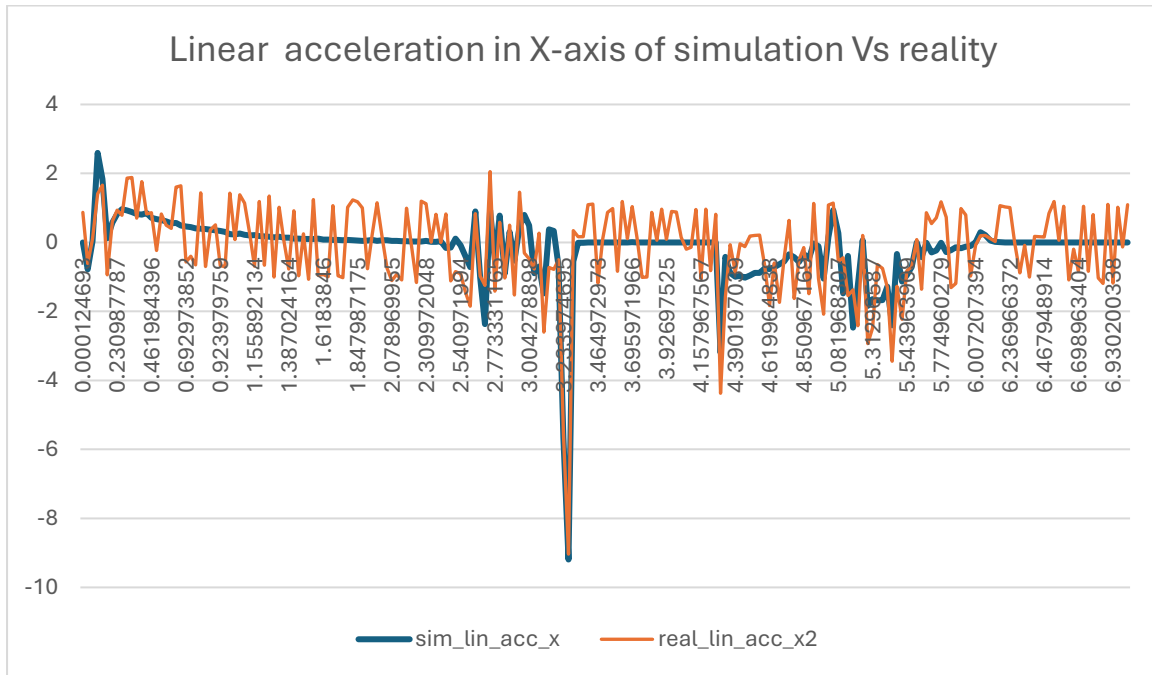


Figure 6, Linear acceleration in X-axis of the simulation Vs physical model

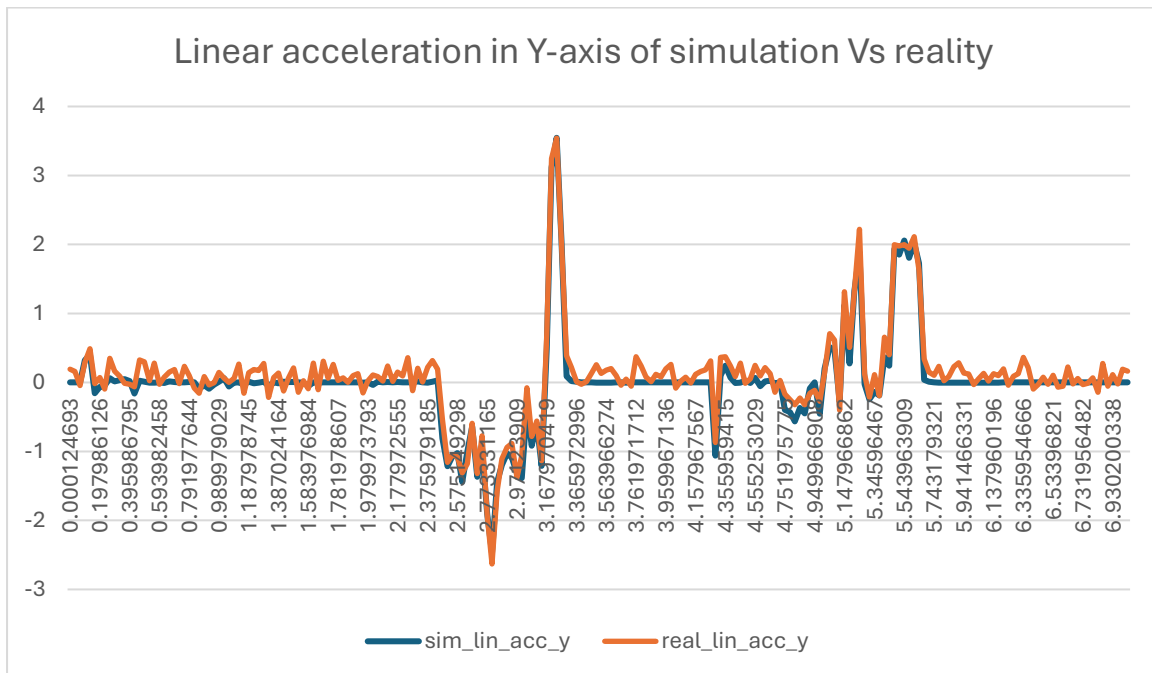


Figure 7, Linear acceleration in Y-axis of the simulation Vs physical model

### 7.4.2 Sources of Dynamic Discrepancy

The remaining mismatch in lateral acceleration originates from several physical phenomena that are difficult to model precisely in CARLA, including:

- micro-sliding effects,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- surface irregularities,
- tire deformation,
- and chassis vibration.

Additionally, the physical IMU measurements contain sensor noise absent from the idealized simulation environment.

Despite these limitations, the simulated lateral dynamics remain sufficiently representative for reinforcement-learning training.

### 7.5 Lateral Offset Validation

#### 7.5.1 Lateral Offset Analysis

Table 7.3 (Lateral Offset)

| Phase              | PCC (Uni.) | PCC (NoU.) | MNCC (Uni.) | MNCC (NoU.) |
|--------------------|------------|------------|-------------|-------------|
| Slot-search        | 0.98       | 0.98       | 0.98        | 0.98        |
| Approach-alignment | 0.98       | 0.98       | 0.90        | 0.90        |
| Reverse-parking    | 0.87       | 0.79       | 0.91        | 0.92        |

Lateral offset maintains strong shape consistency across all phases, with minor discrepancies attributable to reference-frame offsets or localized misalignments between the two platforms' coordinate origins. Combined with the yaw and lateral acceleration results above, the joint PCC/MNCC analysis confirms high-fidelity reproduction of key parking dynamics, while identifying reverse-steer lateral acceleration as the primary remaining source of sim-to-real discrepancy.

#### 7.5.2 Geometric Interpretation

The strong lateral-offset agreement demonstrates that:

