

Dependable Decision Making for Autonomous Vehicles Considering Perception Uncertainties

by

Ruihe Zhang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2024

© Ruihe Zhang 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

- External Examiner: **Amir Aghdam**
Professor, Dept. of Electrical and Computer Engineering,
Concordia University
- Supervisor(s): **Amir Khajepour**
Professor, Dept. of Mechanical and Mechatronics Engineering,
University of Waterloo
- Krzysztof Czarnecki**
Professor, Dept. of Electrical and Computer Engineering,
University of Waterloo
- Internal Member: **James Tung**
Associate Professor, Dept. of Mechanical and Mechatronics
Engineering, University of Waterloo
- Internal-External Member: **Mark Crowley**
Associate Professor, Dept. of Electrical and Computer
Engineering, University of Waterloo
- Internal-External Member: **Bruce Hellinga**
Professor, Dept. of Civil and Environmental Engineering,
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Ruihe Zhang was the sole author of Chapters 1, 4, 5, and 6 which were written under the supervision of Prof. Amir Khajepour and were not published. This thesis consists of part of one manuscript written for publication.

Research presented in Chapters 3:

Chen Sun*, **Ruihe Zhang***, Ahmad Reza Alghooneh, Minghao Ning, Pouya Panahandeh, Steven Tuer and Amir Khajepour, “Cascaded Safety Analysis and Test Scenario Generation Techniques for Autonomous Driving: A Case Study with WATonoBus,” *Automotive Innovation*, 2024, accepted, to appear.

* indicates equal contribution authors. As lead author of this chapter, I was responsible for contributing to conceptualizing study design, carrying out data collection and analysis, and drafting and submitting manuscripts. My coauthors provided guidance during each step of the research and provided feedback on draft manuscripts.

Abstract

Autonomous driving system (ADS), leveraging recent advancements in various learning algorithms, has demonstrated significant potential to enhance traffic safety. However, in the dynamic service environments, one of the crucial challenges in ADS safety evaluation is managing performance uncertainties inherent in these black-box learning algorithms. Among all ADS functional modules, decision-making module is responsible for interpreting sensory results and determining vehicle maneuvers. Thus, developing an uncertainty-aware decision-making module becomes critical for ensuring ADS driving safety.

Building an uncertainty-aware decision-making module necessitates a comprehensive approach to identify the origins of learning algorithm uncertainties within ADS and understand the potential vehicle-level hazards they may cause. Through the associated risk assessment, these identified uncertainties can then guide ADS safety design priority and pinpoint uncertainty quantification requirements. Eventually, the quantified uncertainties and their propagation effects in ADS need to be integrated into the decision-making module to deliver more dependable decisions. However, existing ADS safety research lacks a procedure to connect qualitative uncertainty understanding to quantitative decision-making support evidence. In this thesis, a systematic approach is presented to first qualitatively identify, then quantify, and finally incorporate uncertainties into ADS decision-making process to enhance driving safety.

This thesis presents three main components for constructing a dependable, uncertainty-aware decision-making module. The first part introduces a sequential ADS safety analysis using a combination of Hazard and Operability Study (HAZOP) and System-Theoretic Process Analysis (STPA) to understand the causal relationships and effects of learning algorithm uncertainties within a complex autonomous vehicle system. This analysis aids in generating combinatorial test cases for simulation verification. A detailed real-world case study is presented to demonstrate the effectiveness of the proposed safety analysis method. The second part formulates an uncertainty quantification problem based on the previous analysis results, utilizing High Definition (HD) maps and Polynomial Chaos Expansion (PCE) for statistical analysis. The focus is on pedestrian position uncertainty from the perception module, with simulation and real-world testing results showing promising accuracy of PCE in dynamic environmental conditions. The third part investigates system propagation effect of quantified uncertainty using a Dynamic Bayesian Network (DBN) and integrates the uncertainties into decision-making process through an Influence Diagram (ID) model. By updating the utility functions in the ID, the proposed DBNID method enhances safety performance when encountering unexpected pedestrian behaviors in simulations and changing weather conditions with real-world testing datasets.

Acknowledgements

To me, PhD research is always hard work demanding, sometimes emotionally challenging, and eventually academically or even lifelong rewarding. After I have been through all the ups and downs in this journey and begin to write this section, I am deeply grateful to all the support, guidance, encouragement, and companionship that I received along this way. First and foremost, I would like to thank my supervisor, Prof. Amir Khajepour. His continuous guidance, patient suggestions, and supportive attitude led me through countless difficulties in my research throughout the last five years. With the depth and breadth of his professional knowledge and his profound understanding of automobile industry tendency, I was also fortunate to participate different GM research projects, lead WATonoBus development, and intern twice at GM Research and Development Center in Warren, MI, USA. In addition, I would like to thank Prof. Krzysztof Czarnecki, my co-supervisor, for his religious research attitude and helpful instructions on my work. His expertise in autonomous vehicle safety better shaped my understanding of current research gap in this field. I am also thankful to all my other doctoral committee members for their insightful feedback and constructive suggestions on refining my research.

I want to thank Bakhtiar Litkouhi, Alireza Kasaizadeh, Alon Capua, and Gary Zhang at GM Warren for their support and patient guidance, making my two summers during the internship truly wonderful. I am also very thankful to all my friends at MVS Lab. Thanks to Neel Bhatt who has been my friend since my undergraduate study and whom I can always learn from. Thanks to Chen Sun who is very knowledgeable and would always like to share his new research ideas or insights to broaden my view. Thanks to Yukun Lu and Shucheng Huang for all the lunches, dinners, and good times we enjoyed together as MASc. students. These cherished memories continue to enrich my life today. I cannot list all my treasurable memories with MVS Lab members, but thank you, Ehsan Mohammadbagher, Amin H Korayem, Assylbek Dakibay, Mobin Khamooshi, Yuechi Li, Xin Xia, Yixiao Liang, Meiyin Zhu, Yanjun Huang, Chao Yu, Yufeng Yang, Jiaming Zhong for sharing this journey with me. Special thanks to Ben MacCallum, Ted Ecclestone, Minghao Ning, Yaodong Cui, Ahmad R Alghooneh, Pouya Panahandeh, Reza Valiollahimehrizi, Mohammad Pirani, Aaron Sherratt, Jeff Graansma, Adrian Neill, and Michael Duthie for the fantastic teamwork on WATonoBus development and testing.

Most importantly, I would like to thank the other half of my heart, Zhining Cao, for her unwavering encouragement, companionship, patience, sacrifices, and understanding throughout my 10-year long academic journey from my undergraduate to PhD. In the last decade, our story has woven through Toronto, Waterloo, London and many other corners of the world. With gratitude and love, I look forward to writing the next chapter with her.

Finally, I would like to express my deepest gratitude to my parents, Jingdong and Weiwei, for their immense hard work and sacrifices to offer me the opportunity to freely select and pursuit my path of life. Their love, support, and encouragement have been the foundation upon which I have built my successes. Thank you for always being my pillars of strength and for cheering me on every step of the way.

Dedication

To my parents and Zhining.

Table of Contents

Examining Committee Membership	ii
Author’s Declaration	iii
Statement of Contributions	iv
Abstract	v
Acknowledgements	vi
Dedication	viii
List of Figures	xiii
List of Tables	xvi
1 Introduction	1
1.1 Motivations	1
1.2 Objectives	3
1.3 Outline	4
2 Literature Review	6
2.1 Qualitative ADS Safety Analysis	6

2.1.1	ADS Autonomy Levels and Safety Standards	7
2.1.2	ADS Safety Analysis and Related Tools	8
2.1.3	Simulation Test Case Generation for ADS Safety Verification	10
2.1.4	Remarks	11
2.2	Learning Algorithm Uncertainties and Quantification Techniques	12
2.2.1	Learning Algorithm Uncertainty Fundamentals	12
2.2.2	Learning Algorithm Uncertainty Quantification Techniques	13
2.2.3	Remarks	15
2.3	Decision-making Module with Uncertainty System Propagation	16
2.3.1	Uncertainty Propagation Studies in ADS	16
2.3.2	Uncertainty-aware Decision-making Techniques in ADS	18
2.3.3	Remarks	22
3	Qualitative ADS Safety Analysis	23
3.1	HAZOP and STPA Fundamentals	24
3.2	Sequential ADS Safety Analysis with Stacked HAZOP and STPA	26
3.3	WATonoBus Vehicle-level Safety Hazard and STPA Analysis	33
3.3.1	STPA-1 WATonoBus System Definition	33
3.3.2	STPA-2 System Control Structure of WATonoBus	38
3.3.3	HAZOP Analysis of WATonoBus	45
3.3.4	STPA-3 Identification of UCAs of WATonoBus	48
3.3.5	STPA-4 Identification of Loss Scenarios	52
3.4	WATonoBus Risk Assessment and Vehicle-level Safety Goals	54
3.4.1	WATonoBus Operation Conditions	54
3.4.2	ASIL Assessment for WATonoBus Primary Hazards	54
3.4.3	WATonoBus Vehicle-level Safety Goals	56
3.5	WATonoBus Edge Case Generation	57

4	Learning Algorithm Uncertainty Quantification	61
4.1	Polynomial Chaos Expansion (PCE) Fundamentals	62
4.1.1	PCE Definition and Polynomial Basis	62
4.1.2	PCE Polynomial Coefficient Calculation Methods	64
4.1.3	Bootstrap PCE (bPCE) and Polynomial Chaos Kriging (PCK) . . .	66
4.1.4	Error Estimation and PCE Evaluation	68
4.2	Problem Statement of Quantifying Object Position Uncertainty (OPU) . .	69
4.3	OPU Quantification with PCE	70
4.3.1	Initialization Stage	70
4.3.2	Testing Stage	74
4.4	Simulation and Real-world Experimental Results	80
4.4.1	Simulation Setup	80
4.4.2	Simulation Results	85
4.4.3	Real-world Experiment Setup	91
4.4.4	Real-world Experiment Results	94
4.5	Result Analysis and Discussion	98
5	Uncertainty-aware Decision Making	102
5.1	Bayesian Network and Influence Diagram Fundamentals	103
5.1.1	Bayesian Network (BN) and Dynamic Bayesian Network (DBN) . .	103
5.1.2	Influence Diagram (ID)	108
5.2	Investigate Perception Uncertainty Propagation Effect with DBN	111
5.3	Formulate Uncertainty-aware Decision Making with ID	118
5.4	Simulations and Real-world Experimental Results	124
5.4.1	Simulation Setup	125
5.4.2	Simulation Results	127
5.4.3	Real-world Experimental Setup	137
5.4.4	Real-world Experimental Results	140
5.5	Result Analysis and Discussion	142

6 Conclusion and Future Works	146
6.1 Conclusion	146
6.2 Future Works	148
References	150
APPENDICES	168
A HAZOP Analysis Tables	169
B STPA Analysis Tables	172
C WATonoBus Operation Conditions Generation	175
D ASIL Assessment Table	177
E WATonoBus Primary Hazardous Condition ASIL Assessment	178

List of Figures

1.1	Overview of workflow among three research objectives to achieve ADS uncertainty identification, quantification, and incorporation for safer and more interpretable decision-making performance. One uncertainty and its system propagation trace is highlighted in red in objective 1.	4
3.1	Flow Diagram of HAZOP Process [1]	25
3.2	Flow Diagram of STPA Process [2]	26
3.3	Overall Schematic Diagram of Sequential ADS Safety Analysis and Test Case Generation	28
3.4	WATonoBus Shuttle and Its Operation Path on Ring Road	33
3.5	Overall Schematic Diagram of WATonoBus	36
3.6	Sensor Hardware Monitor and Control Mode Arbiter Workflow of WATonoBus	41
3.7	Sensor Components and Their Checking Criteria (Part 1)	42
3.8	Sensor Components and Their Checking Criteria (Part 2)	43
3.9	Ego Vehicle Status Determination Workflow	44
3.10	Actuator Interface Message Generation Logic of WATonoBus	45
3.11	Causal Factors Considered in STPA [3]	50
3.12	STPA Traceability Diagram Example of PH3	52
3.13	Overall STPA Traceability Diagram of WATonoBus	53
3.14	Example of Determining the Operation Boundary for WATonoBus FSM Decision-Making Module in Atomic Scenario 2	60
4.1	Schematic Diagram of OPU Quantification	71

4.2	Visualization of OPU Quantification with PCE	72
4.3	Statistics of Pedestrian Occurrence from Vehicle Perspective [4]	72
4.4	OPU Distributions in Different Zones from Real-world Data	74
4.5	Visualization of Optimal P^{OD} and P^{ND} Calculation Process	78
4.6	Example of a Simulation Scenario Scene	81
4.7	Detection Zone Representation for PCE Initialization and Testing in Simulation	84
4.8	Visualization of OPU in PCE Simulation	85
4.9	Plots of PCE Validation Results from $M_{z_1}^{PCE}$, $M_{z_2}^{PCE}$ and M_{entire}^{PCE} in Simulation	87
4.10	Comparison between a) Normal OPU and b) Doubled OPU in PCE Simulation	89
4.11	RMSE and APE Results Plots for a) First and b) Second Detection Zones with Optimal P^{OD} and P^{ND}	90
4.12	OPU Quantification Results for Both Detection Zones in Simulation	91
4.13	Three Front Camera Views in a) Initialization Dataset, b) Testing Datasets that Not Requires PCE Model Update, c) Testing Datasets that Requires PCE Model Update	92
4.14	Detection Zone Representation for PCE Initialization and Testing in Real World Experiment	93
4.15	Real World Initialization Dataset Distribution in a) Isometric View and b) Top View	94
4.16	Real World Snowy Testing Dataset a) APE Results of $M_{z_4}^{PCE}$ and b) Data Distribution Visualization	97
4.17	Visualization of Predicted Variance and True Object Intersection	98
4.18	z_4 Data Distribution Comparison for a) Testing Dataset and b) Initialization Dataset with Sunny Weather	100
5.1	BN Example [5]	104
5.2	DBN Example [5]	105
5.3	A Sample Influence Diagram [5]	109
5.4	Highlighted STPA Traceability Diagram for DBN Formulation	111

5.5	Block Diagram of the DBN Model for Assessing Uncertainty Propagation .	112
5.6	Visualization of Three LPC Evaluation Criteria	114
5.7	Visualization of a Sample $g(\cdot)$	117
5.8	Block Diagram of the Proposed ID Model	119
5.9	Utility Function 1 Visualization. (a) U_1 Interpolation Results for $D_1 = Stop$, (b) U_1 Interpolation Results for $D_1 = Go$, (c) U_1 Visualization with Discretized TTC Levels	121
5.10	Utility Function 2 Visualization with $LPC = True$ and Discretized TTC Levels. (Top Row) U_2 Interpolation Results for $D_2 = Autonomous$, (Middle Row) U_2 Interpolation Results for $D_2 = Max Speed Reduction$, (Bottom Row) U_2 Interpolation Results for $D_2 = Max Speed Reduction \& Increase Obstacle Buffer Size$	122
5.11	DBNID Simulation Data Workflow	125
5.12	(Top) Original OPU and (Bottom) Filtered OPU that are Linearly Proportional to the Distance between Ego Vehicle and Object	127
5.13	Unexpected pedestrian road crossing and turning back case. The blue vehicle denotes the DBNID ego vehicle and the purple vehicle denotes the baseline ego vehicle. The true pedestrian crossing trajectory is shown with red segments representing high OPU periods. Sidewalk has been filled with diagonal brick pattern.	132
5.14	Result Comparison between Proposed DBNID Method and Baseline Method	133
5.15	(a) 3D View of Original U_2 (b) 3D View of Updated U_2 (c) Top View of Original U_2 (d) Top View of Updated U_2 , when $LPC = False$	134
5.16	Real-world Dataflow of DBNID	137
5.17	Real-world Dataflow of DBNID	139
C.1	Operation Conditions Summary of WATonoBus	176

List of Tables

3.1	High-level ODD Taxonomy Categories [6]	29
3.2	ASIL Assessment Table [7]	31
3.3	Control Actions of Decision-making Module in WATonoBus	39
3.4	Control Actions of Motion Planning and Vehicle Control Modules in WATonoBus	39
3.5	Control Actions of Safety Monitor Module in WATonoBus	39
3.6	Control Actions of Control Mode Arbiter Module in WATonoBus	40
3.7	ADS Functions of WATonoBus	46
3.8	HAZOP Study Results	47
3.9	Primary Hazards Table	48
3.10	Secondary Hazards Table	49
3.11	Vehicle Control Functional Modules' UCAs	50
3.12	Causal Factor for Implemented Sub-controllers	51
3.13	Variables Covered in WATonoBus Operation Conditions	55
3.14	WATonoBus Primary Hazards ASIL Assessment	56
3.15	WATonoBus Vehicle-level Safety Goals	56
3.16	Generated Test Cases for WATonoBus Operation Conditions 94 and 95	58
4.1	Common Families of Univariate Orthonormal Polynomials	64
4.2	Simulated Initialization Scenarios for PCE	81
4.3	Simulated Testing Scenarios for PCE	82
4.4	Validation Results of PCE, bPCE, PCK in Simulation	86

4.5	Validation Results of PCE with and without Detection Zone Divisions . . .	87
4.6	Testing Results of PCE without Online Model Update	88
4.7	Optimal P^{OD} and P^{ND} for Two Detection Zones	89
4.8	MAE and RMSE of PCE Real-world Validation and Testing Results with Sunny Weather	95
4.9	RMSE of PCE Real-world Testing Results with Different Model Update Cases and Snowy Weather	96
4.10	Comparison Results between PCE and Baseline with Snowy Testing Dataset	97
5.1	Simulated Initialization Scenarios for DBN	126
5.2	N_{STV} Comparison between DBNID and Baseline Methods	128
5.3	TTC_{min} Comparison between DBNID and Baseline Methods	129
5.4	N_{STV} and TTC_{min} Comparison Before and After DBNID update	135
5.5	COTC and TMC Estimation Accuracy at Initialization and Testing Stages	137
5.6	Simulated Initialization Scenarios for DBN	138
5.7	N_{STV} and TTC_{min} Testing Results for DBNID Real-world Experiments . .	140
5.8	Real-world Experiment COTC and TMC Estimation Accuracy	141
A.1	Causal Factors for the Implemented Sub-controllers	169
B.1	Decision-making Module Control Actions	172
B.2	Motion Planning and Vehicle Control Modules Control Actions	173
B.3	Safety Strategy Selection Module Control Actions	174
B.4	Control Mode Arbiter Module Control Actions	174
D.1	ASIL Assessment Table	177
E.1	ASIL Assessment for PH1	178
E.2	ASIL Assessment for PH2	180
E.3	ASIL Assessment for PH3	182
E.4	ASIL Assessment for PH4	183
E.5	ASIL Assessment for PH5	185

Chapter 1

Introduction

Autonomous driving systems (ADS) are at the forefront of technological innovation. They offer numerous benefits like reducing human errors, congestion, emission, and enhancing accessibility. However, ensuring the safe operation of autonomous driving is always the first stride towards fully realizing these benefits. While ADS can navigate complex environments with the help of different learning algorithms, it must contend with various conditions, ranging from fluctuating weather conditions to unpredictable road user behaviours. Failure to adequately address these learning algorithm uncertainties could jeopardize the safety of ADS, undermining public trust and impeding widespread adoption. Among all ADS functional modules, decision making governs ego vehicle maneuvers, interacts with other traffic participants, and is responsible for the driving safety. Thus, uncertainty-aware decision making has gained significant research interest in the autonomous driving field. The objective of this thesis is to present a systematic approach to firstly identify, then quantify, and finally account for learning algorithm uncertainties into ADS decision-making process to enhance driving safety. In this chapter, motivations to develop uncertainty-aware decision-making algorithms are presented in Section 1.1, followed by the objectives of the proposed approach in Section 1.2. Furthermore, Section 1.3 presents the outline of the core chapters in this thesis.

1.1 Motivations

In the last two decades, there has been a growing research interest towards improving the safety performance of ADS technologies to facilitate their widespread commercialization.

Through collaborative efforts involving universities, automobile manufacturers, and governments, several stand-alone advanced driver assistance systems (ADAS) functionalities [8] have been released to the market [9]. Moreover, safety standards like ISO26262 and ISO21448 have served as references for the design and evaluation of ADS safety measures [7, 10]. Nevertheless, a systematic procedure for identifying vehicle-level hazards arising from both noisy observations (i.e. aleatoric uncertainties) and model deficiencies (i.e. epistemic uncertainties) [11] across different functional module levels remains lacking. This challenge stems from the black-box nature of the learning algorithms and the intricate function interactions within ADS [12]. Addressing this issue is crucial as it not only limits a comprehensive understanding of ADS safety performance, sometimes misguiding iterative safety enhancements, but also undermines customer confidence and acceptance of ADS technologies [13].

After understanding the qualitative hazard-uncertainty relationships in ADS, it becomes imperative to quantify the identified uncertainties originating from the ADS learning algorithms, employing appropriate criticality measures. Although learning algorithms demonstrate proficiency in processing vast amounts of data and learning complex patterns, they often lack transparency in their result generation processes [14]. This opacity poses challenges in tracing how uncertainties propagate through the system and assessing their impact on safety-critical decisions, hindering efforts to mitigate potential risks effectively [15]. Moreover, ADS operates within highly dynamic environments where factors such as weather conditions and traffic participants' behaviors continuously fluctuate. These uncertainties manifest diversely, including sensor noises, environmental variability, and temporal shifts in training datasets, presenting substantial hurdles to precise risk assessment and decision making [11, 16]. Furthermore, ADS must make split-second decisions in dynamic environments and this real-time requirement imposes another constraint on many existing uncertainty quantification methods [17, 18]. Hence, developing an online and general uncertainty quantification method capable of addressing various forms of identified uncertainties from ADS learning algorithms and their system propagation impact emerges as a pivotal next stride.

The quantified uncertainties need to be integrated into the decision-making process to enable the generation of uncertainty-aware high-level driving maneuvers for vehicle path planning module and low-level motion control module. Similar to uncertainty quantification, real-time processing is also a valid constraint for designing uncertainty-aware decision-making mechanisms that yield safety-critical responses efficiently. Furthermore, from a safety analysis perspective, ensuring the transparency and interpretability of a decision-making process is essential for instilling public confidence, especially in instances where black-box learning algorithms are prevalent in ADS [19]. Uncertainty interpreta-

tion will be particularly challenging to achieve when some end-to-end or learning-based planning approaches are employed. Moreover, given the diverse forms and dynamic nature of quantified uncertainties, decision-making algorithms should adaptively incorporate them and remain capable of dynamic updates based on evolving conditions to ensure safety across diverse environments. Therefore, formulating an online uncertainty-aware decision-making method that can accommodate varying uncertainties represents another crucial step towards effectively translating the knowledge of identified ADS uncertainties and safety hazards to tangible safety performance enhancements.

1.2 Objectives

This thesis endeavors to enhance the safety and interpretability of ADS by integrating the impact of uncertainties encountered by ADS learning algorithms into decision-making processes, thereby yielding safer and more transparent high-level driving behavior decisions. To realize this goal, the thesis delineates three sequential objectives (see Figure 1.1):

I. Uncertainty Identification and Vehicle-level Hazard Analysis

This objective first necessitates identifying the vehicle-level hazards associated with each functional module in ADS. Subsequently, the traceability between ADS vehicle hazards and learning algorithm uncertainties is analyzed, providing a foundation for later uncertainty quantification work. Furthermore, leveraging the identified hazard-uncertainty traceability, specific edge case test scenarios are generated to facilitate the examination of the proposed uncertainty-aware decision-making algorithm in simulation. Detailed exploration of hazard analysis, identification of ADS hazard-uncertainty relationships, and edge case generation are presented in Chapter 3.

II. Quantification of Identified Uncertainties

This objective entails the development of a comprehensive uncertainty quantification framework capable of addressing various types of identified uncertainties or learning algorithms in real time. Drawing insights from the findings of the first objective, specific uncertainties are quantified in this thesis as illustrative examples. Detailed discussions on formulating a real-time, general uncertainty quantification method to accommodate diverse identified ADS uncertainties are presented in Chapter 4.

III. Uncertainty Incorporation within Decision-making Module

This objective first involves addressing the system propagation impact of the quantified uncertainties and necessitates the proposed decision-making algorithm to be online, adap-

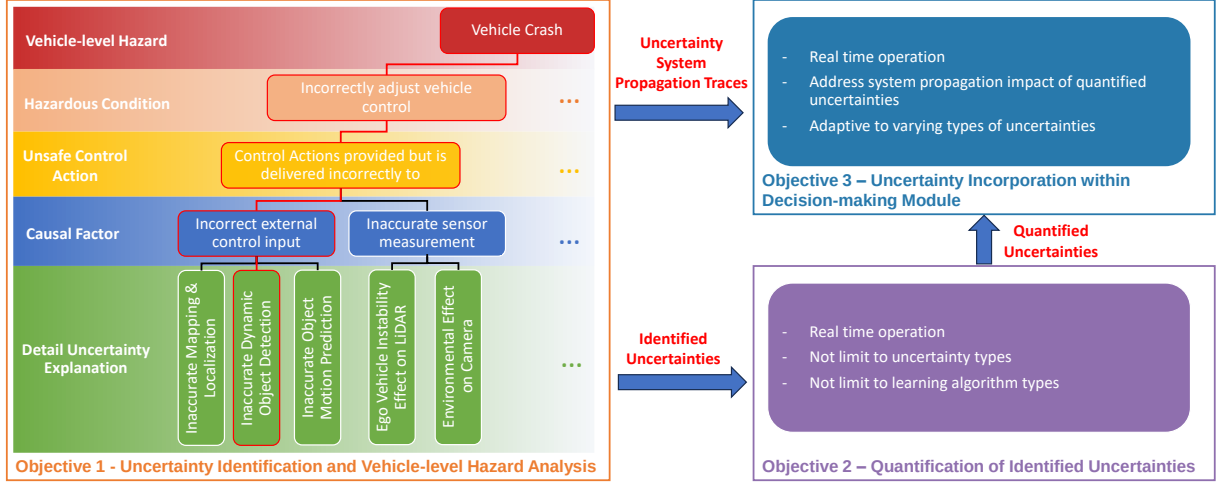


Figure 1.1: Overview of workflow among three research objectives to achieve ADS uncertainty identification, quantification, and incorporation for safer and more interpretable decision-making performance. One uncertainty and its system propagation trace is highlighted in red in objective 1.

tive to varying uncertainties and self-updating. Detailed exploration of this real-time, uncertainty-aware decision-making algorithm is provided in Chapter 5.

1.3 Outline

The remainder of this thesis is organized as below:

In Chapter 2, Section 2.1 firstly reviews existing ADS safety standards, focusing on how these standards identify potential uncertainties and conduct qualitative safety analyses for ADAS functions. The research gap mentioned in the first objective is then addressed. Section 2.2 discusses various uncertainty quantification methods, particularly in ADS perception and prediction, highlighting their limitations and connection to the second research objective. Section 2.3 reviews current uncertainty-aware decision-making techniques, outlining their characteristics and relevance to the third research objective.

Chapter 3 details the process of safety hazard analysis using Hazard and Operability Study (HAZOP) and Systems Theoretic Process Analysis (STPA), along with an edge case generation method. Section 3.1 provides an overview of these safety analysis tools, while Section 3.2 presents the proposed sequential safety analysis method. The chapter

applies these techniques to the WATonoBus project, identifying vehicle-level safety hazards, linking learning algorithm uncertainties to potential malfunctions (Section ??), assessing their risks (Section 3.4), and generating edge cases for testing (Section 3.5).

Chapter 4 focuses on quantifying Object Position Uncertainty (OPU) within the perception module using Polynomial Chaos Expansion (PCE). Section 4.1 introduces the principles of PCE, while Section 4.2 formulates the OPU quantification problem. Section 4.3 details the methodology for quantifying OPU, and Section 4.4 presents simulation and experimental results, comparing the PCE-based approach with a baseline model. Section 4.5 discusses the effectiveness and reliability of the proposed method.

Chapter 5 explores the system propagation effects of quantified uncertainty and implements an uncertainty-aware decision-making method using Dynamic Bayesian Network and Influence Diagram (DBNID). Section 5.1 provides an overview of Bayesian networks and influence diagrams. Sections 5.2 and 5.3 detail the DBNID methodology, focusing on how perception uncertainty affects pedestrian trajectory prediction and threat assessment. Section 5.4 presents comparative results with a baseline model, followed by a discussion in Section 5.5.

Finally, in Chapter 6, the important findings and related future research directions are summarized.

Chapter 2

Literature Review

In this chapter, a comprehensive review of existing safety analyses conducted on ADS and ADAS functions is undertaken, along with their employed Safety Of The Intended Functionality (SOTIF) techniques in Section 2.1. This review aims to elucidate the current efforts invested by both academia and industry in enhancing ADS safety. Additionally, the assumptions and limitations inherent in these analyses are scrutinized, particularly underscoring the need for safety enhancement by leveraging insights into uncertainties present in ADS learning algorithms and their impacts on the decision-making process. Section 2.2 then focuses on current uncertainty quantification methods, especially those tailored for data-driven algorithms utilized in ADS applications. Section 2.3 reviews the mainstream decision-making techniques and how uncertainties have been incorporated in the decision-making module. By reviewing existing practices, this section seeks to identify prevailing approaches and highlight areas for improvement in effectively incorporating uncertainties into decision-making processes. Finally, a synthesis of the literature review findings is presented, outlining how existing studies identify, quantify, and incorporate uncertainties in ADS safety designs. Through this summary, this chapter concludes by pinpointing the research gap that this thesis endeavors to address and solve.

2.1 Qualitative ADS Safety Analysis

Ensuring the safe operation of vehicles is a top priority in the automotive industry, particularly when commercial vehicles increasingly integrate ADAS and autonomous driving functionalities. As these technologies become more prevalent, maintaining safety standards is essential to prevent accidents and uphold public trust in these systems. Established

frameworks like ISO 26262 [7] offer guidelines for developing functional safety measures within automotive systems, encompassing both ADS and ADAS functions. Additionally, qualitative safety analyses related to SOTIF [10] focus on the learning algorithms of ADS to enhance safe decision-making processes. Once the uncertainties are identified, corresponding simulation test cases are usually generated in the first place to examine the ADS designs against the evaluation criteria. The following subsections will delve into the existing safety standards, review the qualitative ADS safety analyses, highlight the advantages and limitations of these analyses, and introduce the commonly used simulation test case generating mechanism in ADS safety verification field.

2.1.1 ADS Autonomy Levels and Safety Standards

The Society of Automotive Engineers (SAE) has defined six levels of vehicle autonomy, known as the SAE autonomy levels, which categorize the degree of automation in vehicles. These levels range from Level 0, where the human driver controls all aspects of driving, to Level 5, where the vehicle is fully autonomous and capable of operating without any human intervention [8]. At Level 1, the vehicle features driver assistance systems that can control either steering or acceleration/deceleration, but not both simultaneously. Level 2 introduces partial automation, where the vehicle can control both steering and acceleration/deceleration simultaneously under certain conditions, but the human driver must remain engaged and always monitor the traffic. At Level 3, conditional automation is achieved, allowing the vehicle to perform all driving tasks under specific conditions or environments, but the human driver must be ready to intervene when prompted by the system. Level 4 represents high automation, where the vehicle can operate autonomously in predefined environments or scenarios without human intervention, although human takeover may still be required in certain situations. Finally, Level 5 signifies full automation, where the vehicle is capable of autonomous operation in all driving conditions without any human input or supervision. As vehicles transition from lower to higher SAE autonomy levels, the importance of handling uncertainties in learning algorithms and making safe decisions becomes increasingly critical [20]. At higher autonomy levels, where vehicles operate with greater automation and reduced human intervention, learning algorithms play a significant role in decision-making processes. However, these algorithms introduce uncertainties due to factors such as data variability, model complexity, and dynamic operating environments. It is essential to effectively manage and mitigate these uncertainties to ensure the safe operation of ADS. Therefore, the existing vehicle safety standards for ADS and ADAS functions are reviewed.