- steering constraints,
- vehicle geometry,
- and low-speed dynamics

were successfully reproduced within the digital twin.

This result is particularly important because parking policies depend strongly on geometric trajectory consistency.

### 7.6 Uniform vs Non-Uniform Sampling Analysis

#### 7.6.1 Uniform Sampling Results

Under uniform sampling, the validation metrics emphasize global trajectory consistency across the entire parking manoeuvre.

The results demonstrate:

- high global similarity,
- smooth temporal alignment,



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- and strong phase consistency.

Uniform sampling provides stable overall performance estimates.

### 7.6.2 Non-Uniform Sampling Results

Non-uniform sampling concentrates evaluation points near transient steering phases.

This configuration produces slightly lower similarity metrics because:

- steering reversals,
- and transient acceleration peaks

become more heavily weighted.

The results reveal that the largest discrepancies occur primarily during:

- rapid steering transitions,
- and reverse-direction corrections.

### 7.6.3 Interpretation

The comparison between sampling strategies demonstrates that:

- the digital twin reproduces steady-state behaviour extremely well,
- while transient dynamics remain the primary source of residual mismatch.

This observation is consistent with the physical limitations of simplified tire and actuator modelling .

## 7.7 PPO Training Results

### 7.7.1 Learning Curve Evolution

The PPO agent was trained entirely within the CARLA digital twin.

The reward curve demonstrates progressive policy improvement with increasing training episodes.

Initially, the agent exhibits:

- random exploration,
- unstable steering,
- and frequent collisions.