To ensure robust safety performance, the vehicle’s electrical system must meet the

functional safety standards outlined in ISO 26262 and mitigate hardware risks [7]. ISO 26262 utilizes Automotive Safety Integrity Level (ASIL) classifications to quantify the level of risk reduction necessary to mitigate specific hazards, ranging from the lowest ASIL-A to the highest ASIL-D. However, traditional safety standards may not fully address the unique challenges presented by machine learning algorithms within ADS systems [21]. In response, the concept of SOTIF has emerged as a complementary approach to the traditional functional safety standards. SOTIF focuses on ensuring the safety of autonomous systems under all foreseeable conditions, including scenarios where the system operates as intended but generates unexpected or unsafe outcomes. Consequently, ISO 21448 was developed to mitigate risks stemming from performance limitations or insufficient situational awareness of learning-based algorithms [10, 22, 23]. Subsequently, a complementary standard, UL 4600, was drafted to address the safety of autonomous vehicles on public roads, with a focus on both urban and highway use cases [24].

Besides the safety standards governing ADS, there are specific standards and testing procedures tailored for various ADAS functions. For instance, ISO standards such as 17361, 15622, and 20035 are formulated for Lane Departure Warning (LDW), Adaptive Cruising Control (ACC), and Cooperative Adaptive Cruising Control (CACC) for Level 1 or Level 2 functions [25, 26, 27]. Moreover, for more advanced Level 4 autonomy functions like Low-speed Automated Driving (LSAD) and Automated Valet Parking (AVP), corresponding safety standards such as ISO 22737 and 23374 have been developed [28, 29]. Additionally, recently released standards like ISO 19237 and 22078 cater to object sensing and detection functions, such as Pedestrian/Bicyclist Detection and Collision Mitigation (PDCM/BDCM) [30, 31]. These ADAS safety standards delineate that functions with lower autonomy levels typically operate within a broader service environment and impose lesser demands on learning algorithms. In contrast, higher-level autonomy systems require the ability to address more intricate perception tasks, thus currently constrained within a narrower operational domain.

2.1.2 ADS Safety Analysis and Related Tools

Given the recognized challenges in evaluating the safety of autonomous driving, including intricate requirements, non-deterministic algorithms, and fail-operational systems [32], coupled with the release of ISO 21448, numerous studies have been conducted to investigate the SOTIF issues in autonomous vehicles. Tchoya et al. [33] delved into the primary uncertainties encountered in the development process of ADAS functions. They identified five key categories of uncertainty, including the inherent uncertainty of trained models, sensor measurements affected by noise in the surrounding environment, uncertainties in

sensor fusion interpretation, decision-making challenges in conflicting scenarios, and uncertainties arising from unexpected traffic actors’ intentions. Additionally, studies outlined in [34] expanded on these findings by incorporating additional sources of uncertainty, such as temporal changes in the static traffic environment and uncertainties associated with low-level vehicle components.

In addition to primary uncertainties, a significant contributor to the inadequacy of SOTIF in ADAS is the provision of inaccurate inputs by learning-based algorithms for decision-making. These algorithms are predominantly utilized in the perception module for tasks such as object detection, tracking, and semantic segmentation [35]. Consequently, many SOTIF-focused investigations aim to qualitatively identify the potential causes behind the uncertain performance of the perception module. M. Kläs and L. Sembach provided an overview in [36, 37], elucidating how uncertainties affect the outcomes of perception algorithms. These uncertainties are broadly categorized into three classes: model fit, data quality, and scope compliance. Model fit pertains to the inherent statistical uncertainties within trained machine learning models, while data quality refers to the noise present in training and testing data due to suboptimal data collection conditions. Scope compliance denotes issues arising when trained models are applied outside the Operational Design Domain (ODD). Additionally, their studies involved the development of a conceptual probabilistic model to approximate uncertainties related to situation awareness, stemming from data quality and scope compliance concerns.

In [38], Oliver Willers et al. delved deeper into the safety implications of deep learning models on a technical level, building upon the previously identified concerns. They highlighted uncertainties associated with deep learning models in safe-critical scenarios such as shifting data distributions in real-world traffic environments and the absence of threat assessment mechanisms to evaluate the results of deep learning algorithms. Drawing from ISO 21448 guidelines, qualitative mitigation methods were then outlined, including the justification of data acquisition strategies, the provision of reliable confidence information, and the utilization of gray-box algorithms. However, these mitigation strategies were presented as future research directions in the study, lacking quantitative explanations.

Alex and Yarin [11] delved deeper into the uncertainties inherent in vision tasks for autonomous vehicles. They delineated two primary types of uncertainties: aleatoric and epistemic uncertainties. Aleatoric uncertainty pertains to environmental observation noises, while epistemic uncertainty involves uncertainties within the Neural Network (NN) model, which can potentially be mitigated with sufficient data coverage. The study also introduced a Bayesian-based deep learning model designed to address uncertainties in vision tasks, particularly focusing on epistemic uncertainties. However, the investigation did not extend to aleatoric uncertainties stemming from dataset noises in the model or the

dynamics of ODD.

Apart from the existing safety analyses on identifying the uncertainties behind ADS learning algorithms, there are also various well-established hazard identification techniques, which can aid the qualitative safety analysis along with the guidelines of SOTIF. These techniques include Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Hazard and Operability Study (HAZOP), and System-Theoretic Process Analysis (STPA) [39, 40, 41]. FTA and ETA primarily concentrate on analyzing system failures and their consequences, providing valuable insights into potential failure modes and their probabilities. In contrast, HAZOP delves into process deviations and associated risks, meticulously examining the variations from intended operations and their potential hazards. STPA takes a holistic approach by considering system interactions and control failures, aiming to uncover underlying vulnerabilities and systemic safety implications. Despite the extensive analyses provided by these methodologies, ongoing research still endeavors to bridge the identified hazards to the ODD of ADS and formulating effective edge case scenarios, aiming to enhance safety assurance in autonomous driving.

2.1.3 Simulation Test Case Generation for ADS Safety Verification

There are various methods available for assessing the SOTIF capabilities of ADS, namely scenario-based approach, formal verification, function-based approach, real-world testing, shadow mode testing, and traffic-simulation-based approach. Among all these methods, the scenario-based approach is very promising in the current state of art [42] and has been employed in large-scale research projects [43, 44]. Therefore, how to generate test cases effectively and comprehensively in simulation obtains heavy research interests.

Overall, simulation test cases can be generated through either a knowledge-based approach or data-driven approach. Specifically, knowledge-based approach uses abstract information or expert knowledge to directly create functional, logical, and concrete scenarios for a test case database. Based on the knowledge of ADS function scope and service environment, [45] firstly proposed a fundamental ontology for ADS test case construction, creating a concrete basis for other later studies. Following that, more ADS-related ontologies are proposed for highway test case generation [46, 47] and other complex scenarios [48]. On the other hand, a data-driven scenario generation approach can directly determine concrete test scenarios without the help from any predefined logical scenarios or expert knowledge heuristics. Given the strength of processing high-dimensional databases from machine learning algorithms, the data-driven approach can sometimes detect novel-

ties in existing data and sometime even generate new concrete test scenarios [42]. One data-driven scenario generation approach is to evaluate the uniqueness among scenarios through compressing the reproducing the time signals using an auto-encoder to identify the concrete ones [49]. In [50], generative neural networks are employed to model maneuver trajectories from the historical data and generate new trajectories in the test cases. Similarly, recurrent neural networks and Bayesian networks are used to model vehicle driving data and to generate new test scenarios in [51, 52].

The previously introduced two test case generation approaches feature on the effectiveness of new, concrete scenario generation, but lack of the focus on low-cost or automated test case scenario generation, which is a practical requirement from industry perspective. [53] highlights the endeavor to reduce the cost of scenario collection by using microscopic traffic simulation. By leveraging public data and simulation techniques, many urban traffic scenarios are generated, covering various driving situations including collision accidents. Meanwhile, [54] uses combinatorial interaction testing combined with trajectory planning to automatically generate test scenarios for ADS. This approach focuses on generating efficient test sets with variable coverage in a virtual simulation environment. [55] presents a method for generating test scenarios that demonstrate critical transitions in autonomous system performance modes. By running the autonomous system through full missions in a simulated environment and measuring performance based on high-level mission criteria, this approach can automatically search for challenging scenarios. Lastly, [56] proposes GeoScenario as a Domain-Specific Language (DSL) for scenario representation, aiming to substantiate test cases in simulation for ADS evaluation.

2.1.4 Remarks

The complexity of learning algorithms used in ADS poses challenges in accessing its safety performance, especially in dynamic service environments. To assert the safety and robustness of ADS, researchers and engineers need to have a comprehensive understanding about what are the possible learning algorithm uncertainties and where they stem from. Hence, it is essential to systematically identify the potential ADS hazards and link them with the causal learning algorithm uncertainties and then effectively generate test cases to cover these identified hazards. However, challenges still exist in systematically identifying the causal relationships between learning algorithm uncertainties and ADS hazards, and use that as a reference to generate corresponding testing scenarios, which is our first research objective: systematically identifying learning algorithm uncertainties and effectively generate edge cases based on a comprehensive vehicle-level hazard analysis.

2.2 Learning Algorithm Uncertainties and Quantification Techniques

Learning algorithms are undeniably strong performers in the realm of data science, especially in the applications demanding precise prediction of complex, non-linear, patterns across high-dimensional datasets. Nevertheless, the efficacy of deep learning algorithms to reason with their output uncertainties or to address detection failures remains questionable. The necessity to characterize uncertainty in learning algorithms arises from the frequent discrepancies between the distributions of training and testing data. As the service environment changes, new test data may deviate from the training data distribution, leading to potentially out-of-distribution scenarios that compromise the reliability of learning algorithm predictions [57, 58, 59]. Moreover, this mismatch can exacerbate the discrepancy between actual predictive performance and measured prediction confidence, resulting in a silent failure. This bottleneck impedes the widespread adoption of safety-critical application L4 autonomous driving systems who heavily depend on different learning algorithms [8, 60]. Consequently, the following sub-sections will explain fundamental concepts of uncertainties encountered in learning algorithms and review contemporary state-of-art techniques for quantifying uncertainty, highlighting their benefits and limitations.

2.2.1 Learning Algorithm Uncertainty Fundamentals

Uncertainty in learning algorithms reflects the confidence level of their inference results. One widely used measure of uncertainty is softmax probability, which stems from the softmax function applied to the output of a neural network’s final layer [61]. In classification tasks, the softmax function converts the raw output scores into probabilities that sum up to 1. These probabilities represent the model’s confidence or certainty in assigning each input to one of the possible classes. A higher softmax confidence indicates greater certainty in the prediction made by the model, while lower confidence values suggest uncertainty or ambiguity. However, using softmax probability as confidence scores can lead to deterministic overconfidence issues, particularly in out-of-domain contexts. In specific, [62] empirically showed that applying softmax activation to a single point estimate can yield inflated and unrealistic probabilities, resulting in overconfidence. Conversely, passing a distribution of estimates through the softmax function produces more reasonable, lower confidence scores. This realization has motivated the research into representing learning algorithm uncertainty based on a predictive distribution rather than a single prediction. Consequently, various uncertainty measurement approaches such as temperature scaling,

Monte Carlo (MC) dropout, and deep ensembles [62, 63, 64] have emerged, aiming to address these issues. These methodologies will be thoroughly examined in the upcoming sub-section.

In addition to softmax probability, quantifiable uncertainty can be decomposed into two constituent components: epistemic (model) uncertainty and aleatoric (data) uncertainty [11, 64]. Specifically, through learning algorithms like Bayesian Neural Networks (BNN), a predictive distribution provides a set of different predictions from which total predictive uncertainty can be estimated. The total predictive uncertainty can then be split into epistemic and aleatoric uncertainties. Epistemic uncertainty, inherent to the model, arises from limitations in training data [65]. For example, underrepresented minority groups in a facial recognition dataset can contribute to epistemic uncertainty. This uncertainty diminishes with an increase in the volume of training data, indicating the model’s growing familiarity with diverse data. Aleatoric uncertainty, on the other hand, stems from the inherent randomness in the data and cannot be explained away [65]. For instance, aleatoric uncertainty includes noisy telemetry data or low-resolution images. Furthermore, aleatoric uncertainty can be categorized as either homoscedastic or heteroscedastic, based on either the data inherent randomness is treated as constant or a function of input explanatory variables [11, 66]. It is usually not sufficient to assume constant (homoscedastic) aleatoric risk since this assumption rarely reflects the reality. It may overlook valuable information about the relationships between targets and noisy inputs. In contrast, variable (heteroscedastic) aleatoric risk can be calculated through averaging a sample set of predictive variances and can be more informative in model parameter learning. Information theory like Shannon entropy is also used to aid describing total uncertainty. Mutual information then enables the breakdown of the total uncertainty into its epistemic and aleatoric components. By incorporating information-theoretic aleatoric uncertainty, the robustness of the overall uncertainty estimation can be enhanced [67, 68, 69]. Furthermore, this predictive entropy is available to both Bayesian and non-Bayesian neural networks.

2.2.2 Learning Algorithm Uncertainty Quantification Techniques

For a long time, prediction probabilities from softmax functions were presumed to reflect the confidence of model predictions. However, this assumption no longer holds true for today’s neural network models and learning algorithms because they often have large architectures with millions of learning parameters [63]. To address this, various calibration methods have been proposed as post-processing steps to adjust the output prediction probabilities of the model. Among them, temperature scaling is a simple yet effective technique [58, 63]. Essentially, temperature scaling uses a scalar parameter, T , to rescale

logit scores before applying softmax function. By adjusting T , the recalibrated probabilities are distributed more evenly, rather than being heavily skewed towards either 1 or 0. Furthermore, an optimal T can be determined through minimizing the negative log likelihood for a held-out validation dataset. However, despite its effectiveness in quantifying total predictive uncertainty, temperature scaling lacks robustness in capturing epistemic uncertainty, particularly when facing data drift scenarios [63].

Another widely utilized method for assessing uncertainty involves deriving predictive distributions through BNN or Bayesian techniques. Monte Carlo (MC) dropout offers a scalable way to model predictive distributions by intermittently deactivating neurons within a neural network, thereby regularizing its performance [62]. Each dropout configuration generates a distinct sample from the parametric predictive distribution, and aggregating multiple forward passes with varied dropout configurations produces an overarching predictive distribution. This technique garnered substantial attention upon its introduction. In comparison to conventional non-maximum suppression methods, MC dropout is adopted in [17] to enhance the reliability of uncertainty quantification in object detection algorithms. In addition, MC dropout has been implemented to quantify both epistemic and aleatoric uncertainty in semantic segmentation tasks across diverse domains such as autonomous driving [70, 71, 72, 73], medical image interpretation [74, 75, 76, 77, 78, 79] and flight delay prediction [80]. Since there is no need to change original model training process, MC dropout has been implemented to various learning algorithms in the previously mentioned studies, including Xception [18], DeepLab v3+ [81], SegNet [82], Convolution Neural Network (CNN), Mixture Density Network (MDN) [83], and so on. However, MC dropout exhibits limitations when tested with out-of-domain data and struggles to effectively quantify uncertainty in deep BNNs [84, 85, 86].

Deep ensemble is another powerful method for quantifying uncertainty in learning algorithms. At its core, ensemble methods leverage a collection of models, each with its own strengths, to collectively compensate for individual weaknesses. The process typically involves two key steps: training diverse models and integrating their predictions. The two basic ensembling techniques are boosting [87] and random forest [88]. Boosting involves iteratively training models with subpar performance, enabling subsequent weak learners to learn from the errors of previous iterations, thereby boosting overall performance. On the other hand, random forests entail training multiple decision tree models without pruning, often on bootstrapped data samples or by selecting random subsets of features. In their work [89], Gregory Kahn et al. calculated collision prediction uncertainties by assessing the variances of bootstrapped neural network models. These uncertainties are iteratively aggregated until accurate collision probabilities are achieved, facilitating proficient navigation for robots within their environment. Similarly, in [90], Bjorn Lutjens et al. employed

an ensemble of long short-term memory models trained and tested using MC dropout and bootstrapping to estimate collision probabilities. Beyond employing bootstrap and dropout techniques in neural network models, the mean and variance of prediction results are utilized to gauge uncertainties in collision predictions, enabling cautious behavior in novel scenarios. In another application [91], real-time uncertainty quantification combines LiDAR-based (e.g., PointPillars [92]) and camera-based (e.g., SMOKE [93]) object detection algorithms within a deep ensemble framework. This approach evaluates the spatial uncertainty of detected objects to assess the effectiveness of single-frame perception results. Deep ensembles have demonstrated superior performance compared to MC dropout in uncertainty quantification across numerous classification and regression tasks [94] and out-of-distribution datasets [59]. However, they tend to demand more resources and time and may not perform optimally on smaller problems [60].

In addition to the aforementioned methods, several other approaches for uncertainty quantification have been proposed, each with its unique strengths. Mario Berk et al. in [95] address the reliance on ground truth information by examining sensor perception reliability through exploiting redundancies in sensors. Their study integrates object detection accuracy and object occurrence in the test environment into the likelihood function used in Bayesian inference, demonstrating the potential to address inherent model uncertainties, measurement noise, and changes in ODD. Similarly, Xiupeng Shi et al. [96] utilizes unsupervised machine learning techniques to evaluate the risk of surrounding traffic based on four different sensor configurations. They employ specific feature learning techniques to identify critical parameters from both ego and other vehicles’ motion parameters, and Bayesian optimization is employed to automatically tune hyperparameters in the learning algorithms. To accommodate various perception learning algorithms and adapt to data drifts, [97] proposes a probabilistic approach for evaluating perception systems, considering spatial and temporal aspects of perception uncertainty as well as the operational domain of the perception module. The perception uncertainty model is parameterized from both spatial and temporal perspectives in the offline evaluation process. In pursuit of the improved real-time uncertainty quantification performance, [98] suggests that quantification can be achieved not only through confidence estimation but also through model monitoring based on past experiences of success or failure cases [99, 100], or inconsistencies during inference arising from model performance degradation [101, 102].

2.2.3 Remarks

Understanding and effectively quantifying uncertainty in learning algorithms is paramount for reliable decision-making in various applications. The fundamentals of learning algo-

rithm uncertainty reveal the limitations of traditional approaches like softmax probability and advocate for predictive distribution-based methods to capture both epistemic and aleatoric uncertainties. Techniques such as temperature scaling, Monte Carlo dropout, and deep ensembles offer promising avenues for addressing these challenges. While each method has its strengths and weaknesses, their collective contributions advance the field’s understanding of uncertainty quantification. Additionally, emerging approaches, such as exploiting sensor redundancies, unsupervised machine learning, and probabilistic evaluation frameworks, further enrich the toolbox for robust uncertainty estimation. However, a gap remains in uncertainty quantification methods that meet our second research objective: being real-time operational while also comprehensive enough to address various types of uncertainties arising from different learning algorithms.

2.3 Decision-making Module with Uncertainty System Propagation

The Decision-Making (DM) module is a critical component of ADS, responsible for interpreting sensor data and making real-time decisions to navigate through surrounding traffic safely and efficiently. The performance and reliability of this module are paramount, directly impacting the vehicle’s ability to handle complex driving scenarios and ensure passenger safety. However, as highlighted in Section 2.2, it is challenging to quantify uncertainties inherent in learning algorithms. These uncertainties can propagate through the system and influence the effectiveness of the DM module [103]. They may lead to incorrect decisions, potentially compromising ADS safety and operational performance. Understanding and mitigating the effects of such uncertainties is essential for enhancing the robustness and reliability of ADS decision-making processes. In the following sub-sections, we will review studies on uncertainty system propagation in ADS, focusing primarily on uncertainty propagation from perception and prediction modules, as they are upstream functional modules with the most substantial impact on downstream modules. We will then present commonly employed ADS decision-making methods, followed by recent advancements in uncertainty-aware decision-making works.

2.3.1 Uncertainty Propagation Studies in ADS

Perception module uncertainties can originate from both the external environment and the internal system. Specifically, environmental factors such as sensor resolution variation,

internal sensor noise, and measurement fluctuations can significantly impact the perception module [104]. Additionally, perception learning algorithms are susceptible to both aleatoric and epistemic uncertainties [11]. Even worse, these uncertainties can propagate through the system, affecting the performance of other ADS modules. To address perception uncertainty propagation challenges, deep learning techniques are commonly employed. [105] incorporates state uncertainty from the perception module into trajectory predictions for ADS using a Conditional Variational Autoencoder (CVAE) framework. This approach explicitly models the multimodality of actors’ future behaviors and employs a statistical distance-based loss function to better align predicted uncertainties with real-world data. Evaluation results demonstrate that this method effectively propagates perceptual uncertainties, resulting in more accurate and reliable trajectory forecasts. In another study, [62] suggests leveraging Bayesian Deep Learning (BDL) within ADS functional modules, particularly in the perception, prediction, and decision-making layers, to quantify and propagate uncertainties throughout the system. Although BDL shows promise in enhancing safety by addressing observation and model uncertainties, the study notes several limitations. These include a lack of formal explanation and quantitative evaluation metrics for the proposed BDL design, as well as the absence of experimental validation using simulations or real-world data. Other than deep learning techniques, statistical techniques are also investigated in this field. [106] proposes a framework for evaluating uncertainty propagation in perception architectures by modeling the ADS environment and its interactions as random variables. This method uses Probability Density Functions (PDFs) derived from empirical data to simulate variations in the state of the surrounding environment. By identifying inputs that generate high uncertainties, the approach highlights system limitations. However, challenges include ensuring comprehensive parameter scopes, defining connections and weights between components, calculating sensor data uncertainty, and managing the extensive computational requirements of testing all input permutations. Further contributions come from [107] and [108], which utilize a Bayesian network approach to monitor, quantify, and propagate perception uncertainty for rear-end collision avoidance and system diagnosis purposes. These studies underscore the potential of Bayesian models in enhancing ADS safety through collision risk assessment and fault management, yet also indicate the need for further empirical validation and refinement in real-world applications.

Similar to the perception module, the prediction module uncertainty mainly includes aleatoric and epistemic uncertainties [103]. While learning-based prediction algorithms are powerful in handling high-dimensional data, they can perform poorly when the scene differs significantly from the training datasets. Therefore, understanding how prediction uncertainty propagates is crucial for monitoring potential performance degradation and ensuring safer decision-making. Specifically, this challenge is addressed by various proba-

bilistic modeling techniques. [109] introduces a Bayesian framework for maneuver-based trajectory prediction and criticality assessment in autonomous driving. This approach uses Bayesian inference to determine the probability of various driving maneuvers, including a unique class for irrational behaviors. By accounting for the uncertainties of all vehicles' current and future maneuvers, this probabilistic trajectory prediction model enhances the information passed to the decision-making module, significantly improving ADS safety. Further, [15] uses non-linear motion models to derive closed-form expressions for the probability distributions of both state estimation and prediction uncertainties. The derived uncertainties and how their propagation effect will then be used to assess threat levels in ADS. However, it only considers the ego vehicle's data and assumes no interaction among traffic participants. In addition, closed-form expressions are used in [110] to address uncertainty propagation in nonlinear stochastic dynamical systems. This method focuses on the moments of probability distributions over a planning horizon, contributing to better estimation, planning, control, and safety analysis in stochastic systems.

2.3.2 Uncertainty-aware Decision-making Techniques in ADS

In this sub-section, we first review the existing mainstream decision-making techniques. Subsequently, to incorporate the quantified uncertainties from other functional modules, this section then presents the current uncertainty propagation handling techniques used in decision-making module.

The decision-making module of an autonomous vehicle generally comprises three sub-modules: route planning, behavioral decision-making, and motion planning [111]. At the top level, the route planning module develops an overall route based on global map information. The behavioral decision-making module then continuously generates appropriate driving maneuvers by interpreting the perception results of the traffic environment. These maneuvers ensure that the ego vehicle complies with local traffic rules and reaches its destination. Finally, the motion planning module constructs a local path to navigate through the traffic environment according to the high-level driving maneuvers.

The existing decision-making approaches can be divided into two categories, namely rule-based and learning-based methods. The first decision-making approach is to model the ego vehicle maneuvers as the states in a finite state machine (FSM) [110, 112, 113, 114]. The state transitions are governed by various logic conditions according to both perception module inputs and local traffic rules [115]. These transitions are predefined based on expert knowledge and have strong interpretability and transparency in ADS maneuver reasoning. Build upon on rule-based decision-making methods, more advanced probabilistic planning

like Markov Decision Process (MDP) and Partially Observable Markov Decision Process (POMDP) are employed to approximate the behaviors of other traffic participants [116] or unobserved traffic participants' intents [117]. Nevertheless, these predefined transitions need to be well-designed to accommodate the dynamic external environment and traffic conditions, which is usually infeasible and becomes a fundamental limit for enhancing the performance of rule-based decision-making methods. In contrast, the learning-based methods are data-driven and rely on interactions with the environment. Recently, studies have been done using both Reinforcement Learning (RL) [118, 119] and deep learning [120, 121, 122]. In addition, other learning-based techniques like Gaussian Process Regression (GPR) are utilized to consider other traffic participants' intentions [123]. However, due to the lack of transparency in these learning-based decision-making methods, the safety of ADS may be compromised when ADS encounters out-of-distribution traffic patterns or unseen traffic scenarios.

The decision-making module typically interprets the outputs of both perception and prediction modules to generate driving maneuver commands. As a result, it is affected by the uncertainties propagated from these modules. The remaining section reviews various techniques used in the decision-making module to handle such uncertainties. Deep learning techniques are commonly employed to address this issue. For instance, one study proposes a perception uncertainty adaptive planner using RL with a distributed value function, demonstrating that the planner can maintain adequate safety performance even with poor perception quality [124]. Additionally, a deep ensemble technique has been used to estimate the results of the motion prediction model and integrate them into an uncertainty-aware potential field for safer decision making [125]. Similar research has been conducted to capture epistemic uncertainty for another prediction model in [126]. Furthermore, a Bayesian deep learning method has been proposed to evaluate the influence of perception module uncertainties on other functional modules in ADS [127]. This method considers perception, prediction, and decision-making modules, estimating the reliability of each module as a Gaussian probabilistic distribution through MC dropout. Consequently, perception module uncertainties can be statistically propagated through each modeled functional module. Evaluation metrics are designed to assess the accuracy of the propagated uncertainties. However, these metrics are only presented qualitatively, without results from related simulations or real-world experiments.

In addition to deep learning techniques, Bayesian inference methods have been employed to investigate the influence of uncertain driving maneuver predictions on the criticality assessment module [109]. Bayesian inference is used to estimate the probabilities of various driving maneuvers of other vehicles, resulting in different vehicle trajectory predictions. Eight driving maneuvers are mathematically modeled for other vehicles to

improve trajectory predictions. Both sensor measurement noises and the uncertainties of the Bayesian inference algorithm are modeled using Gaussian distributions. The joint probability distributions of each vehicle’s future trajectory are then examined to assess the criticality of potential vehicle collisions. Similarly, another study uses Bayesian inference to analyze the impact of uncertainty propagation on the ego vehicle’s collision avoidance function [15]. This function comprises four components: vehicle motion analysis, global-local frame transformation, collision probability calculation, and criticality measurement implementation. Mathematical models are formulated for various vehicle motions, such as constant velocity, constant turn rate, and acceleration scenarios, to analyze other vehicles’ motions within the prediction horizon. Due to discrepancies between actual and modeled vehicle motions, uncertainty propagation can occur during the transitions between vehicle motion analysis and global-local frame transformation stages. Based on Bayesian inference, the uncertainties in the vehicle motion model are expressed as variances in the probability distributions. The effect of uncertainty propagation is then approximated through the transformation matrices between the vehicle frame and the global frame.

In one study [128], POMDP and deep RL are integrated with Monte Carlo Tree Search (MCTS) to determine the ego vehicle’s lane change maneuvers. The generated lane change maneuvers provide feedback to improve the accuracy of the proposed deep POMDP algorithm. Vehicle dynamics measurement errors, modeled with Gaussian distributions, represent the uncertainties in the decision-making module. Compared to two baseline methods—MCTS-only planning and the Intelligent Driver Model (IDM) presented in another study [129]—the proposed MCTS + deep POMDP method performs better in highway driving and highway exit scenarios. In addition to highway scenarios, POMDP is applied to intersection decision-making to handle uncertainties arising from perception sensor occlusions. Another study [130] proposes a POMDP algorithm to decide the optimal longitudinal acceleration policy at intersections, considering uncertainties such as the unknown intentions of other drivers, their future longitudinal positions, their probabilistic interactions with the ego vehicle, and sensor noises. Various intersection geometries are tested in a simulated environment. The Adaptive Belief Tree (ABT) algorithm [131] is used as the POMDP solver to achieve real-time operation. Similarly, another approach integrates POMDP and utility fusion to handle unobservable objects at pedestrian crosswalks and intersections [132]. In this case, the uncertainties are due to sensor measurement errors of other traffic actors’ positions and velocities. Online POMDP solver is used to perform the computation, achieving real-time performance at a millisecond scale. However, the testing scenarios need to be significantly simplified to ensure the real-time performance of these POMDP algorithms.

Network reliability is used in [133] to develop a Reliability Graph with General Gates

(RGGG) for mathematically modelling the system reliability. Specifically, this paper demonstrates two RGGG examples, namely acyclic structure and bridge structure, and their strengths on evaluating system reliability for decision making. Similarly, influence diagram is also studied to assess decision making due to the simple interpretability. The structure of a decision-making influence diagram contains several chance nodes, decision nodes and a utility node. [134] proposes an approach using influence diagrams to handle the uncertainties of object detection in lane change decision-making process. Specifically, the unscented transform of object detection results is used to approximate the Gaussian distributions of situation parameters like Normalized Lateral Position (NLP) and Deceleration to Safety Time (DST). These situation parameters are then used in the proposed adaptive likelihood functions in the influence diagrams to evaluate each lane status and provide safe lane changing decisions. [135] extends the influence diagram design from [134] to consider the uncertainties from object occlusions in perception module. The influence diagram is constructed for decision making using maximum expected likelihood method. Moreover, probability distributions are assigned for both vehicle detections and lane detections. These probabilities form the chance nodes to represent lane occupancy probabilities and will be considered in the expected utility calculation. In addition, this paper introduces entropy representation to quantify the uncertainties of the lane changing decisions. Simulation and field test results are presented in this work. [110] implements influence diagrams to perform a Bayesian probabilistic threat assessment on the traffic conditions between current lane and adjacent lanes. The threat assessment on different lanes is obtained by using threat measures like Time to Collision (TTC), Minimal Safety Margin (MSM), and Time to Brake (TTB). According to the threat assessment results, an FSM is designed to make high-level ego vehicle maneuver decisions. The uncertainties considered in the paper are the probabilities of missing vehicle detections and false positive detections in perception module. To resolve these uncertainties, a time window is added to filter out the detection noises. Tests have been performed in both simulation and real-world environments and the focused traffic scenario is highway normal driving. Apart from single level influence diagram methodologies, Dimia Iberraken et al in [136] presented a Two-Sequential Level Decision Network (TSLDN) to handle uncertainty system propagation. In this paper, TSLDN employs two lane changing safety criteria, namely Extended Time-to-Collision (ETTC) and Predicted Inter-Distance Profile (PIDP). The objective of developing TSLDN is to perform both driving situation assessment and decision-making during highway lane changing at the same time. The proposed influence diagram contains a two-level network: Maneuver Decision Level (MDL) and Safety Verification Decision Level (SVDL). MDL is responsible for selecting the most appropriate lane change maneuver, considering the current situation assessment using ETTC and sensor measurement uncertainties. Based on PIDP, SVDL will then perform a safety checking on the driving maneuvers generated from MDL. As

the chosen maneuvers are instantaneously taken, if a false alarm is triggered by the wrong information from the perception module, the MDL will immediately abort the previous decision and compute another one. Furthermore, SVDL verifies the coherence of driving maneuvers and minimizes the risks of abrupt maneuver changes.

2.3.3 Remarks

The decision-making module of ADS plays a pivotal role in interpreting sensor data and making real-time decisions to ensure safe navigation through traffic. However, uncertainties inherent in learning algorithms pose significant challenges to the reliability and robustness of the DM. These uncertainties, stemming from perception and prediction modules, can propagate throughout the system, potentially leading to incorrect decisions and compromising ADS safety. To address these challenges, various techniques have been explored. Deep learning methods offer promising avenues for modeling and propagating uncertainties. However, these approaches often lack formal explanations and require rigorous validation to ensure their effectiveness in real-world scenarios. Additionally, Bayesian inference methods and statistical techniques have been employed to handle uncertainties in both perception and prediction modules. These methods aim to enhance trajectory predictions and assess threat levels, contributing to safer decision-making processes in ADS.

In terms of decision-making approaches, a dichotomy exists between rule-based methods and learning-based methods. Rule-based methods provide interpretability but may struggle to adapt to dynamic environments. On the other hand, learning-based methods offer data-driven approaches but may lack transparency and robustness in unforeseen scenarios. While these methods offer valuable insights, challenges remain in achieving real-time performance and empirical validation. Furthermore, network reliability and influence diagrams have been utilized to model system reliability and handle uncertainties in decision-making processes. As an extension to Bayesian network, influence network can both model the uncertainties transitions in a system and provide a clear graphical interpretation. However, the proposed influence networks in the existing studies all perform decision making on low-level driving maneuvers like lane changing instead of the overall vehicle driving mode or the driving authority selection. Hence, influence diagram will be employed in this work to analyze the uncertainty propagation in different functional modules so that the proper vehicle driving mode can be decided for the autonomous vehicles to enhance their safety performance.

Chapter 3

Qualitative ADS Safety Analysis

Safety assurance stands as a paramount concern in the deployment of high-level autonomy ADS [13] due to its heavy reliance on complex learning algorithms for decision-making processes. While considerable efforts have been dedicated to drafting associated safety standards and conducting related safety analyses, addressing the uncertainties inherent in these intricate learning algorithms from a vehicle system perspective remains a formidable task. Bridging the gap between these uncertainties’ root causes and vehicle-level hazards, and effectively devising test scenarios to encompass these identified uncertainties, continues to pose challenges and has garnered significant interest in ADS safety research. In this work, we propose a sequential ADS safety analysis with stacked HAZOP and STPA to understand the causal relationships and effects of learning algorithms’ uncertainties on vehicle-level hazards. Additionally, we introduce a combinatorial test case generation method tailored to explore the identified uncertainties across a heterogeneous parameter search space. Section 3.1 provides an overview of HAZOP and STPA fundamentals, laying the groundwork for subsequent sections. We then present the stacked HAZOP and STPA framework for ADS safety analysis in Section 3.2, followed by an explanation on the combinatorial test case generation method. Section ?? presents a case study illustrating the sequential safety analysis on a real-world ADS platform, WATonoBus, following by the risk assessment and vehicle-level safety goal definitions in Section 3.4. Section 3.5 delves into the WATonoBus test case generation process.

3.1 HAZOP and STPA Fundamentals

The Hazard and Operability Study (HAZOP) is a systematic method widely employed across industries to identify hazards in complex systems and mitigate risks effectively. Figure 3.1 illustrates the typical procedure for conducting a HAZOP [1]. Initially, a multidisciplinary team is assembled, and the scope and objectives of the study are outlined. Nodes, which represent specific components within the system under analysis such as equipment, control systems, or procedural steps, are then identified for examination during the analysis. The team utilizes HAZOP guidewords, including "No or not," "Part of," "More than intended," "Less than intended," "Late function performance," and "Incorrect performance," to systematically stimulate brainstorming and explore potential deviations from ideal operating conditions.

During HAZOP sessions, these guidewords are applied to each selected node to uncover possible functional deviations. If a deviation is identified, its impact on system safety performance or other relevant aspects is evaluated, along with an assessment of its likelihood and severity. The findings, including detailed descriptions of deviations, assessed consequences, and recommended mitigation measures, are then documented. Based on the severity and likelihood of deviations, a risk assessment is conducted to prioritize follow-up actions and allocate resources efficiently.

The unique advantage of HAZOP lies in its thoroughness on breaking down a system into its constituent parts and examining each systematically. By considering all possible deviations from normal operating condition with the aid from HAZOP guidewords, it can uncover both common hazards and rare or unforeseen hazardous scenarios. This advantage enables the design team to develop corresponding safety protocols. Furthermore, it can guide the design modifications aimed at reducing the likelihood of future accidents. Additionally, the collaborative nature of HAZOP promotes effective risk assessment and knowledge sharing among experts from diverse backgrounds.

System-Theoretic Process Analysis (STPA) is a model-based safety analysis technique tailored for complex systems. Figure 3.2 illustrates the steps for performing a STPA [2]. The initial step in STPA involves delineating the system boundaries, encompassing the identification of system components, functional module interfaces, system functionalities, and environmental interactions. Subsequently, a meticulous analysis is conducted to unearth potential hazards associated with the system, culminating in the identification of hazardous scenarios. STPA then scrutinizes the specific control actions within the system, termed Unsafe Control Actions (UCAs), that could cause the aforementioned hazardous scenarios.

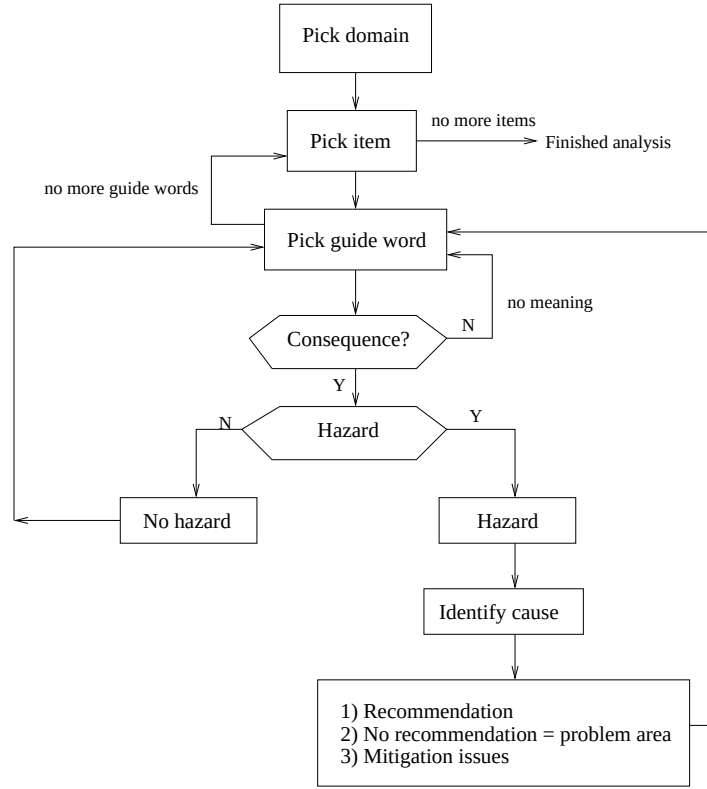


Figure 3.1: Flow Diagram of HAZOP Process [1]

Any UCAs deemed insufficient to forestall hazardous scenarios are flagged, and pertinent safety constraints are documented for iterative design enhancements. Each scenario undergoes assessment for potential harm, leading to the identification of loss scenarios if the assessed harm surpasses acceptable thresholds. Leveraging the identified hazards, UCAs, and loss scenarios, STPA facilitates the generation of safety requirements aimed at mitigating or preventing hazards via design modifications or the incorporation of risk control measures. Ultimately, STPA concludes with the verification and validation of the efficacy of implemented risk control actions or design modifications.

Unlike HAZOP, STPA not only scrutinizes component failures but also examines the interactions or dependencies among different system elements. Specifically, it focuses on the underlying control structures and feedback loops within the interested system. Thus, STPA is better at uncovering systemic vulnerabilities that might not be apparent through conventional analysis techniques. Furthermore, STPA offers a clear traceability path from

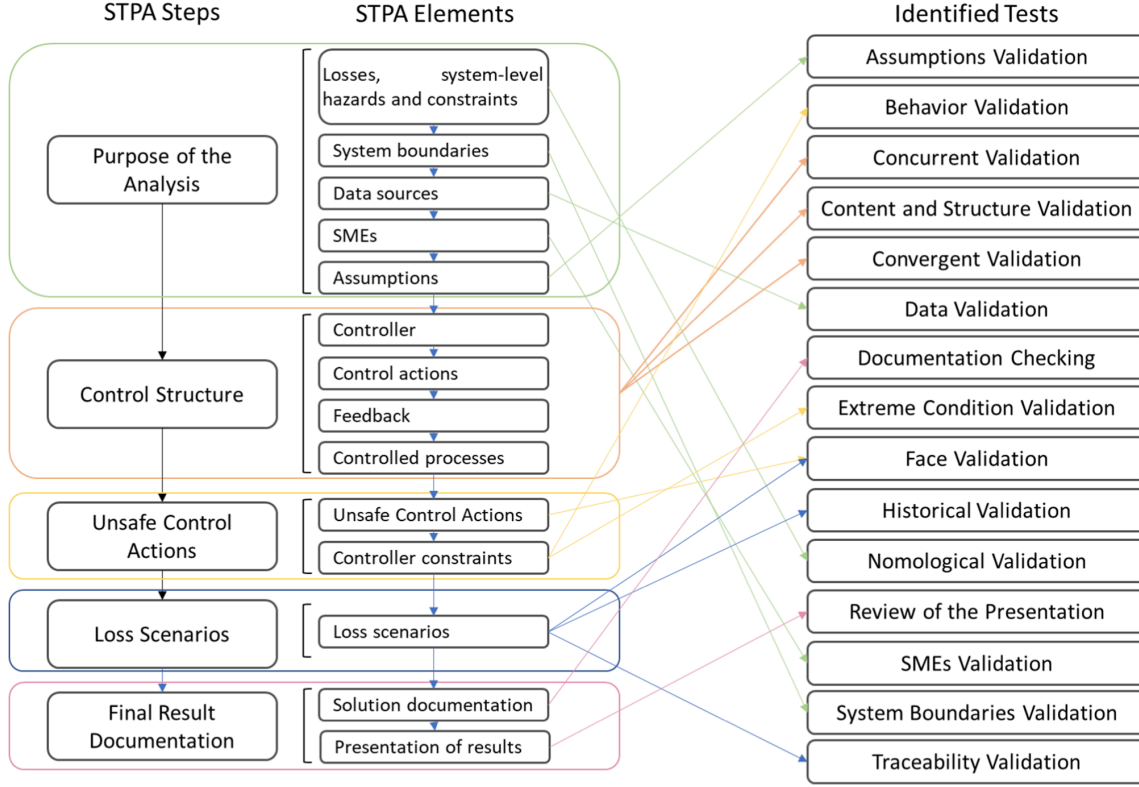


Figure 3.2: Flow Diagram of STPA Process [2]

identified hazards back to specific design features, enhancing its comprehensiveness and utility in safety analysis for complex systems. Eventually, STPA enables the designers to deliver more resilient systems to withstand unexpected disturbances in complex service environments.

3.2 Sequential ADS Safety Analysis with Stacked HAZOP and STPA

The proposed sequential ADS safety analysis initially incorporates stacked HAZOP and STPA methodologies to conduct a qualitative hazard analysis focused on different operational conditions. Subsequently, it generates test cases based on hazard priorities determined through risk assessments using the Automotive Safety Integrity Level (ASIL) table.

The overall schematic of this safety analysis workflow is illustrated in Figure 3.3.

In the proposed ADS safety analysis, HAZOP and STPA are integrated in a stacked manner. The initial phase of this sequential safety analysis adheres to the preparatory requirements of STPA. This involves delineating the scope of analysis for the ADS system, outlining assumptions, crafting a system block diagram, and specifying all pertinent ADS functions ($F = \{F_1, \dots, F_n\}$), their operational contexts ($OC = \{OC_1, \dots, OC_m\}$), and output control actions to other modules ($CA = \{CA_1, \dots, CA_k\}$). In this instance, there are a total of n ADS functions, m applicable operational conditions, and k control actions. Subsequently, a system control structure is established alongside the system block diagram, listed ADS functions, F , and defined control actions, CA . To pinpoint the UCAs of the ADS system, it is imperative to initially comprehend the potential hazards the system could encounter. Leveraging HAZOP's proficiency in identifying and evaluating potential process deviations, each relevant ADS function is scrutinized through HAZOP within defined operational parameters to uncover malfunctions and potential hazardous scenarios. Consequently, under specific operational conditions, l primary hazards ($PH = \{PH_1, \dots, PH_l\}$) are identified stemming from these hazardous scenarios, capable of precipitating a vehicle crash or traffic conflict. Conversely, those scenarios not resulting in a vehicle crash are categorized as secondary hazards (SH).

At this juncture, the primary hazards uncovered through HAZOP are deemed incomplete as they solely assess each ADS function in isolation under various operational conditions to generate malfunctions. This approach overlooks the potential cascading effects across functions within the ADS system, potentially neglecting other hazardous conditions arising from interactions between multiple functions. In contrast, STPA offers a comprehensive perspective by analyzing system-wide component interactions and control mechanisms. Therefore, the malfunction cases identified by HAZOP serve as the basis for identifying UCAs in the third phase of STPA, primarily examining if the deviation of one function can trigger deviations in others within the presumed control structure. By scrutinizing UCAs in the ADS system, additional primary hazards can be unearthed, augmenting the list of hazards, PH as $PH = \{PH_1, \dots, PH_l, \dots, PH_{l+h}\}$ with newly identified ones through the UCA identification process. Subsequently, chains of UCAs among multiple functions leading to a PH are defined as loss scenarios (LS) in the fourth step of STPA. Ultimately, this process yields q loss scenarios in STPA step 4. Importantly, it's crucial to note that the initial l primary hazards identified are not exclusive to all HAZOP cases. As depicted in Figure 3.3, the identified loss scenarios may also trigger the same primary hazards in $PH_1, \dots, PH_l$.

The operational conditions of the studied ADS require thorough investigation to ensure a comprehensive coverage of all potential service environments for each function. In this

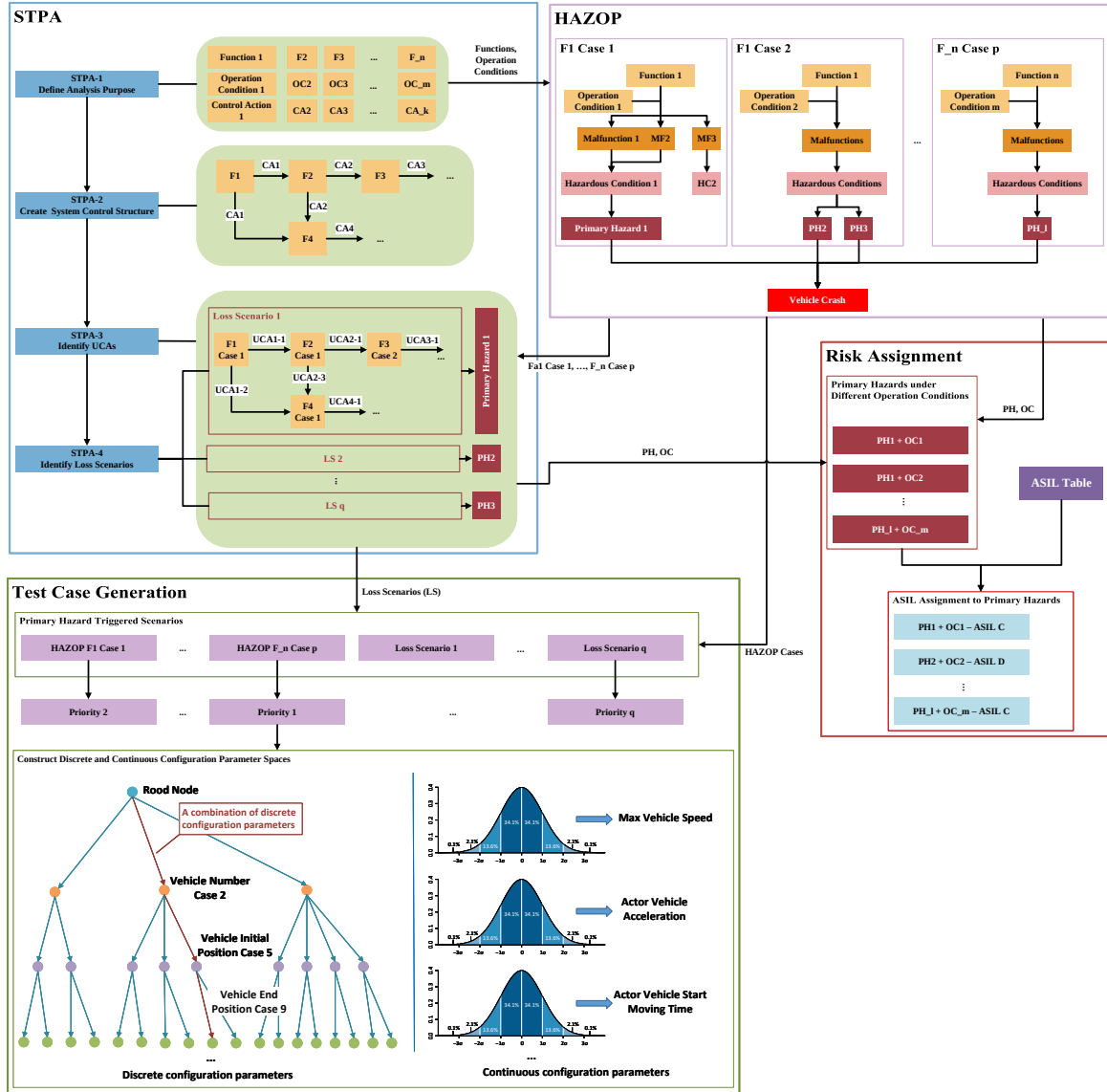


Figure 3.3: Overall Schematic Diagram of Sequential ADS Safety Analysis and Test Case Generation

study, we adhered to the SOTIF scenario derivation guideline to delineate operation conditions for the sequential safety analysis. Specifically, these derived conditions concentrate on

Table 3.1: High-level ODD Taxonomy Categories [6]

Top-level Category	Immediate Subcategory
Physical Infrastructure	Roadway Types
	Roadway Surface
	Roadway Edges
	Roadway Geometry
Operational Constraints	Speed Limit
	Traffic Conditions
Objects	Signage
	Roadway Users
	Non-Roadway User/Obstacles/Objects
Connectivity	Vehicles
	Traffic Density Information
	Remote Fleet Management System
	Infrastructure Sensors and Communications
Environmental Conditions	Weather
	Weather-Induced Roadway Conditions
	Particulate Matter
	Illumination
Zones	Geofencing
	Traffic Management Zones
	School/Construction Zones
	Regions/States
	Interference Zones

scenarios external to the ADS. [6] established a high-level taxonomy for the Operational Design Domain (ODD), identifying 6 overarching categories and 22 immediate subcategories, as illustrated in Table 3.1. [137] further elaborated on this taxonomy, expanding it into a comprehensive list of 200 variables classified into 41 detailed subcategories. In accordance with ISO 21448 standards, scenario variables can be classified utilizing Table 3.1. For example, within the "Physical Infrastructure – Roadway Type" category, sample scenario variables might include "Two-way, divided, unprotected road" or "One-way trafficway."

To organize the variables in Table 3.1 into scenario construction, [137] categorized each

scenario variable as either permanent-regional, permanent-local, a compounding event or condition, or a potential threat:

- **Permanent-regional** – Permanent-regional variables do not change over time or over significant spatial portions of the trip.
- **Permanent-local** – Permanent-local variables persist over time but are localized spatially. Permanent-local variables may only be encountered by an ADS for portions of a trip.
- **Compounding event or condition** - Compounding event or condition variables are temporary events or conditions that may occur within the scenery defined by permanent-regional or permanent-local variables.
- **Threats** – Threats variables are temporary events or conditions that relate to specific roadway threats to which ADS may need to response.

After the operation conditions have been constructed, the risk of the identified primary hazards is assessed in each operation condition. ASIL, which is a risk classification scheme used in automotive industry to assess and mitigate safety risks, is employed to perform the risk assessment to the identified primary hazards. The ASIL assessment first identifies vehicle operational scenarios as previously presented. Then, for each identified vehicle-level hazard, the ISO 26262 risk assessment framework is applied. This involves assessing the probability of exposure to the operational scenario, identifying potential crash scenarios, evaluating the severity of harm to individuals involved in the crash, and considering the controllability of the situation and the vehicle in the potential crash scenario. Based on these factors, ASIL table in ISO 26262 (See Table 3.2) is utilized to correlate the risk of each hazardous condition with an appropriate ASIL level (A, B, C, or D). The ASIL levels are defined based on the severity of hazard consequence, the probability of exposure to the hazard, the feasibility of mitigating the risk through system design or other measures. Higher ASIL levels correspond to more severe consequences and require more stringent safety measures to mitigate the associated risks. In the table, S is severity, E is exposure, C is controllability, and QM is quality management. Finally, the worst-case ASIL is assigned to the hazard, ensuring that safety efforts are appropriately prioritized, and resources allocated effectively to manage risks in alignment with industry standards and regulatory requirements.

After conducting risk assessments on the primary hazards identified in all HAZOP cases and STPA loss scenarios, higher priority is given to cases with greater risk. For

Table 3.2: ASIL Assessment Table [7]

Severity Class	Probability Class (Exposure)	Controllability Class		
		C1	C2	C3
S1	E1	QM	QM	QM
	E2	QM	QM	QM
	E3	QM	QM	A
	E4	QM	A	B
S2	E1	QM	QM	QM
	E2	QM	QM	A
	E3	QM	A	B
	E4	A	B	C
S3	E1	QM	QM	A
	E2	QM	A	B
	E3	A	B	C
	E4	B	C	D

instance, as illustrated in Figure 3.3, the HAZOP case p with function F_n has been assigned the highest priority, indicating that the initial focus of test case generation efforts should be put on this case. Although the test case generation process employs the working principle of combinatorial testing [138], it undergoes several modifications. Beginning with the atomic scenario represented by the root HAZOP case p , the test case generation process first identifies the scenario variables based on its operational conditions OC . The scenario variables like compounding event variables, condition variables, and threat variables are then categorized into discrete or continuous configuration parameters. For instance, discrete configuration parameters may encompass the number of object vehicles, their starting and ending waypoints, requiring discrete values. Conversely, continuous configuration parameters may include maximum vehicle speed, acceleration of actor vehicles, or their starting times, typically falling within a numeric range. Furthermore, continuous configuration parameters are generated based on modeled contributions, utilizing domain knowledge to determine their values. Assuming there are n_d discrete configuration parameters and n_c continuous configuration parameters, they can be mathematically represented

by the following equation:

$$\mathcal{P} = \{P_1, \dots, P_{n_d}, P_{n_d+1}, \dots, P_{n_d+n_c}\} \quad (3.1)$$

For each parameter P_i , if there are m_i possible values, v :

$$\mathcal{V}_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,m_i}\} \quad (3.2)$$

Then combinatorial testing will generate all combinations ($\mathcal{C}$) like:

$$\mathcal{C} = P_1 \times P_2 \times \dots \times P_{n_d+n_c} = \{(v_{i,1}, v_{j,1}, \dots, v_{n,1}), (v_{i,1}, v_{j,1}, \dots, v_{n,2}), \dots, (v_{i,m_i}, v_{j,m_j}, \dots, v_{m_{n_d+n_c}})\} \quad (3.3)$$

where $\times$ represents the Cartesian product between parameters P_i and P_j . Nonetheless, one of the drawbacks of combinatorial testing, as shown in Equation (3.3), is the significant increase in generated combinations as n_d+n_c and m_i increase. To address this issue, unrealistic value combinations are initially defined based on domain knowledge about the HAZOP case p and its OC before combining all configuration parameters. This filtering process considers both range constraints, η_{RC} , and exclusion constraints, η_{EC} . Range constraints, η_{RC} , limit parameter values to specific ranges, with examples such as $v_{i,\min} < v_i < v_{i,\max}$. On the other hand, exclusion constraints, η_{EC} , specify combinations of parameters that are not permitted together, such as $(v_i \neq v_{i,1}) \vee (v_j \neq v_{j,1})$. The resulting valid value combinations can then be written as:

$$\mathcal{C}_{valid} = \mathcal{C} \text{ if } (\eta_{RC} \wedge \eta_{EC}) \quad (3.4)$$

If a primary hazard is triggered in the generated test case, it will be classified as a failure scenario and documented accordingly. Moreover, given the infinite search space of continuous configuration parameters, simply randomly generating numerous test cases may not suffice. Therefore, a coverage-driven test near the nominal level of these continuous parameters is conducted to establish operational boundaries for the relevant ADS. One commonly employed technique in this regard is the 3-sigma rule, a statistical principle delineating the distribution of data within a specified number of standard deviations from the mean in a normal distribution. By subjecting the studied ADS to test cases generated within 3 sigma ranges of their nominal values, it can be deemed statistically safe at a 99.7% confidence level for handling the given HAZOP case or STPA loss scenario. Conversely, failure to meet this criterion results in documenting the test scenario as a "cannot handle scenario," necessitating further testing and the formation operational boundaries for the ADS.

3.3 WATonoBus Vehicle-level Safety Hazard and STPA Analysis

The proposed sequential ADS safety analysis method is then implemented to a real-world ADS platform, WATonoBus, a low-speed autonomous shuttle service operating on Ring Road in the University of Waterloo campus (See Figure 3.4). Following the 4 STPA steps shown in Figure 3.3, this section first outlines the WATonoBus project’s system definition, including scope, assumptions, system block diagram, and component details. It then constructs the control structure within the ADS system. Afterwards, HAZOP is performed to identify hazardous scenarios and primary hazards due to different functional module malfunctions. Following this, STPA then pinpoints unsafe control actions and associated primary hazards. Together with the defined operation conditions and ASIL risk assessment of each hazard, these STPA findings inform the generation of precise test cases for safety verification and design enhancement.



Figure 3.4: WATonoBus Shuttle and Its Operation Path on Ring Road

3.3.1 STPA-1 WATonoBus System Definition

WATonoBus, led by Mechatronics Vehicle Systems Labs (MVSL) at the University of Waterloo, is an autonomous shuttle project with an in-house hardware and software stack.

Its aim is to address challenges in urban autonomous vehicle deployment. This section will define the scope and assumptions as well as describe each subsystem of the WATonoBus project. The focus of this study will be on the holistic system level of the proposed autonomous driving design rather than the technical development and implementation of each functional module.

System Analysis Scope

The scope of this project mainly includes the following aspects, namely service scope, service environment, and available sensing modalities.

Service scope: the WATonoBus project aims to provide students with another transportation option within the campus in the form of autonomous minibus. With the successful launching and running of this project on campus, the autonomous minibus service will also accumulate its commercial value for being deployed in public communities. For now, the service scope of WATonoBus project includes:

- Service location: University of Waterloo campus ring road.
- Service vehicle: Mini shuttle bus.
- Service target: University students and staff.
- Service: On-campus transportation, pullover and merging at bus stops.

Service environment: the WATonoBus is designed to operate in diverse environmental and traffic conditions. Its autonomous functionalities must be robust in the face of the following variables:

- Weather conditions: Sunny, rainy, snowy, foggy.
- Traffic participants: Vehicles, motorcycles, pedestrians, cyclists, geese.
- Road types: Single driveway, all-way stop intersection, potential constructions, bus stop area, pedestrian crossing area, speed bumps.

Available sensing modalities: this project aims to optimize service tasks while minimizing costs, particularly in sensor selection. Currently, the project utilizes the following sensors:

- 3 front and 1 rear camera sensors.
- 1 side camera for merging.
- 1 front LiDAR sensor and 2 blind spot LiDAR sensors.
- 1 GPS sensor.
- 1 front and 1 rear radar sensors.

Interested error: in this work, the interested error will be mainly focused on the SOTIF errors of the designed algorithms. The related perception sensors' malfunctions will also be considered from hardware perspective. However, other hardware components like vehicle actuators or wiring are not within the scope of this report and are assumed to work properly all the time.

Analysis Assumptions

This section enlists several key assumptions regarding the operation of the WATonoBus project. Each assumption is addressed with the explanation to justify its validity in this project.

Prior knowledge of High-Definition (HD) map for localization use: It is assumed that accurate mapping data of the university campus ring road and associated traffic elements are available beforehand and can serve as a localization reference during operations. Such mapping data is commonly utilized not only in various autonomous driving tasks but also in routine driving scenarios, particularly for route planning. Consequently, detailed mapping data of the campus ring road has been gathered and incorporated into this project. Furthermore, static traffic features and signage, such as driveway lanes and intersection stop signs, will also be accessible through the HD map in the current project phase.

Predominant driving scenarios are normal driving, intersection passing, and bus stop pullover and merging: Considering the straightforward yet delimited nature of the campus road infrastructure within the designated service area, the operational scenarios for the ego vehicle are restricted to the aforementioned cases. Throughout its operation, the ego vehicle must adeptly navigate various traffic situations within these three scenarios and execute appropriate maneuvers. The service scope does not necessitate more intricate driving actions such as responding to traffic lights or executing lane changes while overtaking other vehicles.

The project operates at autonomy level 2: In the vehicle, a human operator will be present to assume control if necessary. It is assumed that this operator is fully capable of executing emergency safety protocols during testing and restoring the vehicle to a safe state. Given the defined scope of accessible sensors and considerations for public safety, the project aims to achieve a Level 2 automation vehicle in this phase. The pursuit of higher levels of automation will be considered in subsequent project phases.

System Block Diagram

The high-level WATonoBus system block diagram is illustrated in Figure 3.5, including 5 major functional modules, namely calibrated sensors, localization, perception and prediction, decision-making and planning, and control modules. These components will be described in the following paragraphs in detail.

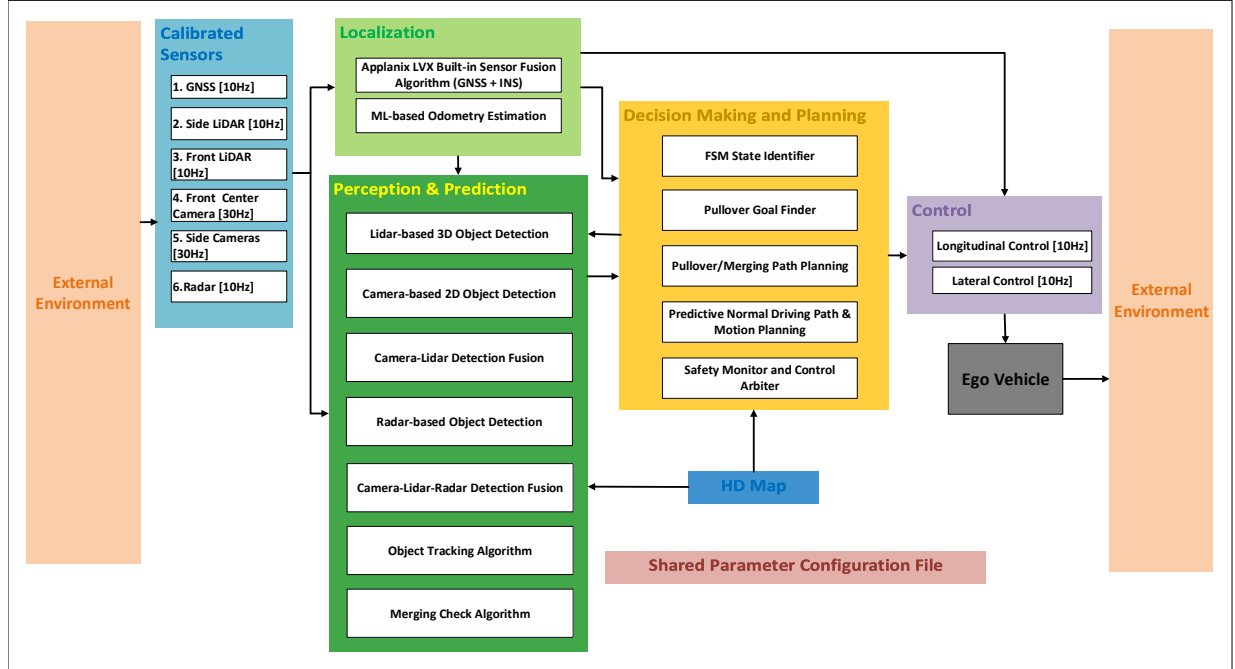


Figure 3.5: Overall Schematic Diagram of WATonoBus

Calibrated sensors on WATonoBus capture raw sensor measurements for localization and perception modules. This includes 3 front and 1 rear cameras for visual data capture, supplemented by 1 side camera for monitoring rear left traffic for merging purposes. A

front LiDAR sensor aids in depth perception. Two side LiDAR sensors prevent blind spot collisions. Front and rear radar sensors assist in object tracking and prediction.