As training progresses, the policy gradually learns:

- reverse manoeuvre coordination,
- steering correction timing,
- and slot alignment behaviour.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle

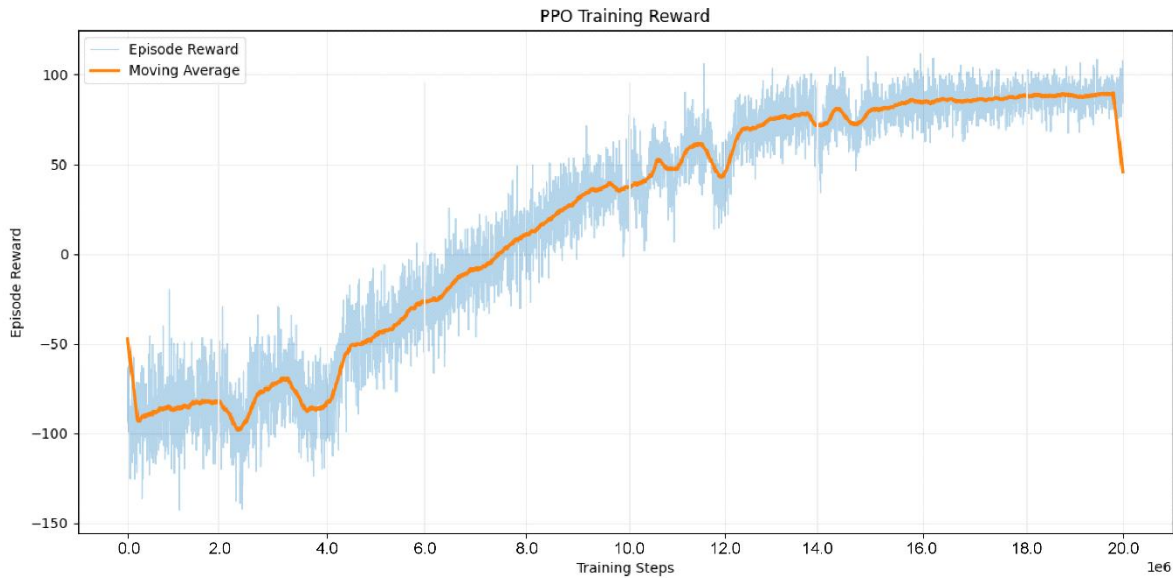


Figure 8, PPO training reward curve over 2,000,000 environment steps.

### 7.7.2 Convergence Analysis

The training reward curve (Figure 8) shows the pattern of reinforcement learning convergence:

an initial period of unstable, low-reward exploration, followed by steady improvement, and a later plateau in which reward gains become marginal. A formal convergence threshold was not quantified in this work; qualitatively, the policy used for deployment (7.8) was taken from the final training checkpoint at 2,000,000 environment steps, by which point the reward curve had visibly stabilized.

### 7.7.3 Emergent Parking Behaviour

One notable observation is that the PPO agent autonomously discovers several human-like parking behaviours, including:

- pre-alignment before reverse entry,
- steering reversal near midpoint insertion,
- and gradual heading correction during final alignment.

These behaviours emerge naturally from reward optimization rather than being explicitly programmed.

## 7.8 Zero-Shot Sim-to-Real Deployment

### 7.8.1 Deployment Procedure

After training completion, the PPO policy was deployed directly onto the physical vehicle without:

- retraining,
- parameter adaptation,
- or fine-tuning.

The same neural-network weights trained in CARLA were executed onboard the NVIDIA Jetson Orin Nano.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 7.8.2 Real-World Parking Performance (PPO deployment)

The trained PPO policy was deployed on the physical vehicle across 20 real-world trials, with no fine-tuning or additional adaptation. For comparison, the policy was evaluated across 20 trials in the CARLA digital twin under matched initial conditions. A trial was counted as successful when the final position error  $\|\Delta p\| \leq 0.05$  m and heading error  $|\Delta \psi| \leq 10^\circ$  at the target slot anchor, consistent with the success bonus defined in the reward function (6.5.4).

Table 7.4: Zero-Shot Sim-to-Real Deployment Performance( PPO Policy )

| Configuration                          | Simulation | Physical Platform |
|----------------------------------------|------------|-------------------|
| <b>Vanilla CARLA (no digital twin)</b> | 61%        | 7%                |
| <b>Validated digital twin</b>          | 60%        | 55%               |

With an unmodified ("vanilla") CARLA vehicle (no parameter matching to the physical Traxxas platform and no digital-twin validation ) a policy achieving 61% success in simulation dropped to just 7% on the real vehicle, a 54-percentage-point collapse that illustrates the severity of the reality gap when simulation fidelity is not addressed. Using the validated digital twin as the training environment instead preserved simulation-level performance (60%) and raised real-world success to 55%, reducing the sim-to-real gap to only 5 percentage points. This demonstrates that the signal-level fidelity established in 7.3–7.5 translates directly into preserved closed-loop policy behaviour after deployment.