Localization module on WATonoBus handles ego vehicle positioning with respect to the global frame and captures static traffic features like traffic signs or pedestrian crossing lanes. It employs dual techniques for redundancy: Applanix GPS sensor fusion algorithms for precise localization during normal driving, and a data-driven visual odometry method using LiDAR point cloud measurements for basic localization while using safety fallback strategies. The inputs and outputs of this functional module are outlined below:

- Inputs: GPS data, LiDAR data
- Outputs: Ego vehicle location/orientation, nearby traffic features' locations

Perception and Prediction module provides scene awareness through information obtained from various sensors. Particularly, combining 2D camera and 3D LiDAR detections with radar measurements, several learning-based algorithms have been developed to both identify the 3D locations of the surrounding dynamic objects and classify the detected objects associated with certain confidence level. The potential dynamic objects include pedestrians, vehicles, buses and so on. Object tracking and motion prediction will then be performed for the perceived objects. These results are then used to create a time series of drivable spaces for predictive decision-making process. The inputs and outputs of this functional module are listed as below:

- Inputs: LiDAR data, camera data, radar data, ego vehicle location/orientation
- Outputs: Dynamic objects 3D locations, estimated dynamic objects' orientations, future location, and velocities

Decision-making and planning module manages tactical planning based on the surrounding environment. It employs a Finite State Machine (FSM) to determine the current driving state (normal driving, intersection passage, or bus stop pull-over/merging). For pull-over and merging, specific routes are planned to ensure safety. In addition, the obstacles, which may have the conflict with the ego vehicle, will be filtered from all detected dynamic objects in this module so that the potential traffic hazards can be identified before planning the ego vehicle trajectory and corresponding motion. In normal driving, short-term ego path and velocity profile are iteratively planned based on drivable spaces. The inputs and outputs of this functional module are listed as below:

- Inputs: Estimated dynamic objects' future location and velocities, dynamic objects 3D locations, ego vehicle location, nearby traffic features' locations, Safety speed and steering angle constraints.
- Outputs: High-level behavior decisions, ego vehicle trajectory and speed profiles

Control module employs both longitudinal and lateral Model Predictive Control (MPC). These controllers are finely tuned to track the planned path and velocity profile, providing desired inputs for the vehicle's mechanical actuators.

- Inputs: Initial velocity, ego vehicle trajectory and speed profiles, safety distance threshold to obstacles
- Outputs: Steering angle adjustment, gas pedal adjustment

3.3.2 STPA-2 System Control Structure of WATonoBus

In STPA, the autonomous driving system is analyzed as a composition of controllers responsible for maintaining vehicle safety in hazardous conditions. STPA aims to pinpoint inadequate control actions and their root causes, thereby offering insights for system enhancements. Given the innate complexity of the autonomous driving system, it integrates multiple sub-controllers that administer discrete or continuous control actions. Drawing from the system block diagram depicted in Fig. 3.5, this section will meticulously delineate these sub-controllers and their associated control actions. Although human operator is normally included in STPA, we assume the operator is able to take over WATonoBus all the time and will not generate or be affected by any unsafe control actions. Therefore, human operator is not included in the following STPA.

The initial sub-controller comprises the decision-making module, responsible for generating discrete, high-level vehicle motion commands. These directives guide the motion planning and control modules in orchestrating detailed adjustments to the vehicle dynamics. The decision-making module executes specific control actions, which are shown in Table 3.3.

The second sub-controller integrates both the motion planning and vehicle control functional modules, tasked with executing precise, low-level adjustments to the vehicle dynamics to adhere to the desired trajectory. This controller bridges what the autonomous driving system perceives and analyzes with the physical actuators, enabling the ego vehicle to engage with the external environment. Specific control requests and actions of this sub-controller are tabulated in Table 3.4.

Table 3.3: Control Actions of Decision-making Module in WATonoBus

Functional Module (FM)	Context for Control Request	Control Actions (CA)
FM1: Decision Making and Planning Module	General vehicle motion request at certain traffic scenarios	FM1CA1: Stop FM1CA2: Decelerate FM1CA3: Go FM1CA4: Pull over FM1CA5: Merge

Table 3.4: Control Actions of Motion Planning and Vehicle Control Modules in WATonoBus

Functional Module (FM)	Context for Control Request	Control Actions (CA)
FM2: Control Module	Lateral position adjustment request based on the planned trajectories.	FM2CA1: None FM2CA2: Lateral adjustment δ toward trajectory with steering angle modification
	Longitudinal position adjustment request based on the planned trajectories	FM2CA3: None FM2CA4: Longitudinal adjustment δ forward gas/brake pedal modification

The third sub-controller, situated within the decision-making subsystem, is the safety monitor functional module. Its primary responsibility is to determine appropriate safety strategies based on inputs from monitoring the vehicle’s status. These selected safety strategies are subsequently transmitted to the control subsystem to ensure the ego vehicle always remains in a safe state. Notably, these safety strategies consist of discrete control actions that either adjust or restrict the ego vehicle’s operational mode. Specific control requests and actions of this sub-controller are listed in Table 3.5.

Table 3.5: Control Actions of Safety Monitor Module in WATonoBus

Functional Module (FM)	Context for Control Request	Control Actions (CA)
FM3: Safety Strategy Selection in Decision Making Module	Safety state request based on vehicle status monitoring results	FM3CA1: Normal autonomous driving FM3CA2: Limited autonomous driving functionalities FM3CA3: Manual control

The final sub-controller is the control mode arbiter functional module, responsible for

determining the necessary operational mode and transmitting the corresponding vehicle control inputs to the vehicle components. Consequently, the specific control actions of this sub-controller are shown in Table 3.6.

Table 3.6: Control Actions of Control Mode Arbiter Module in WATonoBus

Functional Module (FM)	Context for Control Request	Control Actions (CA)
FM4: Control mode arbiter in control module	Vehicle operational mode confirmation request before implementing vehicle control	FM4CA1: Engage autonomous driving mode FM4CA2: Engage manual control mode

Furthermore, alongside the sub-controllers outlined in the WATonoBus software stack, a complementary control structure diagram has been devised from the hardware standpoint, as depicted in Figure 3.6. Within this diagram, the use of AND and OR gates with block diagrams delineates the control action workflow from individual sensor components to the safety monitor function module, and ultimately to the control mode arbiter module for WATonoBus.

Referring to Figure 3.6, the hardware sensors initially perceive the surrounding environment, after which their raw measurements undergo examination against predefined thresholds or criteria. Following the assessment of each sensor, the collective status of WATonoBus sensors is synthesized through the utilization of AND and OR gates. This aggregated sensor status then triggers a sequence of determinations regarding the ego vehicle’s driving mode, safety state, and Human Machine Interface (HMI) LED indication. Subsequently, predicated upon these determinations, the schematic disparities between manual and autonomous operation are delineated within the figure, spanning from the vehicle Controller Area Network (CAN) to the Microcontroller Unit (MCU), and the control mode arbiter module. A detailed explication of each primary component depicted in Figure 3.6 is presented in the subsequent paragraphs.

Sensor status examination: In this study, the hardware sensor suite comprises Applanix GPS, Inertial Measurement Unit (IMU), Distance Measurement Indicator (DMI), Global Navigation Satellite System (GNSS), GNSS Azimuth Measurement System (GAMS), LiDAR, and camera sensors (refer to Figures 3.7 and 3.8). For the localization sensors, the raw measurements, namely the accuracy of position, IMU, GAMS, DMI, and GNSS working status, undergo an initial examination against predefined continuous or discrete thresholds to determine their acceptability. If the initial checks reveal inaccuracies, two additional temporal criteria are employed to assess whether these inaccuracies persist over

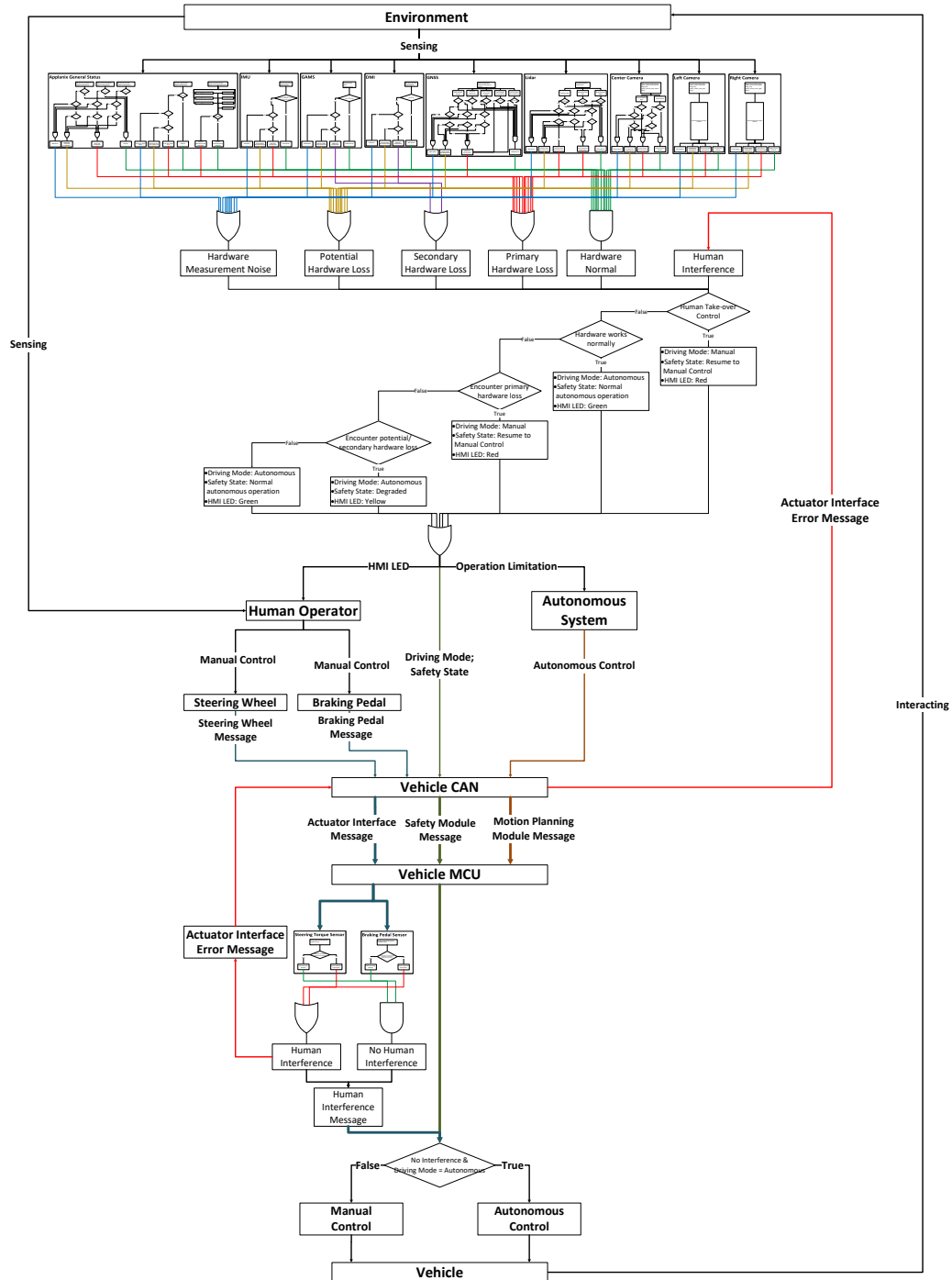


Figure 3.6: Sensor Hardware Monitor and Control Mode Arbiter Workflow of WATonoBus

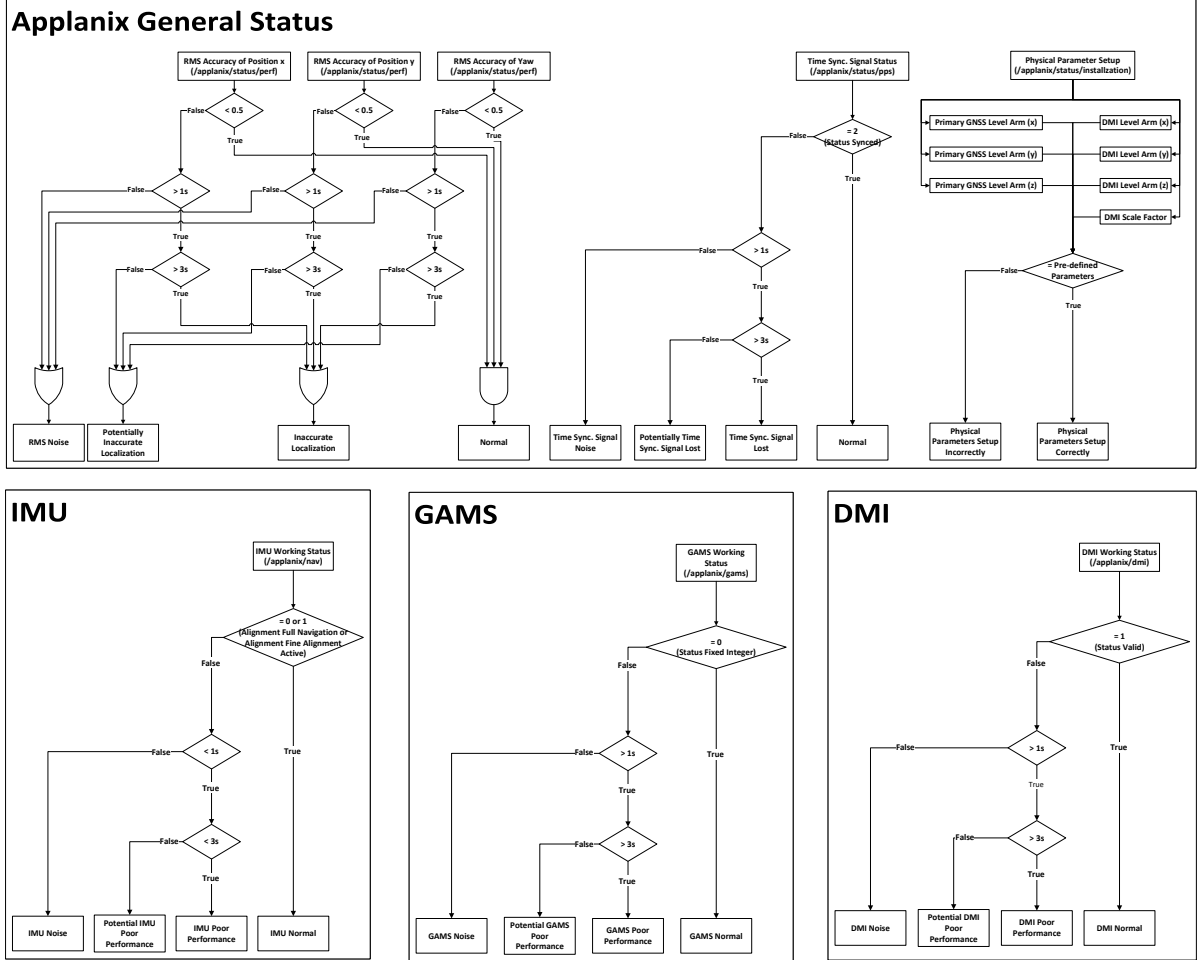


Figure 3.7: Sensor Components and Their Checking Criteria (Part 1)

time, with the temporal thresholds set at $> 1s$ and $> 3s$. Based on the temporal evaluations, sensor status is categorized as sensor noise, potentially inaccurate sensor measurement, or inaccurate sensor measurement. Additionally, Applanix's general status is verified through checks on time signal synchronization and physical parameter setup against specific criteria to ensure synchronized localization sensor signals with proper physical model parameterization. Likewise, diagnostic messages from front and blind spot LiDAR sensors are scrutinized to assess their operational status. For camera sensors, apart from examining camera status, sampling rates are evaluated against predefined thresholds. Subsequently, temporal criteria are similarly applied to LiDAR and camera sensor checks.

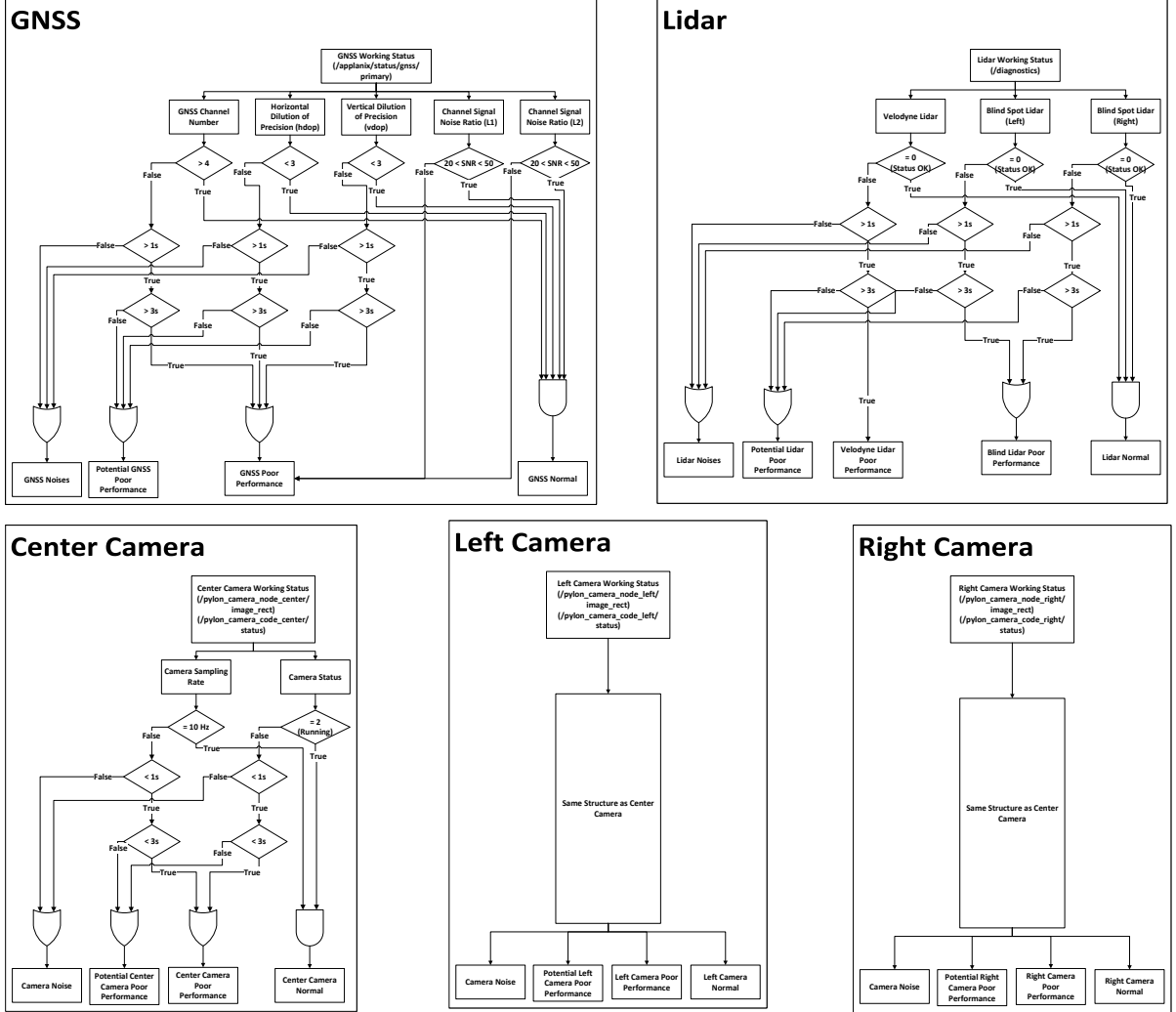


Figure 3.8: Sensor Components and Their Checking Criteria (Part 2)

Sensor status aggregation and ego vehicle status determination: After each sensor status is determined, these sensor status information will then be used to determine the overall sensor suite status by categorizing the status to hardware working normally, primary or secondary hardware loss, potential hardware loss or experiencing hardware measurement noise (See Figure 3.9). If all sensor are working normally, then the overall sensor suite status will be hardware working normally. However, if one of the sensor status is showing inaccurate or poor performance, primary hardware loss will then dominate the overall

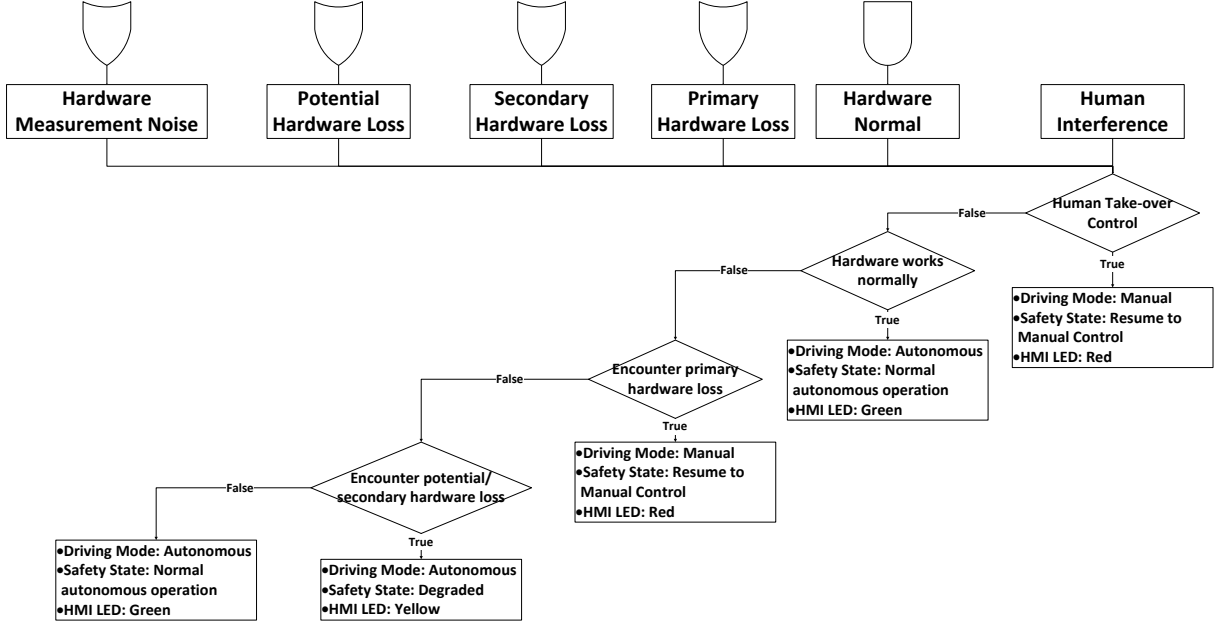


Figure 3.9: Ego Vehicle Status Determination Workflow

sensor suite status, except for DMI or GAMS poor performance. Given DMI and GAMS are secondary information in localization process, their poor performance status will only trigger secondary hardware loss. Similarly, hardware measurement noise is triggered by at least one sensor noise status and other sensors are working normally. Apart previously mentioned hardware status, human interference is also included for manual control situation. Afterwards, according to the status priority, a series of determinations are done from considering whether human operator is taking over control or not to whether sensor suite is experiencing sensor noises or not. Accordingly, the ego vehicle driving mode, safety state, and HMI LED status are decided. Possible ego vehicle driving modes and safety states are tabulated in Table 3.6 and 3.5 before. In terms of HMI LED light, three colors, red, yellow, and green are chosen to show manual, degraded autonomous and fully autonomous driving modes, respectively.

Control mode arbiter logic and driving mode comparison: If ego vehicle is in manual driving mode, the vehicle control messages will be directly obtained from driver's wheel steering and pedal pressing angles and sent through vehicle CAN to MCU. On the other hand, the autonomous driving mode will generate autonomous control messages from the ADS software stack and send them through vehicle CAN as motion planning messages. For manual driving mode, actuator interface is also examined for indicating whether human

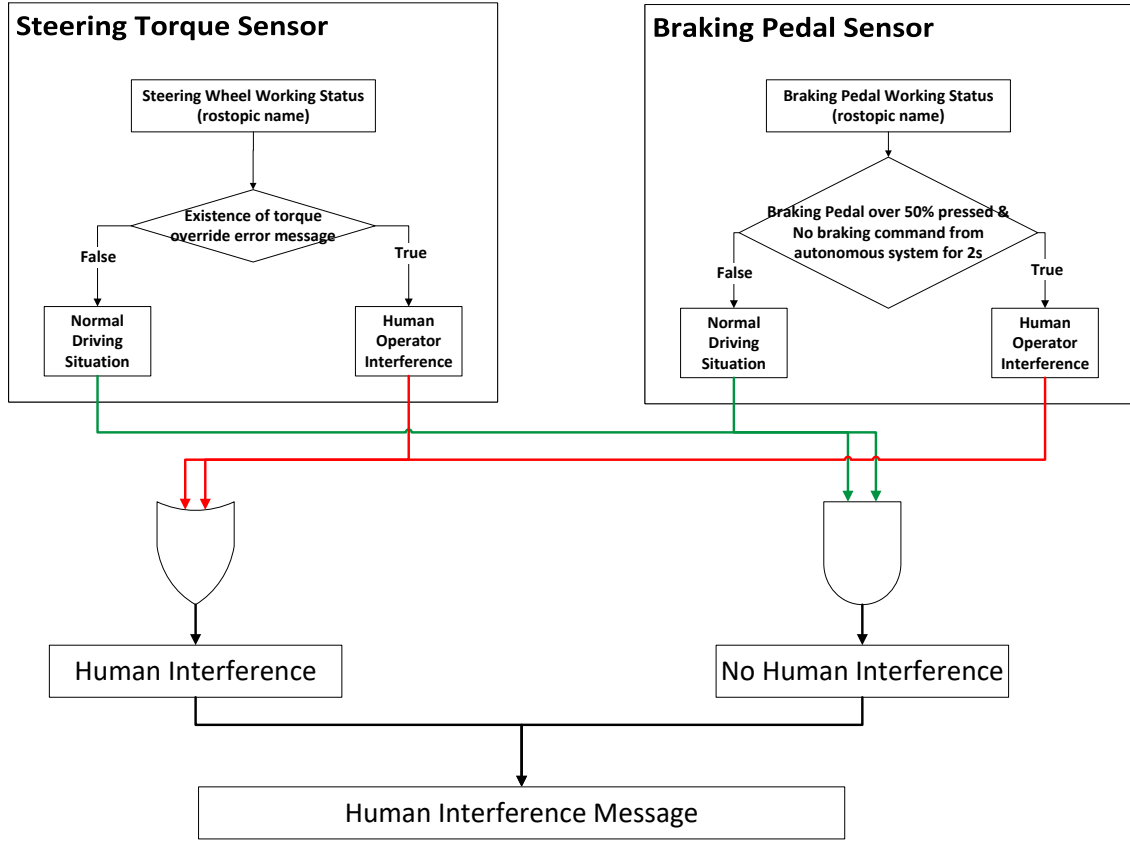


Figure 3.10: Actuator Interface Message Generation Logic of WATonoBus

operator is driving the ego vehicle or not. The actuator interface examination logic is demonstrated in Figure 3.10. Once human interference is confirmed through actuator interface examination, this message will be sent to control mode arbiter module to override autonomous control mode and through vehicle CAN to the previous ego vehicle status determination logic.

3.3.3 HAZOP Analysis of WATonoBus

The Hazard and Operability Study (HAZOP) is performed to identify the potential hazards associated with the operation of the designed ADS. The following paragraphs will demonstrate the system functions, hazard derivations, and summarize the found vehicle-level

hazards.

With the assumptions and scope defined in the Section 3.3.1, the major functions of WATonoBus are listed in Table 3.7:

Table 3.7: ADS Functions of WATonoBus

Autonomous Driving Subsystem	Functions
Calibrated Sensors	• Sense surrounding environment.
Perception	• Localize ego vehicle. • Localize existing traffic objects. • Predict dynamic traffic objects' trajectories.
Decision Making & Motion Planning	• Instruct ego vehicle motion. • Identify potential traffic obstacles. • Arrange vehicle dynamics adjustments.
Control	• Execute low-level vehicle control. • Implement required vehicle control mode.
Safety Monitor & Control Model Arbiter	• Assess autonomous vehicle performance. • Provide safety measures.

According to the detailed description of each WATonoBus functional module, malfunctions and related hazardous conditions are identified using HAZOP guidewords. The employed HAZOP guidewords include: “No or not”, “Part of”, “More than intended”, “Less than intended”, “Later function performance”, “Incorrect performance”. Overall, 35 malfunctions are identified from the 11 functions listed in Table 3.7. Among the analysis of these 35 malfunctions, there are 13 hazardous conditions generated from this HAZOP analysis and they are tabulated in Table 3.8. Please refer to Appendix A for the detailed HAZOP study results.

Of these 11 hazardous conditions, a deeper analysis is conducted to ascertain that there are no causal relationships among the primary hazardous conditions. Consequently, hazardous conditions H2-4 and H7-9 are eliminated from consideration as they can indirectly trigger H1 or H6. Thus, the primary vehicle-level hazardous conditions, which can directly lead to a vehicle collision, are outlined in the Table 3.9. The rest of hazardous conditions are then categorized as secondary hazardous conditions in Table 3.10.

Table 3.8: HAZOP Study Results

	Hazardous Condition	Description
H1	Unable to adjust vehicle control autonomously	The ego vehicle is not capable of properly perceiving or analyzing its surrounding environment due to the malfunctions of system hardware or software. No reference can be interpreted to output to adjust the ego vehicle's motion in autonomous driving mode correspondingly.
H2	Unable to detect nearby traffic objects	The ego vehicle's perception module cannot properly detect other traffic participants like pedestrians or vehicles due to the malfunctions of system hardware or software.
H3	Unable to estimate distance to dynamic objects	The ego vehicle's localization module is not capable of correctly localizing the ego vehicle and making the calculation of relative distances between other traffic participants with ego vehicle impossible.
H4	Unable to understand dynamic objects' intention	The ego vehicle's prediction module cannot properly predict other traffic participants' intent due to the malfunctions of system hardware or software.
H5	Unable to proceed safety fallback strategy	The ego vehicle's control mode arbiter is not capable of modifying the vehicle control mode between autonomous driving and manual control and making the implementation of fallback strategy impossible.
H6	Incorrectly adjust vehicle control autonomously	The ego vehicle is not capable of accurately following the current traffic rules due to the poor-quality results generated from system hardware or software. Coarse reference information will influence the vehicle dynamics adjustment in autonomous driving mode consequently.
H7	Omit certain nearby traffic objects	The ego vehicle's perception module cannot completely identify all nearby traffic objects due to the malfunctions of system hardware or software.

H8	Incorrect estimation on distance to dynamic objects	The ego vehicle’s localization module is not capable of accurately localizing the ego vehicle, resulting in the coarse estimation on relative distances between other traffic participants and the ego vehicles.
H9	Incorrect understanding of dynamic objects’ intention	The ego vehicle’s prediction module is not capable of accurately estimating other traffic participants’ intents due to the poor performance from system software.
H10	Incorrectly implement safety strategies	The ego vehicle is not capable of accurately executing corresponding safety strategy and maintain the ego vehicle within the safe state due to the poor performance of the Reliability and Safety subsystem.
H11	Slow reaction to surrounding environment	One or more functional modules in the designed autonomous driving system cannot efficiently process the algorithms and result in long latent time to the environment changes.

Table 3.9: Primary Hazards Table

ID	Hazardous Conditions
PH1	Unable to adjust vehicle control autonomously
PH2	Unable to proceed safety fallback strategy
PH3	Incorrectly adjust vehicle control autonomously
PH4	Incorrectly implement safety strategies
PH5	Slow reaction to surrounding environment

3.3.4 STPA-3 Identification of UCAs of WATonoBus

In this section, combining with the primary vehicle-level hazards in Table 3.9, a set of guidewords are employed to identify the Unsafe Control Actions (UCAs) so that other vehicle-level hazardous conditions can then be determined. The primary vehicle-level hazardous conditions triggered by the UCAs will be first selected from those listed in Section 3.3.2. The guidewords are listed as below:

Table 3.10: Secondary Hazards Table

ID	Hazardous Conditions
SH1	Unable to detect the nearby traffic objects
SH2	Unable to estimate the distance to dynamic objects
SH3	Unable to understand the dynamic objects' intention
SH4	Omit certain nearby traffic objects
SH5	Incorrect estimation on the distance to dynamic objects
SH6	Incorrect understanding of the dynamic objects' intention

- Control actions provided:
 - But not needed.
 - Needed but intensity is incorrect.
 - Needed but delivered incorrectly.
 - Needed but start early or late.
- No control action provided but needed.

The generated UCAs for each sub-controller can be found in Appendix B in details. Table 3.11 exemplifies the analysis of the vehicle control module. For instance, the bold PH3 implies that when there's a lateral adjustment request but no longitudinal request, the module provides control commands in both directions, resulting in incorrect adjustment of vehicle control.

The Causal Factors (CF) for each UCA are then identified by analyzing each subsystem and the interactions between different subsystems. According to CFs and UCAs, safety measures can then be executed correspondingly to control or mitigate UCAs in ADS. The guidewords employed in this study are demonstrated in the block diagram Figure 3.11 [3]. Given the different natures of the sub-controllers, some guidewords will be applicable. The detail explanations of the guidewords in the Figure 3.11 have been tabulated in Table 3.12.

Table 3.11: Vehicle Control Functional Modules' UCAs

Context for Control Request		Guide-words for Assessing Control Actions				
Longitudinal position adjustment requests	Lateral position adjustment requests	Provided but not needed	Provided but more than intended	Provided but less than intended	Provided but too late	Not provided but needed
FM2CA3	FM2CA1	PH3	N/A	N/A	N/A	N/A
FM2CA3	FM2CA2	PH3	PH3	PH3	PH3, PH5	PH1
FM2CA4	FM2CA1	PH3	PH3	PH3	PH3, PH5	PH1
FM2CA4	FM2CA2	N/A	PH3	PH3	PH3, PH5	PH1

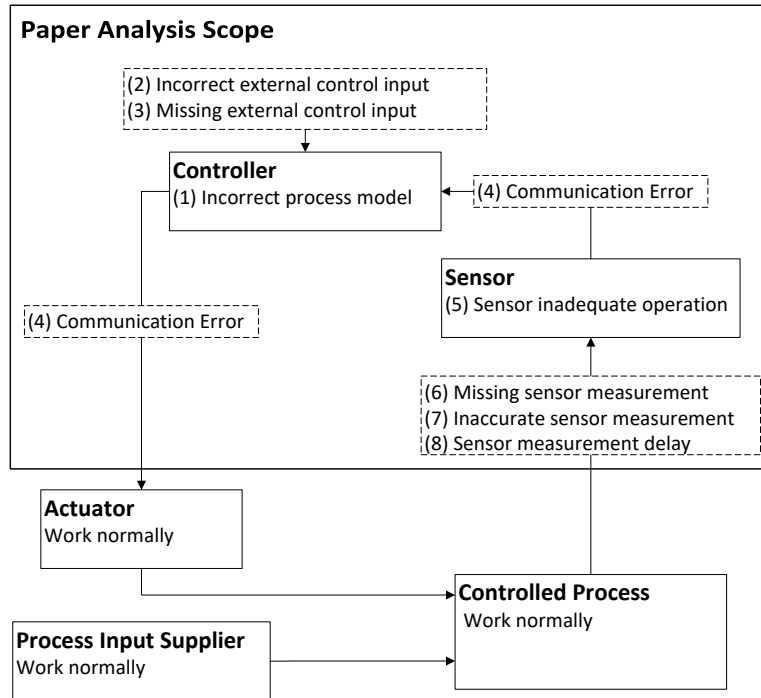


Figure 3.11: Causal Factors Considered in STPA [3]

Table 3.12: Causal Factor for Implemented Sub-controllers

Explanation
(1) Incorrect process model - Continuous control actions: Incorrect model of controlled process or plant - Discrete control actions: Incorrect decision-making control action selection strategy
(2) Incorrect external control input - Poor performance of input functional module - Corrupted signal - Incorrect input signal frequency - Wrong input signal format
(3) Missing external control input - Input functional module breakdown - Inefficient controller algorithm
(4) Communication Error - Failure of message generator or receiver - Incorrect signal priority setting
(5) Sensor inadequate operation - Internal sensor hardware failure - Performance degradation over time - Poor performance due to extreme temperature
(6) Missing sensor measurement - Sensor inputs blocked externally by moisture or dirt - Loss of power supply or disconnection
(7) Inaccurate sensor measurement - External environmental variabilities: weather, light, traffic participants' characteristics, etc. - Sensing limitation: distance, view angle, light sensitivity, scan rate, etc. - Ego vehicle instability: vibration or shock impact
(8) Sensor measurement delay - Incorrect sampling rate and report frequency - Low computation processing capability

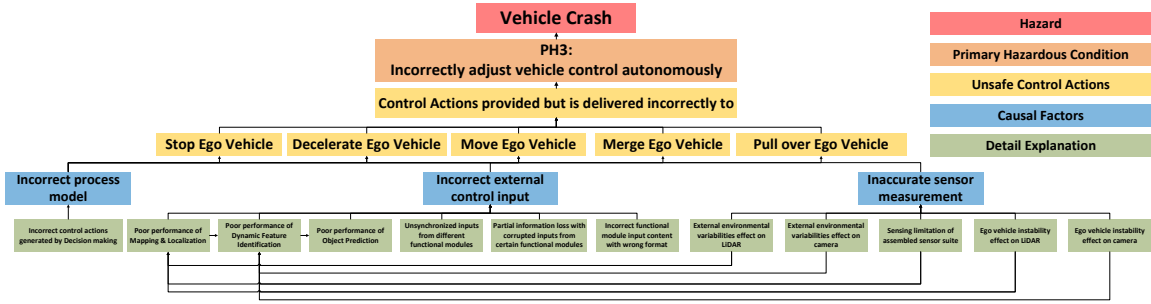


Figure 3.12: STPA Traceability Diagram Example of PH3

3.3.5 STPA-4 Identification of Loss Scenarios

Ultimately, an overall STPA traceability block diagram is constructed, integrating the identified loss scenarios. In this STPA analysis, all primary hazards ultimately lead to a vehicle crash. Hence, the STPA traceability diagram positions "vehicle crash" at the top, succeeded by primary hazardous conditions, unsafe control actions, and their respective causal factors, alongside explanatory details. The directional arrows signify the causal relationships, delineating the STPA loss scenarios within the diagram. Leveraging this STPA traceability diagram, safety strategies can be devised to preempt hazardous conditions. Moreover, the traceability established between hazardous conditions and their causal factors can aid in pinpointing defects within designed functional modules after the occurrence of hazardous conditions. The comprehensive STPA traceability block diagram is depicted in Figure 3.13.

An excerpt from the STPA traceability diagram, focusing on one branch under PH3, is illustrated in Fig. 3.12 as a representative example. Here, the primary hazard arises from erroneously delivered control actions attributed to incorrect model processes, external control inputs, or imprecise sensor measurements. Each of these causal factors is elaborated upon with detailed explanations regarding their root causes, often exhibiting interdependencies that can propagate from one factor to another. For instance, in this scenario, the limitations of the sensor suite's sensing capabilities can directly impact the performance of dynamic feature identification. This diagram serves as a foundation for generating test cases and formulating safety strategies aimed at averting the corresponding hazardous conditions.

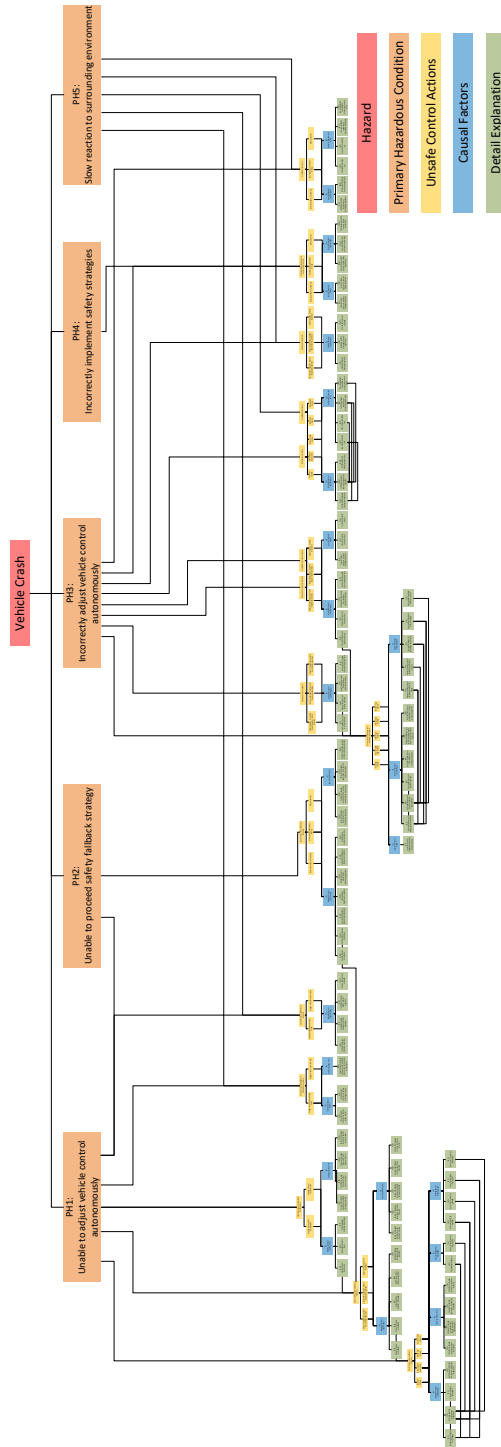


Figure 3.13: Overall STPA Traceability Diagram of WATonoBus

3.4 WATonoBus Risk Assessment and Vehicle-level Safety Goals

To mitigate unreasonable risks as outlined in ISO 26262, an Automotive Safety Integrity Level (ASIL) will be designated to each identified vehicle-level primary hazardous condition, following a thorough analysis of its exposure, severity, and controllability. The operation conditions for WATonoBus are initially delineated based on the provided service scope and environment, in accordance with the specifications detailed in Table 3.1. The hazardous conditions will then be integrated with the potential operation conditions. The same hazardous condition may obtain different ASILs under different operation scenarios. Finally, the vehicle-level safety goals of WATonoBus are listed as the fundamental reference for later edge case generation and simulation testing process.

3.4.1 WATonoBus Operation Conditions

Based on the project scope described in Section 3.3.1, the operation scenarios should consist of the possible traffic situations that WATonoBus will encounter in the real-world, Ring Road testing. The considered scenarios variables are tabulated in Table 3.13. Based on the variables defined in Table 3.13, the ego vehicle operation conditions are enumerated through combining the permanent-reginal, permanent-local, condition and threat variables. Eventually, 135 combinations of operation conditions are developed and can be found in Appendix C. One example operation condition is demonstrated as below:

- Operation Condition 1: Ego vehicle drives on the local, two-way, divided, unprotected roadway with single lane on one direction. The roadway is paved with concrete and is level. The roadway has no restriction to pedestrians and cyclists and the speed limit is less than 30km/h. The ego vehicle is now at a T-shaped/four-way intersection with no restriction. Pedestrians exist in the intersection. The weather is sunny, and daylight is enough. However, one disabled or crashed vehicle present at the intersection.

3.4.2 ASIL Assessment for WATonoBus Primary Hazards

After the operation scenarios have been identified, the ASIL is then assigned to each primary hazardous condition in each operation scenario. Based on Table 3.2, the ASIL

Table 3.13: Variables Covered in WATonoBus Operation Conditions

Top-Level Category	Permanent-Regional Variable	States
Physical Infrastructure	Roadway Type	Local, Two-way, Divided, Unprotected
	Lane Type	Single lane
	Roadway Surface	Paved concrete
	Roadway Grade	Level
Objects	Non-vehicle User Permission on Roadway	- Unrestricted to pedestrian - Unrestricted to cyclist
	Presence of Pedestrian Crosswalk/Intersections	- Unrestricted crosswalks or Unrestricted intersections
Operational Constraint	Speed Limit	Low Speed ($V < 30\text{km/h}$)
Top-Level Category	Permanent-Local Variable	States
Physical Infrastructure	Intersection	- T-intersection - Four-way intersection
	Roadway Geometry	Straight or Curved
Top-Level Category	Condition Variable	States
Physical Infrastructure	Roadway Condition	Bumps on road
	Lane Edge	Cones existence
Objects	Roadway Users	Standard vehicles in roadway
	Non-standard Vehicles	Large vehicles (i.e. bus, truck)
	Pedestrians and cyclists	- In roadway - In intersection - In crosswalk - Wait to cross roadway - Move along or adjacent to roadway
Environment Condition	Weather	Sunny or Medium rain/snow/fog
	Weather-induced Surface Condition	Wet, Snow, or Slush
	Light Conditions	Daylight or Dark with lighting condition
Top-Level Category	Threat Variable	States
Operational Constraints	Altered Traffic Flow	Disabled vehicle or Nearby crash scene
Objects	Roadway Users	- Other vehicle travels on opposite roadway - Other vehicle disobeys stop sign at intersection - Other vehicle fails to yield right-of-way - Large objects in road - Pedestrians or cyclists improperly cross roadway - Pedestrians running towards roadway

Table 3.14: WATonoBus Primary Hazards ASIL Assessment

	Primary Hazardous Condition	ASIL
PH1	Unable to adjust vehicle control autonomously	B
PH2	Unable to proceed safety fallback strategy	D
PH3	Incorrectly adjust vehicle control autonomously	B
PH4	Incorrectly implement safety strategies	C
PH5	Slow reaction to surrounding environment	B

assignment evaluates the aspects of severity, exposure, and controllability in one interested operation condition with one primary hazardous condition. Vehicle crashes will be considered as the ultimate hazard. The ASIL assessment rationales are based on similar studies presented in [137] and [3]. Referring to the ASIL assessment tables in Appendix D, the WATonoBus primary hazards ASIL assessment results are tabulated in Table 3.14. Appendix E lists the detailed ASIL assessment and rationales, considering the primary hazards under each operation scenario.

3.4.3 WATonoBus Vehicle-level Safety Goals

Vehicle-level safety goals represent the pinnacle safety imperatives for ADS, encapsulating the overarching safety requirements essential for ensuring operational integrity and safeguarding against potential hazards. Each safety goal is aligned with a designated ASIL, mirroring the ASIL assessment of the primary hazards. These safety objectives serve as the guiding principles that underpin the design, development, and deployment of WATonoBus, outlining the fundamental benchmarks necessary to achieve and maintain safety performance during operation. Table 3.15 below illustrates WATonoBus vehicle-level safety goals.

Table 3.15: WATonoBus Vehicle-level Safety Goals

	Safety Goals	ASIL
SG1	Capable of correctly adjust vehicle control autonomously	B
SG2	Capable of correctly proceed safety fallback strategies	D
SG3	Ensure on-time reaction to the changes of surrounding environment	B

3.5 WATonoBus Edge Case Generation

Based on the defined safety goals, identified loss scenarios, and HAZOP cases discussed in the previous sections, this thesis proceeds to generate effective test cases for WATonoBus simulation testing. These test cases will help determine the operational boundaries of WATonoBus to ensure it maintains the defined safety goals during operation. In this work, the generated test cases primarily examine the decision-making and motion planning modules and will be used in Chapter 5 to evaluate the proposed uncertainty-aware decision-making mechanism. The test cases focus on interactions with pedestrians while performing lane-following maneuvers. Considering the operational conditions summarized in Section 3.4.1 and shown in Appendix C, the focused operation conditions in this work will be 94 or 95.

- Operation Condition 94/95: Ego vehicle drives on a straight roadway with single lane on one direction. The roadway is paved with concrete and is level. The roadway has no restriction to pedestrians and cyclists and the speed limit is less than 30km/h. Pedestrians exist near the roadway. The weather is sunny, and daylight is enough to obtain sufficient visibility for perception algorithm. However, pedestrians exhibit running behavior towards roadway (i.e., *OC94*), or pedestrians exhibit normal behaviors without threat variables like walking on sidewalks, crossing, or crossing then returning (i.e., *OC95*).

These two relevant operating conditions are expanded to 9 atomic scenarios, as shown in Table 3.16, considering interactions between a lane-following ADS and both single and multiple pedestrians. Each atomic scenario is thoroughly analyzed and broken down into several configuration parameters, following the workflow illustrated in Figure 3.3. The resulting configuration parameters are listed below as an example. The range and number of these parameters are flexible and case dependent.

- Discrete parameters (*DP*):
 - DP_1 - Number of actor pedestrians: $\{1, 2, 3, 4, 5\}$.
 - DP_2 - Pedestrians spawn side: $\{left, right\}$ to the driving lane.
 - DP_3 - Pedestrians speed: $\{1.5, 2.7, 3.5\} m/s$.
 - DP_4 - Pedestrians starting point distance to ego vehicle: $\{25, 45\} m$ away.
 - DP_5 - Pedestrians turning angle during crossing: $\{45, 90\} degrees$.

- DP_6 - Ego vehicle speed: $\{20, 30\}$ km/h
- Continuous parameters (CP):
 - CP_1 - True object position uncertainty in simulation: normal distribution with 0 mean and standard deviation of $\{1.5, 2, 2.5\}$ m .

Table 3.16: Generated Test Cases for WATonoBus Operation Conditions 94 and 95

	Scenario Description	Parameters	Generated Case Num
1	Running pedestrian	$DP_1 = 1, DP_2, DP_3 = \{2.7, 3.5\}m/s, DP_4, DP_6, CP_1$	48
2	Walking on sidewalk in parallel to ego and then crossing	$DP_1 = 1, DP_2, DP_3 = 1.5m/s, DP_4, DP_6, CP_1$	24
3	Returning pedestrian	$DP_1 = 1, DP_3 = \{1.5, 2.7\}m/s, DP_4, DP_5, DP_6, CP_1$	42
4	Single pedestrian crosses the road, behaving unexpectedly.	$DP_1 = 1, DP_3 = 2.7m/s, DP_4, DP_6, CP_1$	12
5	Running to cross, then walking in parallel on sidewalk, and suddenly running to cross	$DP_1 = 1, DP_3 = \{2.7, 3.5\}m/s, DP_4, DP_6, CP_1$	12
6	Running to cross, waiting, and suddenly crossing the road	$DP_1 = 1, DP_3 = 2.7m/s, DP_4 = 25m, DP_6, CP_1$	6
7	Multiple pedestrians cross the road. One of them is behaving unexpectedly.	$DP_1 = 5, DP_3 = 1.5m/s, DP_4 = 25m, DP_6, CP_1$	6
8	Multiple pedestrians walk on sidewalk in parallel to ego and then cross the road.	$DP_1 = 4, DP_3 = \{1.5, 2.7\}m/s, DP_6, CP_1$	24
9	1 running pedestrian with multiple normal crossing pedestrians.	$DP_1 = 3, DP_3 = \{2.7, 3.5\}m/s, DP_4 = 25m, DP_6, CP_1$	24

Unless specified, the considered discrete parameters in Table 3.16 contain all values listed previously.

The generated test cases are initially evaluated using the current FSM decision-making module employed in WATonoBus’ daily operations. The operational boundaries are determined through the 3-sigma rule introduced in Section 3.2. Specifically, to identify these boundaries, edge test cases must be found among the generated test cases, particularly when continuous configuration parameters are involved. In this study, an test case is identified as edge case if it results in a collision between ego vehicle or their safety buffers. Figure 3.14 showcases the validation results for three pedestrian collision avoidance cases generated from atomic scenario 2.

The top row illustrates the s-distance in the Frenet frame between the pedestrian and the ego vehicle under test, with a blue dashed line indicating a rule violation in test scenario 3. The middle and bottom rows display the ego vehicle’s velocity profile and overall TTC metrics, respectively. The black line (test scenario 1) represents a typical scenario where the ego vehicle meets expected performance standards. The red and blue dashed lines depict scenarios with intersecting spatial-temporal trajectories (TTC close to 0) between the ego vehicle and the actor, each with distinct outcomes. In scenario 2, the ego vehicle stops before a collision due to the actor’s inability to react, mitigating the severity. In contrast, scenario 3 is a failure case where the ego vehicle does not slowdown in time, resulting in a collision.

Our method effectively addresses these situations by identifying long-tail cases. Additionally, the bottom plot in Figure 3.14 suggests that the WATonoBus’ effective braking TTC when encountering a sudden object is about one second. If the system detects an object within this timeframe, it likely leads to a failure, with the vehicle failing to decelerate in time and colliding with the actor, thus defining the ODD boundary.

After determining the operational boundaries and filtering out effective test cases based on these boundaries, the remaining test case numbers for each atomic scenario are listed in Table 3.16. These remaining test cases will be used to evaluate the performance of the proposed uncertainty-aware decision-making mechanism in Chapter 5.

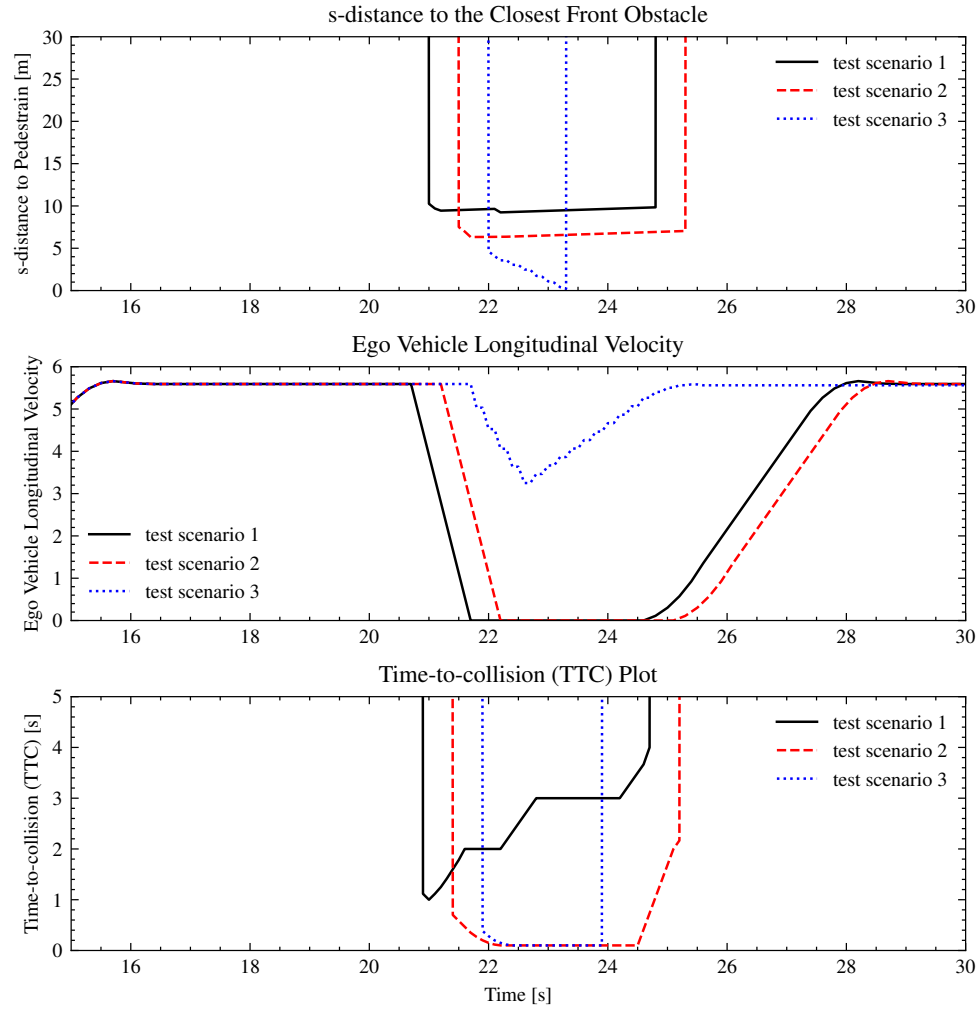


Figure 3.14: Example of Determining the Operation Boundary for WATonoBus FSM Decision-Making Module in Atomic Scenario 2

Chapter 4

Learning Algorithm Uncertainty Quantification

Drawing on the uncertainty-hazard relationships established in the preceding chapter, this section delves into the quantification of Object Position Uncertainty (OPU) originating from object detection algorithms. The decision to focus on OPU in the perception module stems from its significant impact on downstream modules. Nonetheless, the proposed methodology holds universal applicability across various types of uncertainty identified from the previous chapter. We employ Polynomial Chaos Expansion (PCE) for uncertainty quantification. Specifically, OPU is quantified based on the detected pedestrian position and the detection errors of reference objects. OPU represents the uncertainty or bias associated with the detected pedestrian’s position. This uncertainty can be interpreted as a radius, representing the range within which the true pedestrian position is likely to lie, centered around the detected location. By quantifying OPU, we aim to capture the error margins in pedestrian position detection and improve the robustness of autonomous systems in uncertain environments. Section 4.1 elucidating the background and principles of PCE to set the stage for later analysis. Subsequently, Section 4.2 provides a comprehensive problem statement regarding OPU quantification, establishing the groundwork for the subsequent application of PCE. Theoretical specifics on implementing OPU quantification using PCE are outlined in Section 4.3, while Section 4.4 showcases simulation and real-world experiment results, and performance comparison with a baseline model. This is followed by a thorough discussion in Section 4.5, providing insights into the effectiveness and reliability of the proposed approach.

4.1 Polynomial Chaos Expansion (PCE) Fundamentals

Polynomial Chaos Expansion (PCE) stands as a powerful metamodeling tool for quantifying uncertainty arising in complex systems. It provides a functional approximation of a computational model through its spectral representation on a suitably built basis of polynomial functions. These polynomial basis functions are chosen to ensure that the expansion coefficients, namely polynomial chaos coefficients, can be efficiently computed using statistical moments of the random variables involved. By limiting the expansion to a finite order, PCE can adeptly approximate joint, high-dimensional probability distributions with a finite number of terms, presenting a more efficient approach to uncertainty quantification compared to Monte Carlo methods, particularly beneficial for high-dimensional problems [139]. Another key advantage of PCE is its ability to handle non-Gaussian and non-linear uncertainties encountered in practical applications. This strength enables PCE to accommodate arbitrary probability distributions and complex system behaviors, making it well-suited for modeling uncertainties in dynamical systems, stochastic process, and high-dimensional parameters spaces [140]. Compared to the uncertainty quantification techniques reviewed in Section 2.2.2, PCE emerges as a promising choice for our second research objective given its efficiency and adaptability to various types of learning algorithms and uncertainties. Consequently, PCE is selected for exploration in this thesis and its technical fundamentals will be introduced in the subsequent sub-sections.

4.1.1 PCE Definition and Polynomial Basis

Polynomial Chaos Expansion (PCE) was first introduced by Norbert Wiener in 1938 [141]. It is used for representing random processes using orthogonal polynomials, initially being applied to model Gaussian noise and generalized to model non-Gaussian distributions in 1990s [142]. Given a random vector $\mathbf{X} \in \mathbb{R}$ that follows a joint Probability Density Function (PDF) $f_{\mathbf{X}}$, a model M that maps $\mathbf{X}$ to Y like $Y = M(\mathbf{X})$, and with variance of Y across the valid domain of $\mathbf{X}$, $D_{\mathbf{X}}$:

$$\mathbb{E}[Y^2] = \int_{D_{\mathbf{X}}} M^2(\mathbf{x}) f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} < \infty \quad (4.1)$$

Then the polynomial chaos expansion of $M(\mathbf{X})$ is defined as:

$$Y = M(\mathbf{X}) = \sum_{\alpha \in \mathbb{N}} c_{\alpha} \Psi_{\alpha}(\mathbf{X}) \quad (4.2)$$

where $\Psi_\alpha(\mathbf{X})$ are multivariate polynomial basis functions orthogonal under the distribution $f_{\mathbf{X}}$. The max number of variables in $\Psi_\alpha(\mathbf{X})$ is equal to the dimension of $\mathbf{X}$. Orthogonality of the polynomial basis functions is crucial in PCE because it ensures an efficient, accurate, and independent representation of the uncertainty in a system. It simplifies the computation of coefficients, minimizes approximation errors, and aligns the basis functions with the properties of the modeled distribution. $\alpha \in \mathbb{N}$ is used to index each polynomial basis function $\Psi_\alpha(\mathbf{X})$, and $c_\alpha \in \mathbb{R}$ are the polynomial chaos coefficients corresponding to each $\Psi_\alpha(\mathbf{X})$. Practically, the summation in Equation (4.2) needs to be truncated to a finite number of terms and it can be rewritten as:

$$Y = M(\mathbf{X}) \approx \sum_{\alpha \in A} c_\alpha \Psi_\alpha(\mathbf{X}) \quad (4.3)$$

where $A \subset \mathbb{N}$ is a set of selected indices of polynomial basis function $\Psi_\alpha(\mathbf{X})$. The mean and variance of a modeled distribution using PCE can be calculated by the following equations

$$\begin{aligned} \mu &= c_0 \\ \text{var}(f) &= \sum_{i=1}^P c_i^2 \end{aligned} \quad (4.4)$$

where c_0 is the PCE coefficient associated with the zeroth-order term. c_i are the PCE coefficients for the higher-order terms and P is the total number of polynomial basis function terms.

The polynomial basis functions $\Psi_\alpha(\mathbf{X})$ are defined based on orthonormal polynomial definition. A univariate orthonormal polynomial $\phi_q^{(i)}(x_i)$ with respect to another polynomial $\phi_p^{(i)}(x_i)$ of the i^{th} input, x_i , can be defined as below:

$$\langle \phi_p^{(i)}(x_i), \phi_q^{(i)}(x_i) \rangle \stackrel{\text{def}}{=} \int_{D_{\mathbf{X}}} \phi_p^{(i)}(x_i) \phi_q^{(i)}(x_i) f_{X_i}(x_i) dx = \delta_{pq} \quad (4.5)$$

where x_i identifies the i^{th} input variable, p and q are the corresponding polynomial degrees, $f_{X_i}(x_i)$ is the i^{th} input distribution and δ_{pq} is the Kronecker symbol. Based on Equation (4.5), the multivariate polynomial basis function $\Psi_\alpha(\mathbf{x})$ can be treated as the product of their univariate counterparts in Equation (4.6) and the orthogonality can be demonstrated as Equation (4.7)

$$\Psi_\alpha(\mathbf{x}) \stackrel{\text{def}}{=} \prod_{i=1}^K \phi_{\alpha_i}^{(i)}(x_i) \quad (4.6)$$

$$\langle \Psi_\alpha(\mathbf{x}), \Psi_\beta(\mathbf{x}) \rangle = \delta_{\alpha\beta} \quad (4.7)$$

where $\delta_{\alpha\beta}$ is the Kronecker symbol for high-dimensional case and K represents the considered max polynomial degree. For example, if the model M depends on two variables x_1 and x_2 , the univariate, first degree polynomial basis functions will be $\Psi_1^1(x_1)$ and $\Psi_1^2(x_2)$. Then the multivariate polynomial basis function based on the previously two univariate polynomial basis function will be $\Psi_1^1(x_1)\Psi_1^2(x_2)$.