The remaining 5-point gap is not attributable to vehicle dynamics mismatch, which 7.3–7.5 show to be well matched. Instead, it stems primarily from the limitations of short-range ultrasonic perception (including noise, limited field of view, and interference ) that are not fully captured in simulation. Failures occurred predominantly during the final approach phase, where perception uncertainty caused loss of alignment rather than control inaccuracy.

### 7.8.3 Performance Degradation Analysis

Although transfer was successful overall, some performance degradation relative to simulation was observed, consistent with the success-rate gap reported in Table 7.4. The primary differences include slightly larger steering oscillations, occasional overcorrection, and reduced final-position precision. These discrepancies originate primarily from ultrasonic sensor noise, actuator latency, and wheel-ground interaction variability. A detailed breakdown of failure modes (e.g., collision vs. non-collision failures) was not recorded during the deployment trials and is noted as a limitation in 8.4.

## 7.9 Failure Case Analysis

### 7.9.1 Perception-Related Failures

The most common failure cases were associated with inaccurate ultrasonic measurements.

Ultrasonic sensors occasionally produced:

- unstable readings near corners,
- multipath reflections,
- and short-term distance spikes.

These errors sometimes caused incorrect steering decisions during close-range alignment.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 7.9.2 Dynamic Mismatch Failures

A smaller subset of failures originated from dynamic discrepancies between simulation and reality.

These cases typically occurred during:

- rapid steering reversals,
- high-curvature reverse trajectories,
- or low-friction surface conditions.

The results suggest that transient dynamics remain the most difficult component of the reality gap to eliminate completely.

### 7.10 Comparison with Existing Approaches

Compared to many existing sim-to-real autonomous-parking studies, the proposed framework offers several advantages:

- low hardware cost,
- reproducible architecture,
- realistic onboard-equivalent sensing,
- quantitative digital-twin validation,
- and successful zero-shot deployment.

Unlike approaches relying on privileged simulator observations, the proposed method maintains perception consistency between:

- simulation,
- and reality.

This significantly improves practical transferability.

### 7.11 Discussion

The experimental results demonstrate that:

- high-fidelity sim-to-real transfer is achievable using low-cost hardware,
- quantitative digital-twin validation improves deployment reliability,
- and CARLA can effectively support scaled autonomous-vehicle research.

The experiments also reveal that:

- steering and trajectory reproduction are relatively straightforward to model,
- while transient dynamics and noisy perception remain major contributors to the residual reality gap.

Overall, the results validate the central hypothesis of this thesis:

- that a carefully calibrated and validated digital twin can enable direct reinforcement-learning policy transfer from simulation to reality.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 7.12 Summary

This chapter presented the experimental evaluation of the proposed sim-to-real autonomous parking framework.

The digital twin demonstrated strong agreement with the physical platform in terms of:

- yaw evolution,
- lateral acceleration,
- and geometric trajectory consistency.

Additionally, the PPO-based parking controller successfully transferred from simulation to reality without fine-tuning, demonstrating the effectiveness of the validated digital twin.

The final chapter concludes the thesis by summarizing the contributions, limitations, and future research directions.

## Chapter 8 Conclusion and Future Work

### 8.1 Introduction

This thesis investigated the development and validation of a low-cost CARLA-based digital twin for sim-to-real autonomous parking using a 1:10-scale vehicle platform and reinforcement learning.

The work addressed one of the central challenges in autonomous-driving research:

- reducing the reality gap between simulation and physical deployment.

To achieve this objective, the thesis proposed:

- a complete sim-to-real framework,
- a validated digital twin,
- and a PPO-based autonomous parking controller capable of zero-shot deployment.

This chapter summarizes the major contributions and findings of the thesis while discussing limitations and future research directions.

### 8.2 Summary of Contributions

The first major contribution of this thesis is the development of a fully reproducible low-cost autonomous vehicle platform based on a commercial Traxxas Slash RC vehicle.

The platform integrates:

- onboard embedded computation,
- ultrasonic sensing,
- continuous control interfaces,
- and real-time autonomous operation.