Based on the definition of polynomial basis functions, intensive studies have been performed to determine the orthonormal polynomials to different distribution. A few common univariate orthonormal polynomials and their distributions are tabulated in the Table 4.1 below as reference [143]. k is the polynomial degree in this table, a and b are the parameters from the corresponding distributions. If the input random variable distribution is unknown, their polynomial basis functions can be numerically determined based on the available dataset of input random variables.

4.1.2 PCE Polynomial Coefficient Calculation Methods

After identifying the polynomial basis function for the desired distributions in Equation (4.3), the next step is to calculate the polynomial chaos coefficients, c_α . The two principal coefficient calculation strategies are projection and regression. Projection methods compute the coefficients based on the orthogonality of basis functions through numerical integration. Another more commonly used method, regression, formulates Equation (4.2) as a set of linear equations and solves it by linear regression approaches. This sub-section will illustrate two coefficient calculation methods: ordinary least-square regression, and sparse regression, which are employed in this thesis. For the details of all PCE coefficients calculation methods, readers can refer to [144].

Table 4.1: Common Families of Univariate Orthonormal Polynomials

Distribution	Domain	Orthogonal Polynomial Family	Polynomial Basis Function
Uniform	$[-1, 1]$	Legendre $P_k(x)$	$P_k(x)/\sqrt{\frac{1}{2k+1}}$
Gaussian	$(-\infty, \infty)$	Hermite $H_{e_k}(x)$	$H_{e_k}(x)/\sqrt{k!}$
Gamma	$(0, \infty)$	Laguerre $L_k^a(x)$	$L_k^a(x)/\sqrt{\frac{\Gamma(k+a+1)}{k!}}$
Beta	$[-1, 1]$	Jacobi $J_k^{a,b}(x)$	$J_k^{a,b}(x)/\left(\frac{2^{a+b+1}}{2k+a+b+1} \frac{\Gamma(k+a+1)\Gamma(k+b+1)}{\Gamma(k+a+b+1)\Gamma(k+1)}\right)$

Ordinary Least-square Regression

To implement a least-square minimization problem, the infinite series in Equation (4.2) can be rewritten as the sum of Equation (4.3) and a residual [145]:

$$Y = M(\mathbf{X}) = \sum_{j=0}^{P-1} c_j \Psi_j(\mathbf{X}) + \varepsilon_P \equiv \mathbf{c}^T \Psi(\mathbf{X}) + \varepsilon_P \quad (4.8)$$

$$P = \binom{N+p}{p} \quad (4.9)$$

where N is the number of input variables whose total degree is less than or equal to p in all polynomials, ε_P is the truncation residual, $\mathbf{c}_\alpha = \{c_0, \dots, c_{P-1}\}^T$ is the polynomial coefficient vector, $\Psi(\mathbf{X}) = \{\Psi_0(\mathbf{X}), \dots, \Psi_{P-1}(\mathbf{X})\}^T$ is the orthonormal polynomial vector for input $\mathbf{X}$. Then the least-square minimization problem can be formulated as:

$$\hat{\mathbf{c}} = \arg \min E \left[(\mathbf{c}^T \Psi(\mathbf{X}) - M(\mathbf{X}))^2 \right] \quad (4.10)$$

The least-square minimization problem in Equation (4.10) can be approached by ordinary least-square (OLS) solution, which considers a given sample $\mathcal{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\}^T$ with n samples and the corresponding responses $\mathcal{Y} = \{y^{(1)}, \dots, y^{(n)}\}^T$, the OLS solution to Equation (4.10) can be written as below:

$$\hat{\mathbf{c}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathcal{Y} \quad (4.11)$$

$$A_{ij} = \Psi_j(\mathbf{x}^{(i)}), \text{ where } i = 1, \dots, n, j = 0, \dots, P-1 \quad (4.12)$$

where A_{ij} is the regression matrix and contains the values of all basis polynomials given sample $\mathcal{X}$. Compared to projection method, the strength of least-square minimization problem is that an arbitrary number of sample points can be used to determine the polynomial coefficients if they are non-trivial and $N \geq P$. The theoretical proof of least-square minimization problem convergence can be found in [146].

Least Angle Regression (LAR) for Sparse PCE

When constructing high-dimensional PCE to quantify uncertainty, sparsity arises from the orthogonal structure of the selected polynomial basis functions and the limited contribution of many polynomial terms to the representation of uncertainty. To handle the sparsity

in high dimension while effectively calculating PCE coefficients, a regularization term, $\lambda\|\mathbf{y}\|_1$, can be added to the least-square minimization problem in Equation (4.10) and the optimization problem is changed to:

$$\hat{\mathbf{c}} = \arg \min_{\mathbf{c} \in \mathbb{R}^{|A|}} E \left[(\mathbf{c}^T \Psi(\mathbf{X}) - M(\mathbf{X}))^2 \right] + \lambda \|\mathbf{c}\|_1 \quad (4.13)$$

$$\|\mathbf{c}\|_1 = \sum_{\alpha \in A} |c_\alpha| \quad (4.14)$$

where $A \subset \mathbb{N}^M$ is a set of selected indices of polynomial basis function $\Psi_\alpha(\mathbf{X})$. The regularization term forces the optimization towards low-rank solutions and therefore handles the sparsity issue with high-dimensional PCE. There are several existing algorithms that solve Equation (4.13) [147, 148, 149, 150]. The employed algorithm in this thesis, least angle regression (LAR) algorithm [149], will be introduced next.

LAR is a regression algorithm used for dimensionality reduction, particularly in high-dimensional datasets where the number of regressors is much larger than that of observations. Considering Equation (4.13), it means that the number of terms in each polynomial basis function is much larger than the number of polynomial chaos coefficients, y_α . The key idea behind LAR is to gradually add regressors from candidate set to active set while simultaneously shrinking their coefficients towards the least-square estimates. At each step, LAR identifies the regressor most correlated with the response variable and moves in the direction of this regressor until another regressor becomes equally correlated. This process continues until all regressors are included in the model. The optimal number of regressors in the model may be determined using a suitable criterion. The LAR algorithm is shown in Algorithm 1.

When handling with high-dimensional PCE, LAR iteration can be inefficient usually due to error calculations. An early stop criterion can be applied to speed-up the algorithm. The early stop criterion is based on the observation that most error estimate changing tendency is relatively smooth with respect to the increasing iteration number. Hence, an effective LAR early stop criterion is to stop adding regressors after the error estimate is above its minimum for at least 10% of the maximum number of allowed iterations.

4.1.3 Bootstrap PCE (bPCE) and Polynomial Chaos Kriging (PCK)

Other PCE-related techniques are also investigated and compared to make the final selection on the uncertainty quantification method. Two other techniques, namely bootstrap

Algorithm 1: LAR algorithm used in PCE coefficient calculation

Initialize: Initialize polynomial chaos coefficient: $c_\alpha = 0$, candidate regressor set: Ψ_α , active set: $\emptyset$, residual: $r_0 = \mathcal{Y}$.

1 Step 1: Iteration

- 1-1 Step 1-1: Find the most correlated regressor Ψ_{α_j} to the current residual.
- 1-2 Step 1-2: Move all coefficients in the current active set towards their least-square values until their predictors are equally correlated with the residual as another predictor in the candidate set, which will also be the most correlated with the residual in the next iteration.
- 1-3 Step 1-3: Calculate and store the error estimate for the current iteration.
- 1-4 Step 1-4: Update active coefficients, c_α .
- 1-5 Step 1-5: Move Ψ_{α_j} from candidate set to active set.
- 1-6 Step 1-6: Repeat step 1-1 to 1-5 until the active set size is equal to $\min(P, N - 1)$.

2 Step 2: Regressor Selection

- 2-1 Step 2-1: The remaining regressor with the lowest error estimates will be selected as the best sparse candidate basis.
-

PCE (bPCE) [151] and Polynomial Chaos Kriging (PCK) [152] are studied and implemented within simulation environment along with PCE to understand their strengths and weaknesses at the initial stage, facilitating us to choose the most suitable technique to be implemented in OPU quantification task.

Specifically, bPCE integrates bootstrap resampling idea with PCE to generate n_B replications for initialization dataset $\mathcal{X}$ based on sampling with substitution principle, like $\{\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(n_B)}\}$. These replicated datasets will be used to initialize n_B PCE models and eventually will generate n_B predictions on a point x . bPCE offers the capability to calculate the variance of predictions or other interested statistical results.

Universal Kriging is a stochastic algorithm that assumes the model $Y = M(\mathbf{X})$ is a Gaussian process indexed by $\mathbf{X}$. The equation of universal Kriging model, $M^K(\mathbf{X})$, is formulated as below [153]:

$$Y \approx M^K(\mathbf{X}) = \beta^T f(\mathbf{X}) + \sigma^2 Z(\mathbf{X}, w) \quad (4.15)$$

where $\beta^T f(\mathbf{X})$ is the mean value model response that follows the assumed Gaussian process, σ^2 is the Gaussian process variance, and $Z(\mathbf{X}, w)$ is a zero mean, unit variance Gaussian process. Together with Equation (4.3), the PCK equation is written as below:

$$Y = M^{PCK}(\mathbf{X}) = \sum_{\alpha \in A} c_\alpha \Psi_\alpha(\mathbf{X}) + \sigma^2 Z(\mathbf{X}, w) \quad (4.16)$$

where the first half of Equation (4.16) is the sum of orthonormal polynomials, approximating the global trend of Y , and the second half is the Kriging model that interpolates the local variance of Y as a function of neighboring data within $\mathbf{X}$.

4.1.4 Error Estimation and PCE Evaluation

After the surrogate models like PCE, bPCE and PCK are constructed, their prediction accuracies need to be evaluated across the same validation dataset. The prediction error, ε , can be calculated based on the equation below:

$$\varepsilon = \frac{E \left[(M(\mathbf{X}) - M^{PCE}(\mathbf{X}))^2 \right]}{Var[Y]} \quad (4.17)$$

where $M^{PCE}(\mathbf{X})$ is the constructed PCE model. More specifically, normalized empirical error (ε_{emp}) and Leave-One-Out (LOO) cross-validation error (ε_{LOO}) can then be used to evaluate PCE predictions in this case [154]. Using PCE model as an example, normalized empirical error is calculated by replacing the sample mean of validation set response by the PCE model responses shown as below:

$$\varepsilon_{emp} = \frac{\sum_{i=1}^N (M(\mathbf{x}^{(i)}) - M^{PCE}(\mathbf{x}^{(i)}))^2}{\sum_{i=1}^N \left(M(\mathbf{x}^{(i)}) - \frac{1}{N} \sum_{i=1}^N M^{PCE}(\mathbf{x}^{(i)}) \right)^2} \quad (4.18)$$

Nevertheless, Equation (4.18) tends to over-fit the PCE prediction accuracy because it is monotonically decreasing with respect to the polynomial degree p . To overcome this limitation, LOO error is implemented by using cross-validation. The equation can be written as:

$$\varepsilon_{LOO} = \frac{\sum_{i=1}^N (M(\mathbf{x}^{(i)}) - M^{PCE \setminus i}(\mathbf{x}^{(i)}))^2}{\sum_{i=1}^N \left(M(\mathbf{x}^{(i)}) - \frac{1}{N} \sum_{i=1}^N M^{PCE \setminus i}(\mathbf{x}^{(i)}) \right)^2} \quad (4.19)$$

where N PCE models ($M^{PCE \setminus i}$) are built on a reduced dataset $\{\mathbf{x}^{(j)}, j = 1, \dots, N, j \neq i\}$, excluding point $(\mathbf{x}^{(i)}, y^{(i)})$.

Other than comparing different surrogate models' accuracies across one dataset, to determine the accuracy of individual regression model, Mean Absolute Error (MAE) and

Root Mean Square Error (RMSE) are the two commonly employed evaluation criteria with the following equations:

$$MAE = \frac{1}{N} \sum_{i=1}^N |M(\mathbf{x}^{(i)}) - M^{PCE}(\mathbf{x}^{(i)})| \quad (4.20)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (M(\mathbf{x}^{(i)}) - M^{PCE}(\mathbf{x}^{(i)}))^2} \quad (4.21)$$

M^{PCE} in Equation (4.18-4.21) can be replaced by M^{bPCE} and M^{PCK} to check the accuracies of other surrogate models.

4.2 Problem Statement of Quantifying Object Position Uncertainty (OPU)

This section aims to employ the PCE method introduced earlier to quantify the identified OPU discussed in Chapter 3. While various uncertainties and their links to potential vehicle-level hazards were delineated in the previous chapter, precise object positioning is pivotal for several downstream functional modules, including object tracking, trajectory prediction, collision avoidance, path planning, and intersection handling. Moreover, pedestrian detection poses challenges due to factors such as the variability in pedestrian appearance, susceptibility to occlusion, dynamic behavior, and movement against complex backgrounds [155, 156]. Additionally, whether in urban or highway settings, ADS commonly relies on HD map data to enhance reliability across perception [157, 158], localization [159, 160], and decision-making tasks [161, 162]. Specifically, the static landmarks provided by HD maps, such as traffic signs, building outlines, lane boundaries, and light poles, can significantly augment perception, especially in adverse weather and lighting conditions [163]. Therefore, this thesis will specifically investigate pedestrian position uncertainty using the PCE uncertainty quantification method, leveraging HD map data to enhance parameter updating and address out-of-distribution scenarios in real-time operations. While PCE is utilized here to quantify OPU, the proposed methodology is versatile enough to be applied to other identified uncertainties from Chapter 3.

In this study, the distance ranges are represented by the longitudinal and lateral distances (x, y) in ego vehicle's frame. For the field of view of perception module, its detection range can be divided into different zones considering different distance-based safety fallback

reactions (See Figure 4.1). There are two types of objects in this uncertainty quantification work, namely static reference objects and pedestrians. Their true positions can be represented as (x^{ref}, y^{ref}) and (x^{ped}, y^{ped}) . In addition, their perceived positions are $(\hat{x}^{ref}, \hat{y}^{ref})$ and $(\hat{x}^{ped}, \hat{y}^{ped})$. Similarly, the OPU, which can be expressed as the true and estimated position errors, can be expressed as (e_x, e_y) and $(\hat{e}_x, \hat{e}_y)$. Based on that, the L2 norm OPU can also be calculated. Assuming the position uncertainties arising from pedestrian and reference object detections are similar, the newly detected reference object positions can be used to compared with their true positions from HD map over time to evaluate PCE uncertainty quantification performance and re-calibrate the PCE model during real-time operation. The detected pedestrian positions over a period, $[0, t_{th}]$, can be expressed as:

$$(\hat{\mathbf{x}}^{ped}, \hat{\mathbf{y}}^{ped}) = \left\{ \left(\hat{x}_0^{ped}, \hat{y}_0^{ped} \right), \dots, \left(\hat{x}_{t_{th}}^{ped}, \hat{y}_{t_{th}}^{ped} \right) \right\} \quad (4.22)$$

The newly collected reference object position errors are:

$$(\hat{\mathbf{e}}_x^{ref}, \hat{\mathbf{e}}_y^{ref}) = \left\{ \left(\hat{e}_{x,0}^{ref}, \hat{e}_{y,0}^{ref} \right), \dots, \left(\hat{e}_{x,t_{th}}^{ref}, \hat{e}_{y,t_{th}}^{ref} \right) \right\} \quad (4.23)$$

Eventually, the OPU quantification work requires a function that can map both the given perceived pedestrian positions and the detection errors on reference objects over time and within certain distance ranges to explicit OPU like Equation (4.24) or L2 norm OPU if needed like Equation (4.25):

$$(\hat{\mathbf{e}}_x^{ped}, \hat{\mathbf{e}}_y^{ped}) = f(\hat{\mathbf{x}}^{ped}, \hat{\mathbf{y}}^{ped}, \hat{\mathbf{e}}_x^{ref}, \hat{\mathbf{e}}_y^{ref}) \quad (4.24)$$

$$\hat{\mathbf{e}}^{ped} = f(\hat{\mathbf{x}}^{ped}, \hat{\mathbf{y}}^{ped}, \hat{\mathbf{e}}^{ref}) \quad (4.25)$$

Based on the data from on-vehicle operation recordings, Figure 4.2 is drafted to help better visualize how PCE can solve the aforementioned problem. Eventually, a trained PCE model will formulate a joint distribution about the distributions of perceived objects' longitudinal and lateral positions and the corresponding object position error. Occasionally, the model will update itself with the newly collected estimation errors based on checking with reference object positions in HD map.

4.3 OPU Quantification with PCE

4.3.1 Initialization Stage

In this section, based on the previous problem statement about OPU quantification, the PCE theoretical implementation details will be presented. Firstly, the distributions of two

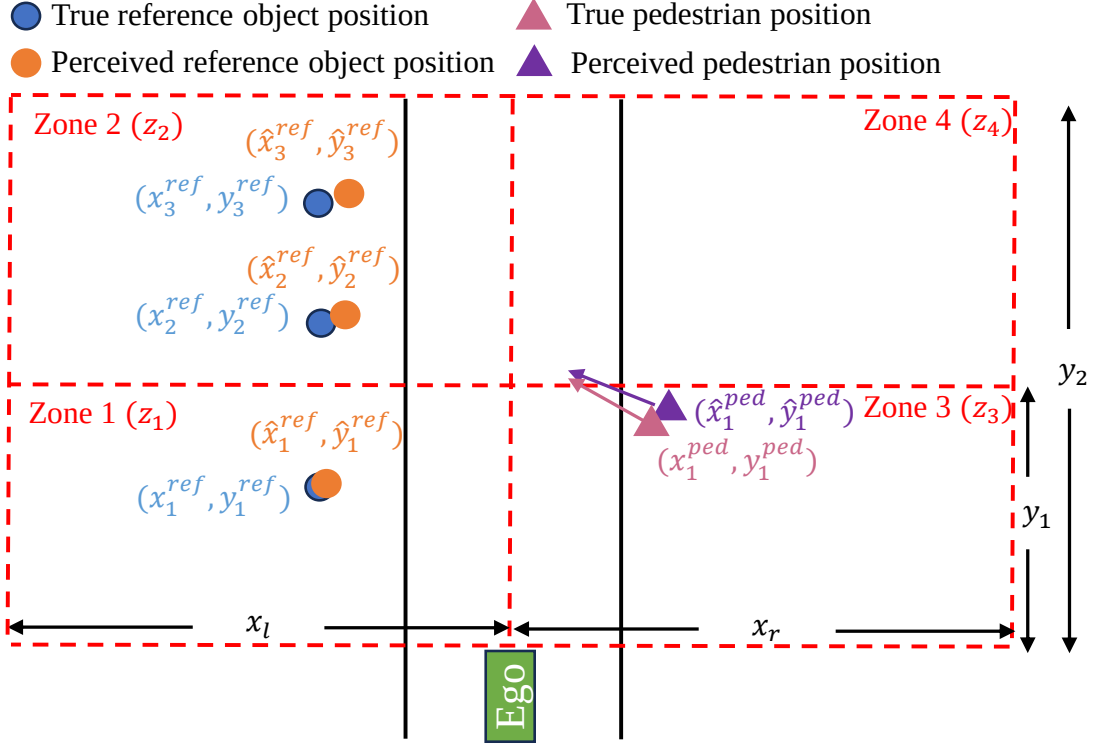


Figure 4.1: Schematic Diagram of OPU Quantification

input variables $(\hat{\mathbf{x}}^{ped}, \hat{\mathbf{y}}^{ped})$ are assumed to be both Gaussian distributions based on the observation of the collected data. Furthermore, pedestrians also exhibit similar behavior logic when a vehicle is approaching, resulting in Gaussian-like distributions (see Figure 4.3) for their longitudinal and lateral distances with respect to the ego vehicle [4, 164, 165].

Therefore, based on Table 4.1, the polynomial basis function will be Hermite polynomial, and its univariate equation is shown as below:

$$\psi_k(x) = \frac{H_k(x)}{\sqrt{k!2^k\sqrt{\pi}}} = \frac{(-1)^k e^{x^2} \frac{d^k}{dx^k} e^{-x^2}}{\sqrt{k!2^k\sqrt{\pi}}} \quad (4.26)$$

where k is polynomial degree, $H_k(x)$ is Hermite polynomial. Furthermore, there are two input variables, $\hat{\mathbf{x}}^{ped}$ and $\hat{\mathbf{y}}^{ped}$, in this case. Therefore, based on Equation (4.6), the multivariate polynomial basis function is formulated as below:

$$\psi_{i,j}(\hat{\mathbf{x}}^{ped}, \hat{\mathbf{y}}^{ped}) = \psi_i(\hat{\mathbf{x}}^{ped}) \cdot \psi_j(\hat{\mathbf{y}}^{ped}) = \frac{H_i(\hat{\mathbf{x}}^{ped})}{\sqrt{i!2^i\sqrt{\pi}}} \cdot \frac{H_j(\hat{\mathbf{y}}^{ped})}{\sqrt{j!2^j\sqrt{\pi}}} = \frac{H_i(\hat{\mathbf{x}}^{ped})H_j(\hat{\mathbf{y}}^{ped})}{\sqrt{(i!j!)2^{i+j}\pi}} \quad (4.27)$$

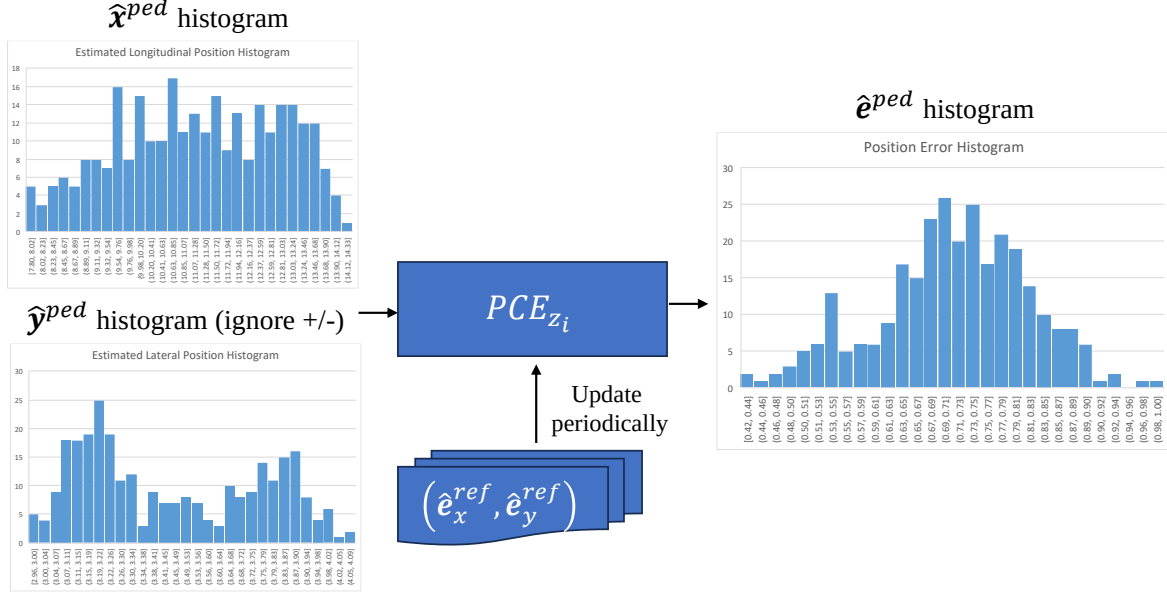


Figure 4.2: Visualization of OPU Quantification with PCE

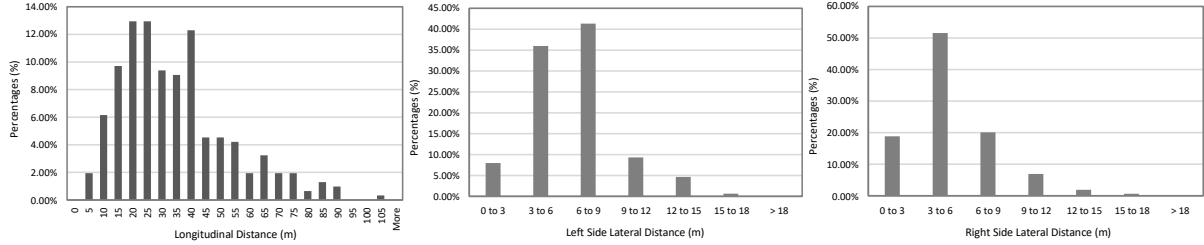


Figure 4.3: Statistics of Pedestrian Occurrence from Vehicle Perspective [4]

where i and j are polynomial degrees for the two polynomial basis functions. According to Equation (4.3), the PCE model for OPU considering multiple variables can then be written as:

$$\hat{\mathbf{e}}^{ped} = M^{PCE}(\hat{\mathbf{x}}^{ped}, \hat{\mathbf{y}}^{ped}) = \sum_{i=0}^{K_{long}} \sum_{j=0}^{K_{lat}} c_{i,j} \psi_{i,j}(\hat{\mathbf{x}}^{ped}, \hat{\mathbf{y}}^{ped}) \quad (4.28)$$

where M^{PCE} represents the PCE model considering the two input variables, K_{long} and K_{lat} are the max polynomial degrees for the two polynomial basis functions ψ_i and ψ_j , $c_{i,j}$ is the polynomial chaos coefficient for its associated polynomial basis functions $\psi_{i,j}$.

However, in cases where the Gaussian assumption does not hold, such as in our study, the output variable ($\hat{\mathbf{e}}^{ped}$) in the collected data exhibits diverse distribution patterns across different zones (See Figure 4.4). To address this issue, Box-Cox transformation can be used to transform the collected data to a normal distribution. Box-Cox transformation is a statistical technique used to stabilize variance and make data more closely approximate a normal distribution [166]. The working principle of Box-Cox transformation involves applying a power transformation to the data, which is controlled by a parameter, typically denoted as λ . The transformation is defined by the following equation:

$$u^{(\lambda)} = \begin{cases} \log(u) & \text{if } \lambda = 0 \\ \frac{u^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \end{cases} \quad (4.29)$$

where u is the original data and $u^{(\lambda)}$ is the transformed data. The transformation is governed by a parameter λ . If $0 < \lambda < 1$, the transformation is useful for addressing positively skewed data, where most data points are concentrated towards lower values. Conversely, if $\lambda > 1$, the transformation increases data variability, making it less homoscedastic. This is beneficial when the original data is tightly clustered around the mean with limited variability. For negative skew, $\lambda < 0$ helps alleviating the skewness. In case of $\lambda = 0$, a natural logarithm is applied to the data to stabilize its variance, particularly when the data shows exponential growth or decay. Finally, if $\lambda = 1$, it means the data is already normally distributed, rendering the Box-Cox transformation unnecessary.

To determine the optimal λ , the log-likelihood function (llf) for Box-Cox transformation is firstly defined as below:

$$llf = (\lambda - 1) \sum_{i=1}^n (\log(u_i)) - n/2 \log\left(\sum_{i=1}^n (u_i^{(\lambda)} - \bar{u}^{(\lambda)})^2 / n\right) \quad (4.30)$$

where n is the number of available sample points and $\bar{u}^{(\lambda)}$ is the mean of transformed sample points. Then the optimal λ that maximizes Equation (4.30) can be determined by solving Equation (4.31). Nelder-Mead simplex algorithm is employed as the optimization techniques in this study [167].

$$\lambda^* = \arg \max_{\lambda} (llf) = \arg \max_{\lambda} \left((\lambda - 1) \sum_{i=1}^n (\log(u_i)) - n/2 \log\left(\sum_{i=1}^n (u_i^{(\lambda)} - \bar{u}^{(\lambda)})^2 / n\right) \right) \quad (4.31)$$

After M^{PCE} is formulated in Equation (4.27), the next step will be determining the polynomial chaos coefficients, $c_{i,j}$, with proper expansion order (K_{long} , K_{lat}) by fitting the PCE

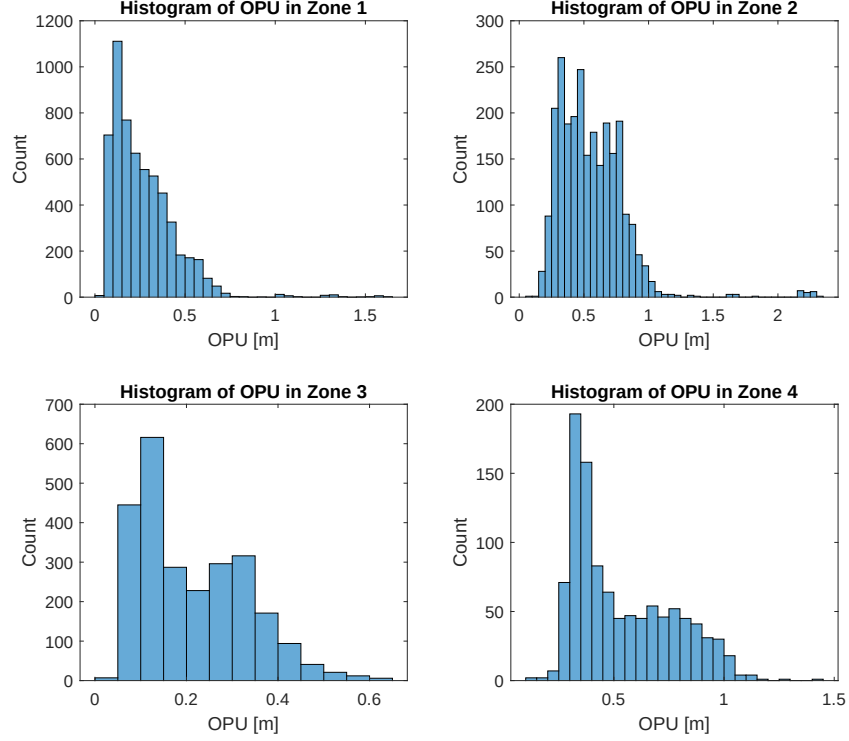


Figure 4.4: OPU Distributions in Different Zones from Real-world Data

model to an initialization dataset. The split ratio between initialization and validation datasets is set at 80% and 20% in this study. Afterwards, the PCE model needs to be validated to ensure that it can accurately capture the behavior of the system under uncertainty. The initialization and validation process for the PCE model is shown in Algorithm 2. Note that for line 1-5 in Algorithm 2, LARS is from Algorithm 1, it can be replaced by other polynomial chaos coefficient calculation method like OLS in Section 4.1.2.

4.3.2 Testing Stage

After initializing and validating the PCE model in the preceding section, it becomes deployable for real-time operations, facilitating the prediction of OPU. Moreover, as the volume of newly acquired samples from reference objects continues to grow, they serve to update the initialized PCE model, enabling dynamic adaptation to external environmental changes such as variations in weather or light conditions. This online model updating workflow is designed in this research and is shown as two parts as Algorithms 3 and 4. Specifically,

Algorithm 2: PCE model initialization and validation algorithm

Input : Number of detection zones (m), detection zones ($\mathbf{z} = \{z_1, \dots, z_m\}$), initialization datasets ($\mathcal{D}^{init} = \{D_{z_1}^{init}, \dots, D_{z_m}^{init}\}$), validation datasets ($\mathcal{D}^{val} = \{D_{z_1}^{val}, \dots, D_{z_m}^{val}\}$), max polynomial basis function degrees of input variables (K_{long} and K_{lat}).