Unlike many research platforms requiring expensive infrastructure, the proposed system remains accessible and reproducible using commercially available hardware.



## **Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle**



The second contribution is the development of a CARLA-based digital twin specifically calibrated to reproduce:

- vehicle geometry,
- steering behaviour,
- longitudinal dynamics,
- sensor placement,
- and timing characteristics

of the physical platform.

The digital twin was designed to achieve behavioural fidelity. The third contribution is the introduction of a deterministic signal-level validation methodology for evaluating digital-twin fidelity.

Instead of relying solely on task-level controller performance, the proposed validation framework compares:

- yaw angle,
- lateral acceleration,
- and lateral offset

between simulation and reality using:

- PCC,
- and MNCC statistical metrics.

This methodology enables quantitative analysis of the remaining reality gap.

The fourth contribution is the implementation of a PPO-based autonomous parking controller trained entirely in simulation using realistic onboard-equivalent observations.

The controller operates over a shared continuous action space compatible with both simulation and reality.

The final major contribution is the successful zero-shot deployment of the learned reinforcement-learning policy onto the physical platform without:

- retraining,
- fine-tuning,
- or domain adaptation.

The successful transfer demonstrates that the validated digital twin reproduced the physical system with sufficient fidelity for reinforcement-learning deployment.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 8.3 Answers to the Research Questions

#### **RQ1: How accurately can a CARLA-based digital twin reproduce the dynamic behaviour of a physical scaled autonomous vehicle during autonomous parking manoeuvres?**

The validation experiments demonstrated strong agreement between simulation and reality across all three evaluated channels: yaw showed near-perfect correlation ( $PCC = 0.99\text{--}1.00$  across all phases), lateral offset remained strong ( $PCC = 0.79\text{--}0.98$ ), and lateral acceleration (the most sensitive channel) ranged from  $PCC = 0.94$  during approach-alignment down to  $PCC = 0.83$  during reverse-parking. This indicates that the digital twin successfully reproduced the dominant low-speed vehicle dynamics relevant to autonomous parking, with transient reverse-steer dynamics remaining the primary source of residual discrepancy.

#### **RQ2: Which vehicle-state characteristics provide meaningful indicators for evaluating digital twin fidelity between simulation and reality?**

Yaw, lateral acceleration, and lateral offset each proved informative, but at different sensitivities: yaw was consistently the most robust indicator, remaining near-saturated ( $PCC \geq 0.99$ ) across every phase, while lateral acceleration was the most diagnostic signal for identifying where fidelity was weakest, revealing degraded agreement specifically during the reverse-parking phase.

#### **RQ3: Can a reinforcement learning controller trained exclusively within the digital twin environment successfully perform autonomous parking tasks on the physical vehicle without additional adaptation?**

The PPO policy trained entirely in the validated digital twin was deployed directly onto the physical vehicle and achieved a 55% real-world success rate across 20 trials (7.8.2), compared to 60% in simulation, a gap of only 5 percentage points. This confirms that zero-shot transfer is achievable with a sufficiently validated digital twin. The comparison against a policy trained on an unmodified CARLA vehicle, which collapsed from 61% to 7% upon deployment, further confirms that digital-twin validation ( not the learning algorithm alone ) is what enables reliable transfer.

#### **RQ4: What are the primary contributors to the remaining simulation-to-reality discrepancy after digital twin calibration and validation?**

The results identify short-range ultrasonic perception (noise, limited field of view, and interference during the final approach phase ) as the dominant contributor to the residual 5-point performance gap, rather than vehicle dynamics mismatch, which 7.3–7.5 show to be well matched. At the signal level, reverse-steer lateral acceleration remains the least well-reproduced channel and is a secondary contributor.

#### **RQ5: To what extent does maintaining consistency in sensing, vehicle dynamics, action spaces, and timing improve the transferability of reinforcement learning policies?**

The direct comparison between the validated digital twin (55% real-world success) and the vanilla, unmatched CARLA vehicle (7% real-world success) provides strong quantitative support for this question: maintaining equivalent observation structures, matched vehicle dynamics, identical continuous action spaces, and synchronized timing between simulation and reality improved real-world transferability by 48 percentage points over an otherwise identical policy trained without these measures.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 8.4 Limitations

Despite the successful results, several limitations remain.

First, the ultrasonic sensing system provides only sparse local perception and suffers from:

- limited angular resolution,
- noise,
- and reflection artifacts.