Output : A list of PCE models ($\mathcal{M}^{PCE} = \{M_{z_1}^{PCE}, \dots, M_{z_m}^{PCE}\}$), a list of lambda parameters ($\mathcal{P}_\lambda = \{\lambda_{z_1}, \dots, \lambda_{z_m}\}$), a list of normalized empirical error ($\mathcal{E}_{emp} = \{\varepsilon_{emp, z_1}, \dots, \varepsilon_{emp, z_m}\}$), a list of Leave-one-out (LOO) cross-validation error ($\mathcal{E}_{LOO} = \{\varepsilon_{LOO, z_1}, \dots, \varepsilon_{LOO, z_m}\}$).

Initialize: Initialize PCE model list: $\mathcal{M}^{PCE} = \{\}$, lambda parameter list: $\mathcal{P}_\lambda = \{\}$, normalized empirical error list: $\mathcal{E}_{emp} = \{\}$, Leave-one-out (LOO) cross-validation list: $\mathcal{E}_{LOO} = \{\}$.

1-1 **for** $i = 1$ **to** m **do**

1-2 Find initialization dataset $D_{z_i}^{init}$ and validation dataset $D_{z_i}^{val}$ based on detection zone z_i .

1-3 Obtain $\mathbf{x}^{init}, \mathbf{y}^{init}, \mathbf{e}^{init}$ from $D_{z_i}^{init}$, $\mathbf{x}^{val}, \mathbf{y}^{val}, \mathbf{e}^{val}$ from $D_{z_i}^{val}$.

1-4 Transform $\hat{\mathbf{e}}$ into normal distribution: $[\mathbf{e}^{(\lambda), init}, \mathbf{e}^{(\lambda), val}, \lambda_{z_i}] = \text{boxcox}(\mathbf{e}^{init}, \mathbf{e}^{val})$

1-5 Calculate polynomial chaos coefficients $\mathbf{c}$ for $M_{z_i}^{PCE}$:
 $[\mathbf{c}, M_{z_i}^{PCE}] = \text{LARS}(\mathbf{x}^{init}, \mathbf{y}^{init}, \mathbf{e}^{init}, K_{long}, K_{lat})$

1-6 Save $M_{z_i}^{PCE}$ in PCE model list $\mathcal{M}^{PCE}$, and λ_{z_i} in $\mathcal{P}_\lambda$.

1-7 $M_{z_i}^{PCE}$ validation: $\hat{\mathbf{e}} = M_{z_i}^{PCE}(\hat{\mathbf{x}}^{val}, \hat{\mathbf{y}}^{val})$

1-8 Determine validation error ε_{emp} and ε_{LOO} with $\hat{\mathbf{e}}$ and $\mathbf{e}^{val}$ as explained in Section 4.1.4.

1-9 Save ε_{emp} and ε_{LOO} to $\mathcal{E}_{emp}$ and $\mathcal{E}_{LOO}$.

1-10 **end for**

Algorithm 3 shows the necessary settings and initialization of this workflow. Algorithm 4 shows the main body of the algorithm. During online operations, at each time step, every object identified by the perception algorithm undergoes categorization into distinct detection zones and classification as either a pedestrian or reference object. For pedestrian objects, their positional uncertainty is quantified via the relevant PCE model tailored to their respective detection zone, with the quantified uncertainty findings being stored for subsequent accuracy assessments. In the case of reference objects like stop signs or light poles recorded in the HD map, their position detection errors, $\hat{\mathbf{e}}^{ref}$, can be computed and supplemented into the new initialization dataset, $\mathcal{D}^{init, new}$, to facilitate the next update of the PCE model. Upon exceeding a predefined threshold in the sample count of the

Algorithm 3: PCE model testing algorithm (Part 1)

Input : Number of detected objects (n_{obj}), detected object information ($\hat{O} = \{\hat{O}_1, \dots, \hat{O}_{n_{obj}}\} = \{\{\hat{x}_1, \hat{y}_1, \hat{C}_1\}, \dots, \{\hat{x}_{n_{obj}}, \hat{y}_{n_{obj}}, \hat{C}_{n_{obj}}\}\}$), number of detection zones (m), detection zones ($\mathbf{z} = \{z_1, \dots, z_m\}$), original PCE model list ($\mathcal{M}^{PCE,old} = \{M_{z_1}^{PCE,old}, \dots, M_{z_m}^{PCE,old}\}$), new initialization sample number threshold ($n_{new,init}$), original initialization datasets ($\mathcal{D}^{init} = \{D_{z_1}^{init}, \dots, D_{z_m}^{init}\}$), original validation datasets ($\mathcal{D}^{val} = \{D_{z_1}^{val}, \dots, D_{z_m}^{val}\}$), max polynomial basis function degrees of input variables (K_{long} and K_{lat}), percentage increment (Δp), discretized percentages for selecting optimal P^{OD} and P^{ND} ($\alpha^{OD,ND} = \{0, \Delta p, 2\Delta p, \dots, 1\}$), pre-defined weights for P^{OD} and P^{ND} ($\mathcal{W}^{OD,ND} = \{w^{OD}, w^{ND}\}$).

Output : Updated PCE model list ($\mathcal{M}^{PCE,new} = \{M_{z_1}^{PCE,new}, \dots, M_{z_m}^{PCE,new}\}$), temporary initialization datasets for newly collected samples ($\mathcal{D}^{init,temp} = \{D_{z_1}^{init,temp}, \dots, D_{z_m}^{init,temp}\}$), temporary validation datasets for newly collected samples ($\mathcal{D}^{val,temp} = \{D_{z_1}^{val,temp}, \dots, D_{z_m}^{val,temp}\}$), enriched initialization datasets ($\mathcal{D}^{init,new} = \{D_{z_1}^{init,new}, \dots, D_{z_m}^{init,new}\}$), original initialization data percentages for each detection zone ($\mathcal{P}^{OD} = \{P_{z_1}^{OD}, \dots, P_{z_m}^{OD}\}$), new initialization data percentages for each detection zone ($\mathcal{P}^{ND} = \{P_{z_1}^{ND}, \dots, P_{z_m}^{ND}\}$), number of detected pedestrians at each time step (n_{ped}), OPU prediction of detected pedestrians at each time step ($\hat{\mathbf{e}}^{ped} = \{\hat{e}_1^{ped}, \dots, \hat{e}_{n_{ped}}^{ped}\}$).

Initialize: $\mathcal{M}^{PCE,new} = \{M_{z_1}^{PCE,new}, \dots, M_{z_m}^{PCE,new}\} = \mathcal{M}^{PCE,old} = \{M_{z_1}^{PCE,old}, \dots, M_{z_m}^{PCE,old}\}$, $\mathcal{D}^{init,new} = \{D_{z_1}^{init,new}, \dots, D_{z_m}^{init,new}\} = \mathcal{D}^{init,old} = \{D_{z_1}^{init,old}, \dots, D_{z_m}^{init,old}\}$, $\mathcal{P}^{OD} = \{P_{z_1}^{OD}, \dots, P_{z_m}^{OD}\} = \{0.5, \dots, 0.5\}$, $\mathcal{P}^{ND} = \{P_{z_1}^{ND}, \dots, P_{z_m}^{ND}\} = \{0.5, \dots, 0.5\}$, $n_{ped} = 0$, $\hat{\mathbf{e}}^{ped} = \{\}$, number of newly collected initialization samples: $\mathcal{N}^{init,new} = \{n_{z_1}^{init,new}, \dots, n_{z_m}^{init,new}\} = \{0, \dots, 0\}$, thresholds of newly collected initialization sample number in each detection zone: $\mathcal{N}^{init,thres} = \{n_{z_1}^{init,thres}, \dots, n_{z_m}^{init,thres}\}$.

Note : Part 2 of PCE model testing algorithm is continued at Algorithm 4

new initialization dataset for a specific detection zone, $D_{z_i}^{init,new}$, the optimal percentages of incorporation from the new and old initialization datasets are determined, leading to

Algorithm 4: PCE model testing algorithm (Part 2)

```

1-1 for  $i = 1$  to  $n_{obj}$  do
1-2   Obtain detected object information,  $\hat{O}_i = \{\hat{x}_i, \hat{y}_i, \hat{C}_i\}$ .
1-3   Determine the detection zone,  $z_j (j \in \{1, \dots, m\})$ , where the detected object
       locates based on  $\hat{x}_i, \hat{y}_i$ 
1-4   Predict OPU of the detected object:  $\hat{e}_i = M_{z_j}^{PCE,old}(\hat{x}_i, \hat{y}_i)$ .
1-5   if  $C_i == \text{reference object}$  then
1-6     Save  $\hat{e}_i$  to  $D_{z_j}^{init,temp}$  in  $\mathcal{D}^{init,temp}$ 
1-7      $n_{z_j}^{init,new} = n_{z_j}^{init,new} + 1$ 
1-8     if  $n_{z_j}^{init,new} \geq n_{z_j}^{init,thres}$  then
1-9        $[P_{z_j}^{OD}, P_{z_j}^{ND}, M_{z_j}^{PCE,new}] =$ 
          $optiODND(\alpha^{OD,ND}, w^{OD,ND}, D_{z_j}^{init,temp}, D_{z_j}^{init,new}, D_{z_j}^{val,temp}, D_{z_j}^{val,old}, K_{long}, K_{lat})$ 
1-10      Add the newly collected initialization samples to original initialization
         dataset:  $D_{z_j}^{init,old} = D_{z_j}^{init,old} \cup D_{z_j}^{init,temp}$ .
1-11       $D_{z_j}^{init,new} = D_{z_j}^{init,old}$ 
1-12    end if
1-13  end if
1-14  if  $C_i == \text{pedestrian}$  then
1-15    Save  $\hat{e}_i$  as  $\hat{e}_i^{ped}$  to  $\hat{\mathbf{e}}^{ped}$ .
1-16     $n_{ped} = n_{ped} + 1$ 
1-17  end if
1-18 end for

```

the update of the PCE model for that detection zone, $M_{z_i}^{PCE}$. This iterative process is conducted for each detected object within the respective time step until all objects have been scrutinized.

Algorithm 3 and 4 delineate the comprehensive procedure for testing the PCE model concerning the newly detected objects from the perception algorithm. To accommodate the changing environment and the data drift challenges for the learning algorithms, continuous updating of the initialized PCE model is imperative during online operations, leveraging the influx of fresh samples obtained from the detection of reference objects. This model updating process is facilitated through the function $optiODND(\cdot)$ in lines 1-9 in Algorithm 4. Specifically, P^{OD} and P^{ND} , are determined based on two pivotal criteria: RMSE of validation samples, $RMSE_{\hat{\mathbf{e}}^{val}}$, and average prediction error of each subset of validation

samples, $APE_{\hat{\mathbf{e}}^{val}}$. These criteria are computed utilizing the following equations:

$$RMSE_{\hat{\mathbf{e}}^{val}} = \sqrt{\frac{1}{n_{\hat{\mathbf{e}}^{val}}} \sum_{i=1}^{n_{\hat{\mathbf{e}}^{val}}} (e_i^{val} - \hat{e}_i^{val})^2} \quad (4.32)$$

$$APE_{\hat{\mathbf{e}}^{val}} = \frac{1}{n_{sub}} \frac{1}{n_{\hat{\mathbf{e}}^{val},sub}} \sum_{i=1}^{n_{sub}} \sum_{j=1}^{n_{\hat{\mathbf{e}}^{val},sub}} |e_j^{val} - \hat{e}_j^{val}| \quad (4.33)$$

where $\hat{\mathbf{e}}^{val}$ is the validation errors, $n_{\hat{\mathbf{e}}^{val}}$ represents the number of validation samples, n_{sub} represents the number of subsets divided within the validation dataset, and $n_{\hat{\mathbf{e}}^{val},sub}$ is the sample number in each subset. Eventually, the optimal P^{OD} and P^{ND} can be solved through Equation (4.34) below:

$$[P^{OD}, P^{ND}] = \arg \min_{P^{OD}, P^{ND}} (w^{OD} RMSE_{\hat{\mathbf{e}}^{val}} + w^{ND} APE_{\hat{\mathbf{e}}^{val}}) \quad (4.34)$$

where w^{OD} and w^{ND} are the pre-defined weight assigned to $RMSE_{\hat{\mathbf{e}}^{val}}$ and $APE_{\hat{\mathbf{e}}^{val}}$ while performing the optimization. The detailed $optiODND(\cdot)$ algorithm is shown in Algorithm 5.

While RMSE provides an indication of the overall accuracy across the entire validation dataset, APE offers insights into periodic accuracy for specific subsets of validation data, revealing localized improvements post model updates (See Figure 4.5). The heat map with the RMSE result plot suggests that the optimal values for P^{OD} and P^{ND} appear to range from [20%, 50%] to [50%, 80%]. Upon examination of the APE in the middle plot, the

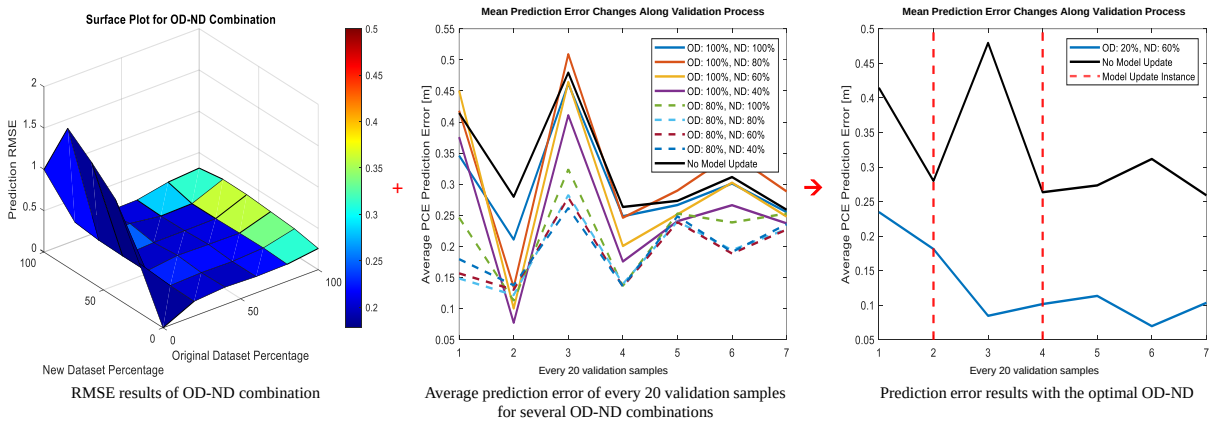


Figure 4.5: Visualization of Optimal P^{OD} and P^{ND} Calculation Process

Algorithm 5: Optimal P^{OD} and P^{ND} calculation algorithm (*optiODND*)

Input : Discretized percentages for selecting optimal P^{OD} and P^{ND}
 $(\alpha^{OD,ND} = \{0, \Delta p, 2\Delta p, \dots, 1\})$, pre-defined weights for P^{OD} and P^{ND}
 $(\mathcal{W}^{OD,ND} = \{w^{OD}, w^{ND}\})$, temporary initialization dataset for z_j
 $(D_{z_j}^{init,temp})$, enriched initialization dataset for z_j ($D_{z_j}^{init,new}$), temporary
validation dataset for z_j ($D_{z_j}^{val,temp}$), original validation dataset for z_j
 $(D_{z_j}^{val,old})$, max polynomial basis function degrees of input variables
 (K_{long}, K_{lat}) .

Output : Optimal data percentage for original initialization dataset at z_j ($P_{z_j}^{OD}$),
optimal data percentage for temporary initialization dataset at z_j
($P_{z_j}^{ND}$), updated PCE model at z_j ($M_{z_j}^{PCE,new}$)

Initialize: Temporary P^{OD} : $P_{temp}^{OD} = 0.5$, temporary P^{ND} : $P_{temp}^{ND} = 0.5$, minimum
error summation of RMSE and APE: $s_{min} = \inf$.

```

1-1  $n_\alpha = \text{length}(\alpha^{OD,ND})$ 
1-2  $D_{z_j}^{val} = D_{z_j}^{val,old} \cup D_{z_j}^{val,temp}$ , obtain  $\hat{\mathbf{x}}^{val}, \hat{\mathbf{y}}^{val}$  from  $D_{z_j}^{val}$ .
1-3 for  $k = 1$  to  $n_\alpha$  do
1-4   for  $l = 1$  to  $n_\alpha$  do
1-5      $P_{temp}^{OD} = \alpha^{OD,ND}\{k\}$ 
1-6      $P_{temp}^{ND} = \alpha^{OD,ND}\{l\}$ 
1-7      $D_{z_j}^{init} = P_{temp}^{OD} D_{z_j}^{init,new} \cup P_{temp}^{ND} D_{z_j}^{init,temp}$ , obtain  $\mathbf{x}^{init}, \mathbf{y}^{init}, \mathbf{e}^{init}$  from  $D_{z_j}^{init}$ .
1-8      $[\mathbf{c}_i, M_{z_i}^{PCE}] = \text{LARS}(\mathbf{x}^{init}, \mathbf{y}^{init}, \mathbf{e}^{init}, K_{long}, K_{lat})$ 
1-9      $M_{z_i}^{PCE}$  validation:  $\hat{\mathbf{e}} = M_{z_i}^{PCE}(\hat{\mathbf{x}}^{val}, \hat{\mathbf{y}}^{val})$ 
1-10    Calculate  $RMSE_{\hat{\mathbf{e}}^{val}}$  and  $APE_{\hat{\mathbf{e}}^{val}}$  based on Equation (4.32), (4.33).
1-11     $s = w^{OD} RMSE_{\hat{\mathbf{e}}^{val}} + w^{ND} APE_{\hat{\mathbf{e}}^{val}}$ 
1-12    if  $s < s_{min}$  then
1-13       $M_{z_j}^{PCE,new} = M_{z_j}^{PCE}$ 
1-14       $P_{z_j}^{OD} = P_{temp}^{OD}$ 
1-15       $P_{z_j}^{ND} = P_{temp}^{ND}$ 
1-16       $s_{min} = s$ 
1-17    end if
1-18  end for
1-19 end for

```

optimal P^{OD} and P^{ND} are determined to be $P^{OD} = 20\%$ and $P^{ND} = 60\%$. Furthermore, the APE difference between no model update and updating with optimal P^{OD} and P^{ND}

is clearly illustrated in the right plot, highlighting the efficacy of PCE model updates in response to changing external conditions. In this scenario, transitioning from nominal, sunny weather to snowy conditions, PCE model updates enable the capture of weather changes, subsequently reducing prediction errors.

4.4 Simulation and Real-world Experimental Results

The proposed PCE uncertainty quantification technique is implemented using both simulation and real-world data to assess its performance. Section 4.4.1 details the simulation setup and Section 4.4.2 provides the results of PCE uncertainty quantification with and without model updates across various testing scenarios. Subsequently, Section 4.4.3 outlines the real-world experiment setup involving WATonoBus, and Section 4.4.4 evaluates the performance of PCE, and compares it with a Kalman filter baseline.

4.4.1 Simulation Setup

The PCE model simulation is conducted using MATLAB and its Driving Scenario Designer (DSD). In the simulation scenarios, vehicles and their position uncertainty are examined due to their ease of spawning and richer attributes in DSD compared to pedestrian actors. However, during real-world testing, the focus of the PCE model shifts to pedestrians, as initially proposed. In each scenario, reference objects marked in the HD map are represented as vehicles, as DSD does not support the spawning of objects such as traffic signs or light poles. Figure 4.6 illustrates a typical scenario scene, comprising a top view of the scenario and both cuboid and 3D ego-centric views. Notably, the ego vehicle is depicted in blue, the moving vehicle in red represents the object whose OPU needs quantification, and the yellow vehicles represent the known reference objects, standing in for light poles or stop signs in the real-world environment in this instance.

In total, the evaluation of PCE performance in a simulated environment comprises 9 initialization scenarios and 9 testing scenarios, designed using DSD. These scenarios vary in complexity and encompass diverse driving conditions to comprehensively assess the efficacy of the PCE model. For a comprehensive overview, detailed descriptions, model update requirement, and simulation durations of both the initialization and testing scenarios are presented in Tables 4.2 and 4.3, respectively, providing insights into the scope and extent of the simulations conducted.

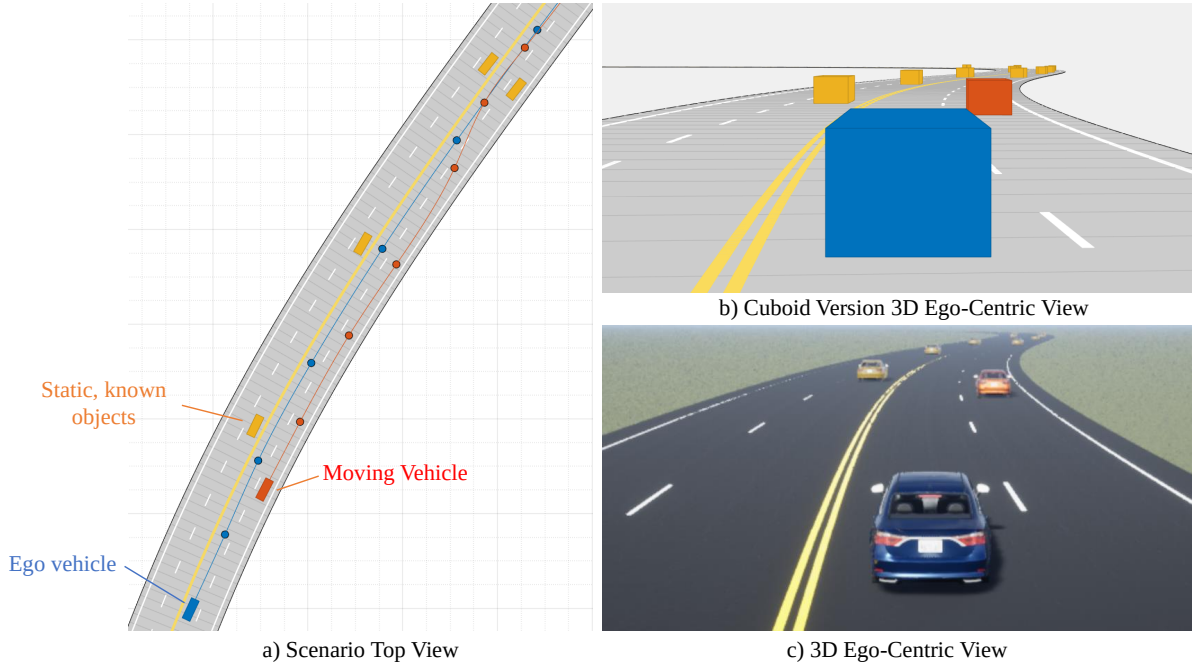


Figure 4.6: Example of a Simulation Scenario Scene

Table 4.2: Simulated Initialization Scenarios for PCE

	Description	Length [s]
1	There are 32 static vehicles positioned approximately 3.5 to 4m laterally from both sides of the ego vehicle, with their locations known. No moving vehicle to detect.	65
2	There are 32 static vehicles positioned approximately 3m laterally from both sides of the ego vehicle, with their locations known. No moving vehicle to detect.	65
3	There is 1 moving vehicle that stops in front of the ego vehicle lane, leaving enough distance for ego vehicle to detect. No reference vehicle in this scenario.	22
4	There are 32 static vehicles positioned approximately 4.5 to 5m laterally from both sides of the ego vehicle, with their locations known. No moving vehicle to detect.	65

5	There are 16 static vehicles positioned approximately 8m laterally away from the left side of the ego vehicle, with their locations known. No moving vehicle to detect.	65
6	All static vehicles are placed side by side for 3 rows on a straight road. No moving vehicle to detect.	22
7	Like scenario 6, however, instead of driving straight, ego vehicle performs lane change in this scenario. No moving vehicle to detect.	26
8	All static vehicles are placed on a curved road and ego vehicle is driving along this road. No moving vehicle to detect.	26
9	Ego vehicle is driving straight and there is a moving vehicle keeps performing double lane change in front of ego vehicle. No reference vehicle in this scenario.	127

Table 4.3: Simulated Testing Scenarios for PCE

	Description	Model Update	Length [s]
1	Ego vehicle is approaching to one stopping actor vehicle in front. No reference vehicle in this scenario.	Not required	12.5
2	There is one actor vehicle stopped in the right lane of the ego vehicle, positioned approximately 3.5 to 4.5 meters laterally away, as the ego vehicle approaches. No reference vehicle in this scenario.	Not required	12.5
3	There are 5 actor vehicles stopped in the right lane of the ego vehicle, positioned approximately 3.5 to 4.5 meters laterally away, as the ego vehicle approaches. No reference vehicle in this scenario.	Not required	45
4	There are 7 actor vehicles stopped at different lateral and longitudinal distances to the ego vehicle, as the ego vehicle approaches. No reference vehicle in this scenario.	Not required	35

5	One actor vehicle drives on the right lane relative to the ego vehicle, initially traveling faster than the ego vehicle before coming to a stop in front of it. No reference vehicle in this scenario.	Not required	40
6	One actor vehicle always drives in front of the ego vehicle. No reference vehicle in this scenario.	Not required	40
7	The actor vehicle initially drives in the left lane relative to the ego vehicle and later changes lanes to drive in front of the ego vehicle. Unlike the straight roads used in scenarios 1-6, scenario 7 is designed on a curved road. No reference vehicle in this scenario.	Not required	88
8	The actor vehicle initially drives in the right lane relative to the ego vehicle and later changes lanes to drive in front of the ego vehicle. After a while, the actor vehicle changes back to the right lane. This scenario is designed on a curved road and reference vehicles are spawned in this scenario.	Not required	76
9	There is one actor vehicle drives together with ego vehicle. They both perform lane changes on a curved road during simulation. Reference vehicles are spawned in this scenario.	Required	493

To prevent overfitting issue associated with initializing a single PCE model across all zones, distinct zones are established in this study, tailored to the corresponding initialization of PCE models. In our simulation setup, these object detection zones are delineated based on varying longitudinal distances while maintaining a consistent lateral distance span. Drawing upon existing data on stopping distances and velocities [168], as well as the criteria for maximum acceptable deceleration during confrontational driving [169], both y_1 and y_2 are defined as 15m and 30m respectively. The lateral distances x_l and x_r are set at 10m. Furthermore, considering the service environment and scope of WATonoBus in Section ??, we have established an emergency stop zone directly in front of the ego vehicle, spanning a longitudinal distance of $y_0 = 10m$. The determination of this zone is based on several factors, including the vehicle’s maximum deceleration capability, the typical reaction time required for the braking system to engage, and the speed limits within the operational environment. The 10-meter distance is calculated to allow the ego vehicle

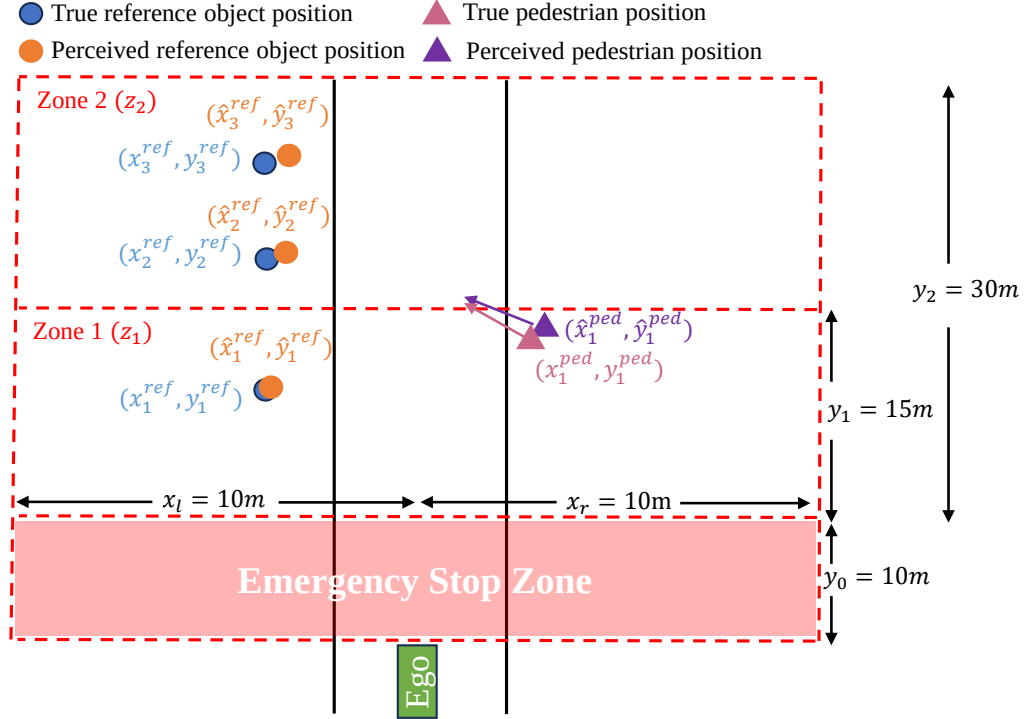


Figure 4.7: Detection Zone Representation for PCE Initialization and Testing in Simulation

sufficient time and space to decelerate safely and come to a full stop when an obstacle is detected in close proximity. This distance can be adjusted depending on variables such as road conditions, vehicle speed, and weather factors, which may influence braking performance. Unlike objects detected within the standard detection zones, those identified within the emergency stop zone are in immediate proximity to the ego vehicle and typically require an urgent response. This specialized zone is critical for ensuring prompt hazard mitigation, thereby enhancing overall safety. These parameters delineate the two detection zones utilized for initializing the PCE models in our simulation, as illustrated in Figure 4.7.

The true OPU in simulation is then formulated before collecting the initialization data for PCE. The magnitude of OPU in simulation is assumed to constitute 2 parts: model estimation uncertainty (e_{est}) and environment noises (e_{env}). Therefore, the OPU in simulation (e_{total}) is represented as below:

$$e_{total} = e_{est} + e_{env} = \underbrace{a_4 y^4 + a_3 y^3 + a_2 y^2 + a_1 y + a_0}_{e_{est}} + e_{env} \quad (4.35)$$

where y represents the longitudinal distance to ego vehicle, e_{est} is modeled with a 4th degree mono increasing polynomial equation with coefficients a_0 to a_4 , e_{env} is defined with a Gaussian noise as $e_{env} \sim N(0, \sigma^2)$ with mean 0 and variance σ^2 , and $e_{total} \geq 0$. The simulation OPU definition stems from our observation of the escalating impact of object position detection error with increasing distance between the ego vehicle and the object. This understanding underscores the significance of accurately quantifying the uncertainty associated with object positions, especially as they become more distant from the ego vehicle. Figure 4.8 below provides a visual representation of the resultant OPU in the PCE simulation, illuminating the relationship between distance and detection error.