Second, the CARLA tire model cannot perfectly reproduce all physical tire-ground interaction effects present in the real platform.

Third, the experiments focused primarily on:

- low-speed reverse perpendicular parking,
- under controlled environmental conditions.

More complex driving scenarios were not investigated.

Finally, although the PPO policy transferred successfully, some performance degradation relative to simulation remained observable during:

- rapid steering transitions,
- and close-range alignment phases.

### 8.5 Future Work

Several directions may extend the work presented in this thesis.

#### 8.5.1 Improved Perception Systems

Future research may integrate:

- cameras,
- LiDAR,
- depth sensors,
- or sensor-fusion architectures

to improve environmental perception and reduce sensitivity to ultrasonic noise.

#### 8.5.2 Advanced Vehicle Dynamics Modelling

More sophisticated tire and friction models could improve reproduction of:

- transient dynamics,
- steering reversals,
- and low-friction surface behaviour.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### 8.5.3 Broader and Dynamics-Oriented Randomization

The current training procedure already incorporates domain randomization over initial vehicle pose, parking-slot geometry, and minor sensor noise perturbations (6.8.2). Future work could extend this to broader and dynamics-oriented randomization, for example, randomizing tire-ground friction coefficients, actuator response latency, vehicle mass and inertia, and lighting or surface conditions affecting ultrasonic reflection, as well as investigating adversarial domain adaptation and online calibration techniques to further reduce the reality gap.

### 8.5.4 End-to-End Learning

Future systems may replace handcrafted observation engineering with:

- end-to-end visual reinforcement learning,
- transformer-based perception,
- or multimodal sensor fusion.

### 8.5.5 Multi-Scenario Autonomous Driving

The proposed framework may be extended beyond parking to include:

- lane following,
- obstacle avoidance,
- urban navigation,
- and cooperative driving scenarios.

### 8.6 Final Remarks

This thesis demonstrated that validated digital twins can provide an effective bridge between simulation and reality for autonomous-driving research.

By combining:

- low-cost hardware,
- open-source simulation,
- quantitative validation,
- and reinforcement learning,

the proposed framework contributes toward more accessible, scalable, and reproducible sim-to-real autonomous systems research.

The results indicate that even relatively inexpensive scaled platforms can support meaningful autonomous-driving experimentation when coupled with carefully calibrated and behaviourally validated digital twins.

Ultimately, reducing the reality gap remains one of the fundamental challenges in autonomous systems engineering. The methodology presented in this thesis represents a step toward simulation environments that are not only visually realistic, but also dynamically and behaviourally trustworthy for real-world deployment.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



### References

- [1] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “CARLA: An Open Urban Driving Simulator,” Nov. 10, 2017, *arXiv*: arXiv:1711.03938. doi: 10.48550/arXiv.1711.03938.
- [2] I. G. Daza, R. Izquierdo, L. M. Martínez, O. Benderius, and D. F. Llorca, “Sim-to-real transfer and reality gap modelling in model predictive control for autonomous driving,” *Appl Intell*, vol. 53, no. 10, pp. 12719–12735, May 2023, doi: 10.1007/s10489-022-04148-1.
- [3] L. Lazzaroni, A. Pighetti, F. Bellotti, A. Capello, M. Cossu, and R. Berta, “Automated Parking in CARLA: A Deep Reinforcement Learning-Based Approach,” in *Applications in Electronics Pervading Industry, Environment and Society*, vol. 1110, F. Bellotti, M. D. Grammatikakis, A. Mansour, M. Ruó Roch, R. Seepold, A. Solanas, and R. Berta, Eds., in *Lecture Notes in Electrical Engineering*, vol. 1110., Cham: Springer Nature Switzerland, 2024, pp. 352–357. doi: 10.1007/978-3-031-48121-5\_50.
- [4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” Aug. 28, 2017, *arXiv*: arXiv:1707.06347. doi: 10.48550/arXiv.1707.06347.
- [5] A. Al-Mousa, A. Arrabi, and H. Daoud, “A reinforcement learning-based reverse-parking system for autonomous vehicles,” *IET Intelligent Trans Sys*, vol. 19, no. 1, p. e12614, Jan. 2025, doi: 10.1049/itr2.12614.
- [6] B. Thunypoo, C. Ratchadakorntham, P. Siricharoen, and W. Susutti, “Self-Parking Car Simulation using Reinforcement Learning Approach for Moderate Complexity Parking Scenario,” in *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Phuket, Thailand: IEEE, Jun. 2020, pp. 576–579. doi: 10.1109/ECTI-CON49241.2020.9158298.
- [7] J. Shi, K. Li, C. Piao, J. Gao, and L. Chen, “Model-Based Predictive Control and Reinforcement Learning for Planning Vehicle-Parking Trajectories for Vertical Parking Spaces,” *Sensors*, vol. 23, no. 16, p. 7124, Aug. 2023, doi: 10.3390/s23167124.
- [8] R. Berta et al., “Development of Deep-Learning-Based Autonomous Agents for Low-Speed Maneuvering in Unity,” *J. Int. Con. Veh.*, vol. 7, no. 3, pp. 229–244, Sep. 2024, doi: 10.26599/JICV.2023.9210039.
- [9] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec. 2020, pp. 737–744. doi: 10.1109/SSCI47803.2020.9308468.
- [10] J. P. Allamaa, P. Patrinos, H. Van Der Auweraer, and T. D. Son, “Sim2real for Autonomous Vehicle Control using Executable Digital Twin,” *IFAC-PapersOnLine*, vol. 55, no. 24, pp. 385–391, 2022, doi: 10.1016/j.ifacol.2022.10.314.
- [11] T. V. Samak, C. V. Samak, S. Kandhasamy, V. Krovi, and M. Xie, “AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin Ecosystem for Enhancing Autonomous Driving Research and Education,” *Robotics*, vol. 12, no. 3, p. 77, May 2023, doi: 10.3390/robotics12030077.



## Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle



- [12] B. Yu, C. Yuan, Z. Wan, J. Tang, F. Kurdahi, and S. Liu, "ADDT -- A Digital Twin Framework for Proactive Safety Validation in Autonomous Driving Systems," Apr. 13, 2025, *arXiv*: arXiv:2504.09461. doi: 10.48550/arXiv.2504.09461.
- [13] T. V. Samak, C. V. Samak, and V. N. Krovi, "Towards Validation of Autonomous Vehicles Across Scales using an Integrated Digital Twin Framework," in *2024 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Jul. 2024, pp. 1068–1075. doi: 10.1109/AIM55361.2024.10637205.
- [14] B. Lee, Y. Wei, and I. Y. Guo, "AUTOMATIC PARKING OF SELF-DRIVING CAR BASED ON LIDAR," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLII-2/W7, pp. 241–246, Sep. 2017, doi: 10.5194/isprs-archives-XLII-2-W7-241-2017.
- [15] Wan-Joo Park, Byung-Sung Kim, Dong-Eun Seo, Dong-Suk Kim, and Kwae-Hi Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *2008 IEEE Intelligent Vehicles Symposium*, Eindhoven, Netherlands: IEEE, Jun. 2008, pp. 1039–1044. doi: 10.1109/IVS.2008.4621296.
- [16] P. Zhang *et al.*, "Reinforcement Learning-Based End-to-End Parking for Automatic Parking System," *Sensors*, vol. 19, no. 18, p. 3996, Sep. 2019, doi: 10.3390/s19183996.
- [17] SAE International, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," SAE Standard J3016\_202104, Apr. 2021.
- [18] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: 10.1038/nature14236.
- [19] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992, doi:10.1007/BF00992696.
- [20] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30, doi: 10.1109/IROS.2017.8202133.
- [21] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles," in *Field and Service Robotics*, Springer Proceedings in Advanced Robotics, vol. 5, Cham: Springer, 2018, pp. 621–635, doi: 10.1007/978-3-319-67361-5\_40.
- [22] N. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robotsimulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2149–2154, doi: 10.1109/IROS.2004.1389727.
- [23] O. Michel, "Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, Article 5, 2004, doi: 10.5772/5618.
- [24] M. Grieves, "Product lifecycle management: the new paradigm for enterprises," *International Journal of Product Development*, vol. 2, nos. 1/2, pp. 71–84, 2005.



## **Sim-to-Real Validation of a CARLA-Based Digital Twin for a 1:10 Scale Autonomous Vehicle**



[25] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," in *Transdisciplinary Perspectives on Complex Systems*, F.-J. Kahlen, S. Flumerfelt, A. Alves, Eds., Cham: Springer, 2017, pp. 85–113, doi: 10.1007/978-3-319-38756-7\_4.