4.4.2 Simulation Results

After configuring initialization and testing scenarios, constructing detection zones, and defining the true OPU in simulation, the surrogate models for both zones are initially established utilizing the UQLab toolbox [170]. Subsequently, they undergo initialization and validation processes based on the normalized empirical error (ε_{emp}) and LOO cross-validation error (ε_{LOO}), as outlined in Section 4.1.4. The validation performance of PCE,

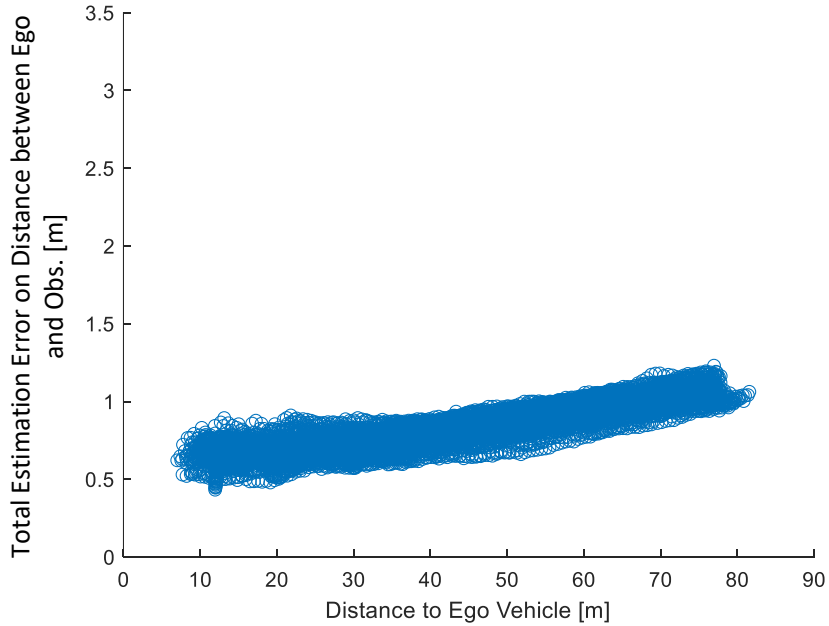


Figure 4.8: Visualization of OPU in PCE Simulation

bPCE, and PCK are first investigated to select the most suitable approach to implemented for quantifying OPU with the simulation and later real-world data. The results are tabulated in Table 4.4.

Table 4.4: Validation Results of PCE, bPCE, PCK in Simulation

Model	Normalized Empirical Error (ε_{emp})	LOO Cross-validation Error (ε_{LOO})
$M_{z_1}^{PCE}$	0.0151	0.188
$M_{z_1}^{bPCE}$	0.181	0.184
$M_{z_1}^{PCK}$	1.988	0.473
$M_{z_2}^{PCE}$	0.236	0.242
$M_{z_2}^{bPCE}$	0.241	0.247
$M_{z_2}^{PCK}$	1.854	0.54

Based on the validation results, it's evident that PCK is not suitable for this application, as both its ε_{emp} and ε_{LOO} are considerably higher compared to the other two candidates. Moreover, the initialization time required for the PCK model is significantly longer, putting it at a disadvantage for subsequent online model updates. Conversely, both the PCE and bPCE models, operating across two distinct detection zones, demonstrate similar error levels. However, considering the additional computational demands associated with bPCE due to bootstrap resampling, particularly for large datasets, the performance of the PCE model is deemed sufficient for quantifying OPU in this scenario. Hence, PCE is chosen as the preferred uncertainty quantification technique, and its simulation and real-world performance will undergo further investigation.

These errors of $M_{z_1}^{PCE}$ and $M_{z_2}^{PCE}$ are tabulated in Table 4.5 for analysis. Furthermore, to facilitate visualization, both prediction results and ground truth data of each validation sample are graphically depicted in Figure 4.9 using histogram and validation scatter plots. Notably, the validation results concerning OPU indicate a high level of agreement between the PCE model predictions and the ground truth. Specifically, for z_1 , the normalized empirical error and LOO error are recorded as 0.151m and 0.188m, respectively. Those of z_2 exhibit slightly higher values, reaching 0.236m and 0.242m, respectively. In addition, to verify the overfit issue encountered by not splitting PCE model detection zones, errors of M_{entire}^{PCE} are also tabulated and plotted in Table 4.5 and Figure 4.9, respectively. Its normalized empirical error and LOO error are at 0.652m and 0.688m.

After reviewing the validation results, the initialized PCE models undergo testing us-

Table 4.5: Validation Results of PCE with and without Detection Zone Divisions

Model	Normalized Empirical Error (ε_{emp})	LOO Cross-validation Error (ε_{LOO})
$M_{z_1}^{PCE}$	0.151	0.188
$M_{z_2}^{PCE}$	0.236	0.242
M_{entire}^{PCE}	0.652	0.688

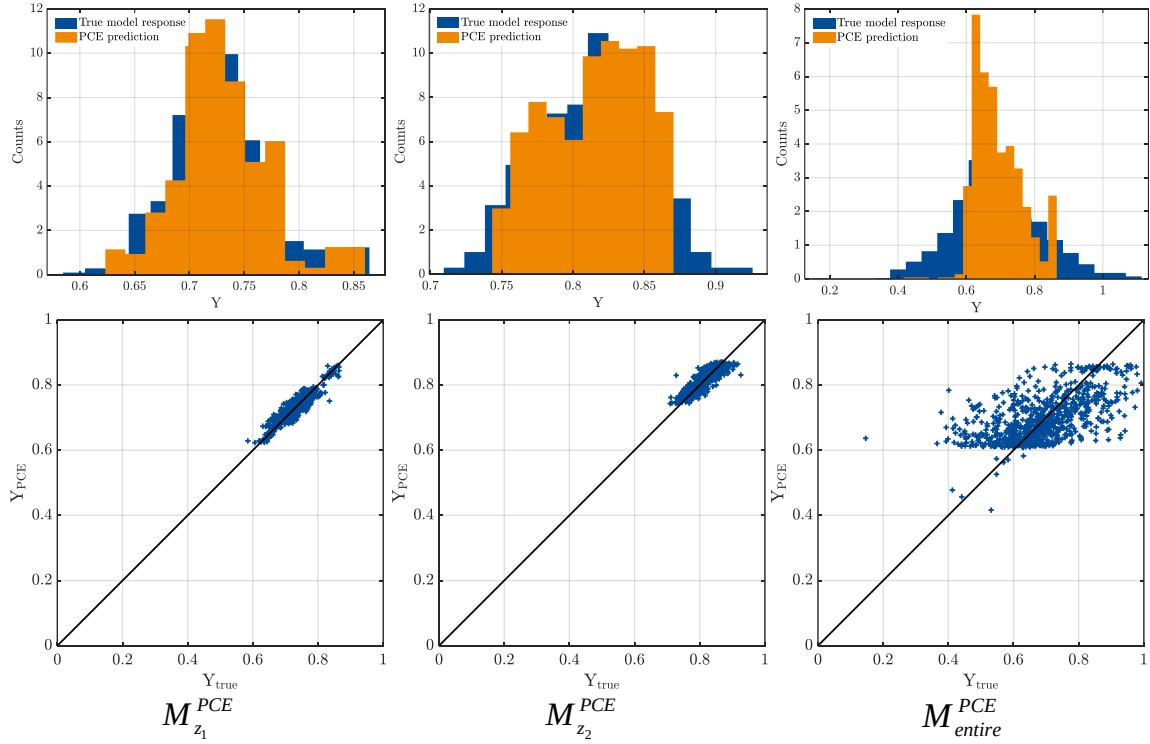


Figure 4.9: Plots of PCE Validation Results from $M_{z_1}^{PCE}$, $M_{z_2}^{PCE}$ and M_{entire}^{PCE} in Simulation

ing the first 8 testing scenarios outlined in Table 4.3, as they do not necessitate online model updates. The resulting MAE and RMSE values for PCE models initialized at both detection zones are presented in Table 4.6. Among all testing outcomes, the minimum MAE and RMSE are observed in scenario 3, zone 2, with values of 0.0089m and 0.0109m, respectively. Conversely, scenario 7, zone 2, yields the highest MAE and RMSE, reaching 0.1190m and 0.1213m, respectively.

Table 4.6: Testing Results of PCE without Online Model Update

Scenario #	Detection Zone	MAE	RMSE
1	z_1	0.0133	0.0264
	z_2	0.0178	0.0198
2	z_1	0.0097	0.0109
	z_2	0.0165	0.0203
3	z_1	0.0157	0.0194
	z_2	0.0089	0.0115
4	z_1	0.0169	0.0224
	z_2	0.0170	0.0209
5	z_1	0.0132	0.0164
	z_2	0.0146	0.0175
6	z_1	0.0147	0.0189
	z_2	0.0130	0.0164
7	z_1	0.0803	0.0862
	z_2	0.1190	0.1213
8	z_1	0.0308	0.0407
	z_2	0.0481	0.0596

The PCE model then undergoes testing in scenario 9, which is distinguished by its notably extended duration compared to other testing scenarios. This prolonged simulation period facilitates the accumulation of a greater volume of new detections on reference objects, enriching both the initialization dataset ($D^{init,temp}$) and the validation dataset ($D^{val,temp}$) used in Algorithm 5, thereby rendering model updates more feasible. Notably, scenario 9 is specifically engineered to necessitate model updates during the testing process. This deliberate design choice entails a modification to the true OPU, wherein its magnitude is doubled during the second half of the testing scenario (See Figure 4.10). This adjustment serves to challenge the PCE model’s adaptability and responsiveness in dynamically evolving environments, thereby enabling a comprehensive evaluation of its efficacy and potential to serve in real-world operation.

In the second half of the testing scenario, a deliberate modification is introduced to sim-

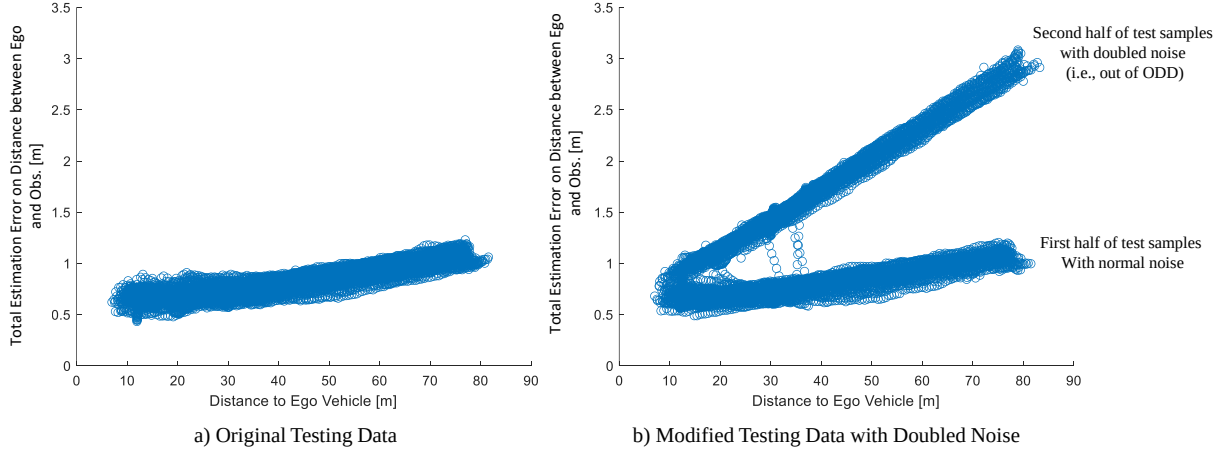


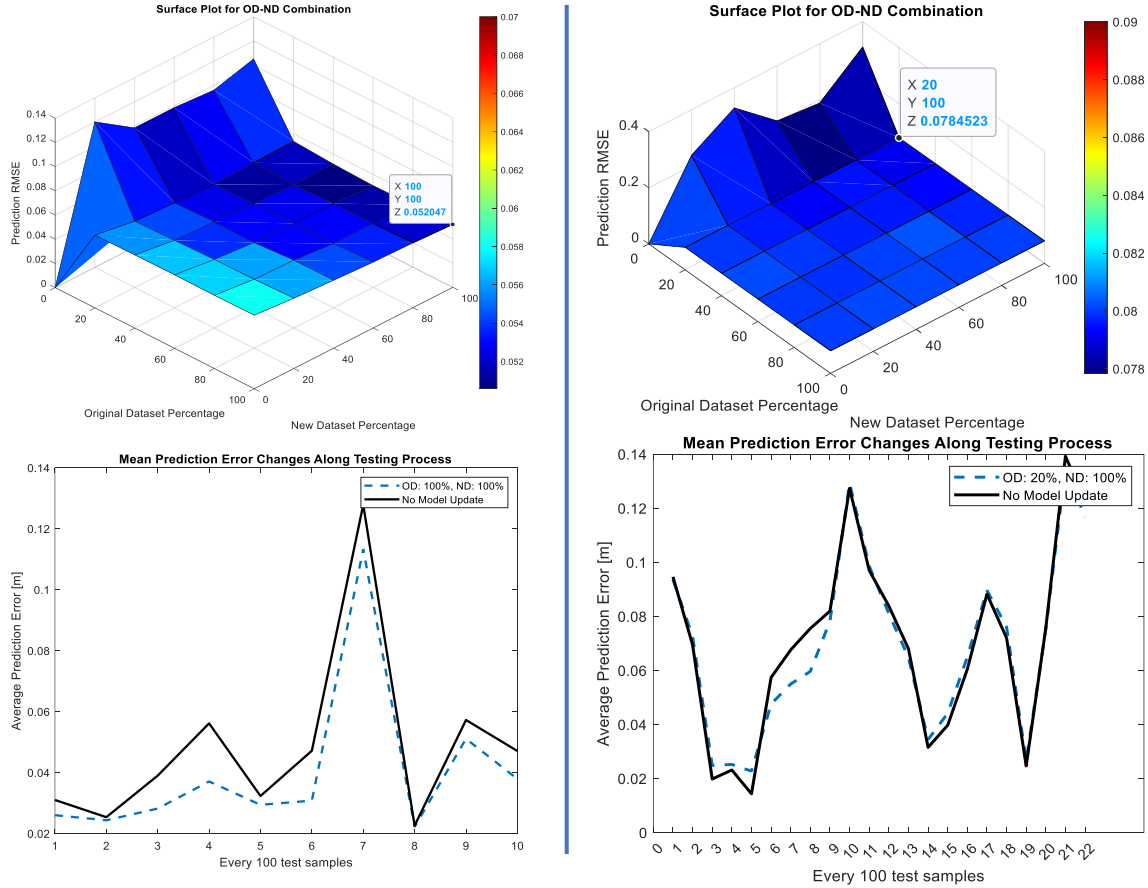
Figure 4.10: Comparison between a) Normal OPU and b) Doubled OPU in PCE Simulation

Table 4.7: Optimal P^{OD} and P^{ND} for Two Detection Zones

Original Initialization (OD) Dataset Percentage		New Initialization (ND) Dataset Percentage	
$P_{z_1}^{OD}$	100%	$P_{z_1}^{ND}$	100%
$P_{z_2}^{OD}$	20%	$P_{z_2}^{ND}$	10%

ulate increasingly challenging operating conditions for the object detection algorithm. This adjustment serves to evaluate the PCE model’s capacity to dynamically adapt to changing conditions by identifying the optimal P^{OD} and P^{ND} . Employing the $optiODND(\cdot)$ algorithm, specifically with the threshold parameter, $n^{init,thres}$, set at 350, allows for the determination of these optimal percentages, which are tabulated in Table 4.7. Moreover, the resulting $RMSE_{\hat{e}_{val}}$ and $APE_{\hat{e}_{val}}$ values for the two detection zones, equipped with the determined percentages, are depicted in Figure 4.11. Notably, the RMSE plots reveal that, with the selected percentages, the error values for these two detection zones reach 0.052m and 0.078m, respectively. Furthermore, the APE results are juxtaposed against those without model updates to illustrate the enhancements achieved through model updates. This emphasizes the PCE model’s inherent adaptability, enabling swift adjustments to out-of-distribution data shifts or changes in service conditions.

Moreover, the actual OPU quantification results, alongside the corresponding simulation time of z_1 and z_2 are recorded and depicted in Figure 4.12. These plots present a comprehensive visualization of the OPU estimation process, wherein the true OPU error is



a) RMSE and APE Results for the First Detection Zone

b) RMSE and APE Results for the Second Detection Zone

Figure 4.11: RMSE and APE Results Plots for a) First and b) Second Detection Zones with Optimal P^{OD} and P^{ND}

represented in blue, OPU quantification results with model updates are denoted in green, and those without model updates are indicated in red. Additionally, vertical black dashed lines delineate the instances of model updates, providing a clear temporal reference. Notably, a single vertical red dashed line is introduced to mark the commencement of the second half phase of the testing scenario, coinciding with the doubling of the true OPU magnitude. These plots vividly demonstrate the efficacy of PCE model updates in mitigating the impact of increased noise levels. Specifically, following approximately 3 and 2 model updates for $M_{z_1}^{PCE}$ and $M_{z_2}^{PCE}$, respectively, the quantified OPU converges towards the true OPU result in the presence of doubled noise. Conversely, in the absence of

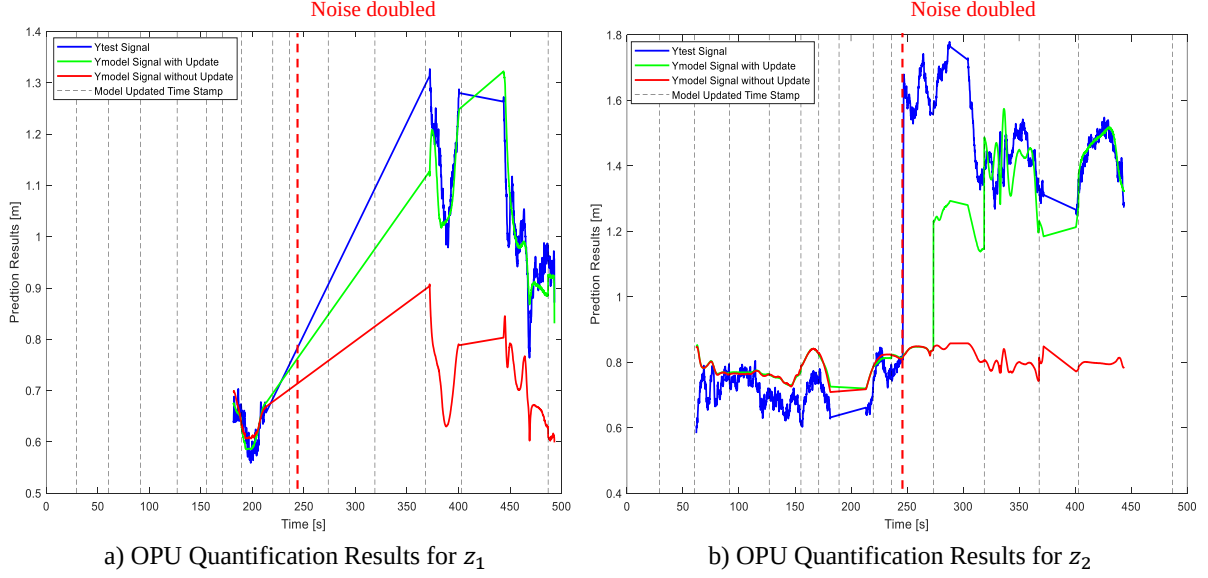


Figure 4.12: OPU Quantification Results for Both Detection Zones in Simulation

model updates, a noticeable discrepancy between the estimated and true results becomes apparent.

4.4.3 Real-world Experiment Setup

The real-world experiment utilizes rosbag data obtained from the WATonoBus platform, as detailed in Section 3.3.1, which expounds on the system scope, service environment, and functionalities of WATonoBus. The initialization dataset is gathered during sunny weather conditions, with a sampling frequency of 10Hz, along the first half of the ring road drive (refer to Figure 4.13a). For OPU quantification on the encountered pedestrians, the stop signs positioned along the ring road are chosen as reference objects, benefiting from their precise locations recorded in the HD map, primarily to aid the decision-making module for intersection handling. Subsequently, another testing dataset is collected during sunny weather conditions along the second half of the ring road drive, characterized by increased pedestrian engagement (refer to Figure 4.13b). This testing dataset is sampled at a frequency of 1Hz, obviating the need for model updates, and will serve to assess the OPU quantification accuracies of the initialized PCE models under identical weather conditions. Furthermore, to evaluate the advantage conferred by PCE model updates in response to

changing perception operation conditions, yet another testing dataset is recorded under snowy conditions at a frequency of 10Hz (refer to Figure 4.13c).



a) Three Camera Views in Initialization Dataset



b) Three Camera Views in Testing Dataset that Not Requires Model Update



c) Three Camera Views in Testing Dataset that Requires Model Update

Figure 4.13: Three Front Camera Views in a) Initialization Dataset, b) Testing Datasets that Not Requires PCE Model Update, c) Testing Datasets that Requires PCE Model Update

In contrast to the PCE simulation setup, the real-world scenario presents a higher likelihood of encountering pedestrians on the right side of the driving lane due to the presence of pedestrian sidewalks positioned on that side. Consequently, the regions to the left and right of the ego vehicle are designated as distinct detection zones. As illustrated in Figure 4.14, this segmentation results in a total of 4 detection zones. The lateral distances,

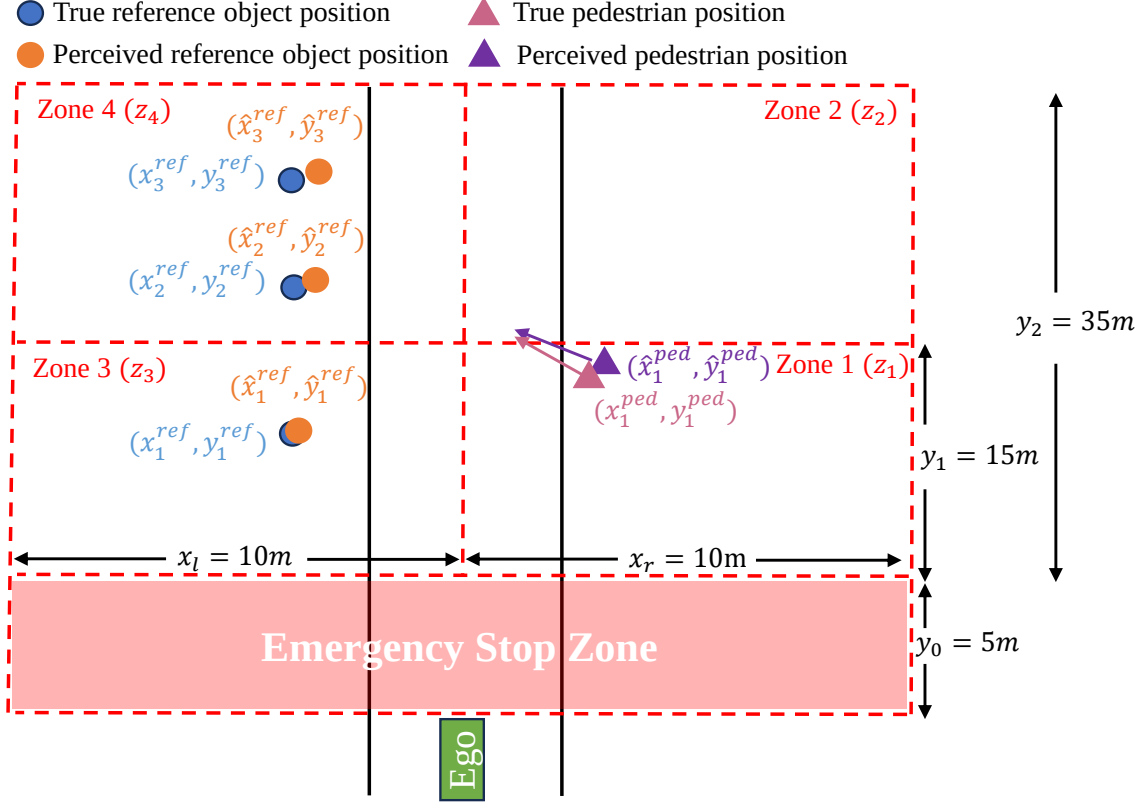


Figure 4.14: Detection Zone Representation for PCE Initialization and Testing in Real World Experiment

denoted as x_l and x_r , remain consistent at 10m. However, the longitudinal distances, represented by y_0 , y_1 , and y_2 , are set at 5m, 15m, and 35m, respectively.

While we cannot manually derive the true OPU for the real-world datasets, we can still plot the sample data distribution to visualize the distribution of OPU concerning longitudinal and lateral distances relative to the ego vehicle. As depicted in Figure 4.15 a), there appears to be a proportional relationship between OPU magnitude and longitudinal distance. Specifically, the OPU magnitudes in z_2 and z_4 are higher than those in z_1 and z_3 , potentially attributed to the diminishing number of pixels or resolution of detected objects at further distances. This finding aligns with our formulated true OPU in PCE simulation, as illustrated in Figure 4.8. Furthermore, the distribution pattern in Figure 4.15 b) clearly indicates a higher prevalence of pedestrians detected in z_1 and z_2 , owing to the presence

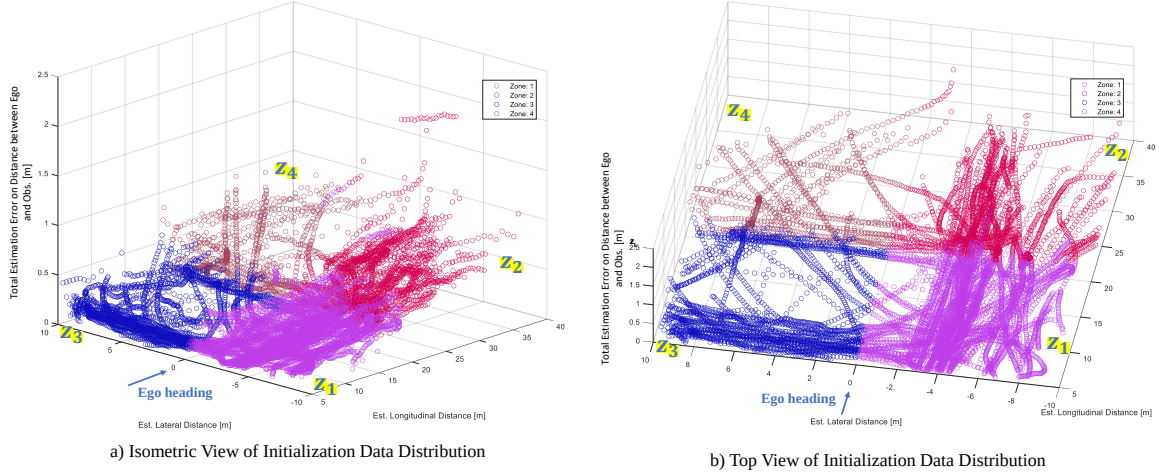


Figure 4.15: Real World Initialization Dataset Distribution in a) Isometric View and b) Top View

of the pedestrian sidewalk on the right side of the driving lane. In contrast, pedestrian detection occurrences in z_3 and z_4 are considerably fewer and typically involve instances of pedestrians crossing the road. This observation underscores the rationale behind the additional division of detection zones based on lateral distances for real world experiments.

4.4.4 Real-world Experiment Results

The proposed PCE uncertainty quantification technique is then applied to real-world datasets for initialization, validation, and testing to assess its effectiveness and robustness. Following model initialization, the MAE and RMSE of the validation results are documented in Table 4.8. These validation outcomes affirm the capability of the initialized PCE models to accurately quantify pedestrian OPU across various detection zones. Notably, the highest MAE and RMSE values are observed in z_2 , reaching 0.155m and 0.224m, respectively, mostly attributable to the increased distance between the detected objects and the ego vehicle. Conversely, z_3 exhibits the lowest MAE and RMSE values, recorded at 0.063m and 0.086m, respectively. Transitioning to the testing phase, $M_{z_1}^{PCE}$ and $M_{z_3}^{PCE}$ continue to demonstrate commendable performance, albeit with slightly increased errors sometimes. $M_{z_2}^{PCE}$, while exhibiting a slight reduction in errors compared to the validation phase, still presents relatively high errors with an MAE of 0.128m and an RMSE of 0.167m. $M_{z_4}^{PCE}$ shows a substantial increase in errors during the testing phase compared to

Table 4.8: MAE and RMSE of PCE Real-world Validation and Testing Results with Sunny Weather

PCE Model	Detection Zone	Validation MAE	Validation RMSE	Testing MAE	Testing RMSE
$M_{z_1}^{PCE}$	z_1	0.112	0.166	0.105	0.160
$M_{z_2}^{PCE}$	z_2	0.155	0.224	0.128	0.167
$M_{z_3}^{PCE}$	z_3	0.063	0.086	0.074	0.104
$M_{z_4}^{PCE}$	z_4	0.116	0.151	0.265	0.231

validation, with an MAE of 0.265m and an RMSE of 0.231m, indicating notable challenges in accurately quantifying OPU in this detection zone. The challenge can stem from the lack of similar samples in z_4 because it has less samples than closer zones like z_1 and z_3 . Overall, the results underscore the varying performance levels across different detection zones, with z_1 and z_3 consistently demonstrating superior accuracy, while z_2 and z_4 exhibit higher error rates, particularly during the testing phase.

The PCE model undergoes further testing using a dataset captured during snowy weather conditions, aimed at evaluating the efficacy of model updates in enhancing uncertainty quantification performance in real-world experiments. The resulting RMSE values for three distinct cases are presented in Table 4.9, which include scenarios of no model update, model updates without new data (i.e., based on original initialization dataset), and model updates with new data during online testing. $M_{z_1}^{PCE}$ demonstrates an initial RMSE of 0.097m without any model updates, which decreases to 0.081m with updates but no new data. The most significant improvement, with an RMSE of 0.070m, is observed when updates are performed with new data, highlighting the substantial enhancement in accuracy with new data incorporation. Conversely, $M_{z_2}^{PCE}$ exhibits notably higher RMSE values across all model update scenarios. Without any model updates, the RMSE is 0.322m, indicating relatively high errors in OPU quantification. However, implementing model updates leads to a reduction in RMSE, with values of 0.269m and 0.211m for cases without new data and with new data, respectively. Despite the improvements, $M_{z_2}^{PCE}$ still exhibits higher errors compared to other zones, underscoring the challenges in accurately quantifying OPU in this detection zone under snowy weather conditions. In $M_{z_3}^{PCE}$, the RMSE values are comparatively lower across all model update scenarios. Without any updates, the RMSE is 0.096m, which decreases slightly to 0.125m with model updates but without new data. Implementing model updates with new data results in the lowest RMSE of 0.092m, indicating improved accuracy. Finally, for $M_{z_4}^{PCE}$, without any model updates, the

Table 4.9: RMSE of PCE Real-world Testing Results with Different Model Update Cases and Snowy Weather

PCE Model	Detection Zone	No Model Update RMSE	Model Update without New Data RMSE	Model Update with New Data RMSE
$M_{z_1}^{PCE}$	z_1	0.097	0.081	0.070
$M_{z_2}^{PCE}$	z_2	0.322	0.269	0.211
$M_{z_3}^{PCE}$	z_3	0.096	0.125	0.092
$M_{z_4}^{PCE}$	z_4	0.342	0.391	0.132

RMSE is 0.342m, which increases to 0.391m with model updates but without new data. However, incorporating new data during model updates leads to a substantial reduction in RMSE to 0.132m, indicating a significant improvement in accuracy. Despite the initial high errors, model updates with new data prove to be highly effective in z_4 , enhancing the accuracy of OPU quantification in under snowy weather conditions.

To better understand the substantial error reduction in z_4 , the APE errors of these three model update cases are plotted with every 20 test samples in Figure 4.16 a). In this case, the dataset percentages used in model update are $P_{z_4}^{OD} = 100\%$ and $P_{z_4}^{ND} = 100\%$. Particularly, blue plot represents model update with new data, red plot represents model update without new data, and yellow plot is no model update. Additionally, the model updates time stamps are plotted with the vertical black dash lines in the plot. Considering the coarser data distribution seen in Figure 4.16 b), there are few times of model updates compared to other detection zones. However, model update with new data from the start helps a lot in reducing the initial APE, leading it to the lowest APE error. Furthermore, although updating the model without new data performs slightly better performance than no model update, solely relying on the historical knowledge on model update can possibly cause overfitting issue and it can be found from the second half of the red plot, where its APE error is even much higher than no model update results.

In the snowy condition testing results, another comparison is made between the proposed PCE model and a baseline model: a constant acceleration Kalman filter designed for object tracking tasks within the perception module. The results are tabulated in Table 4.10. Mathematically, the Kalman filter recursively estimates the state of a dynamic system amidst uncertain measurements, adept at incorporating noisy inputs and predicting future object states based on past observations. As a result, it finds extensive application

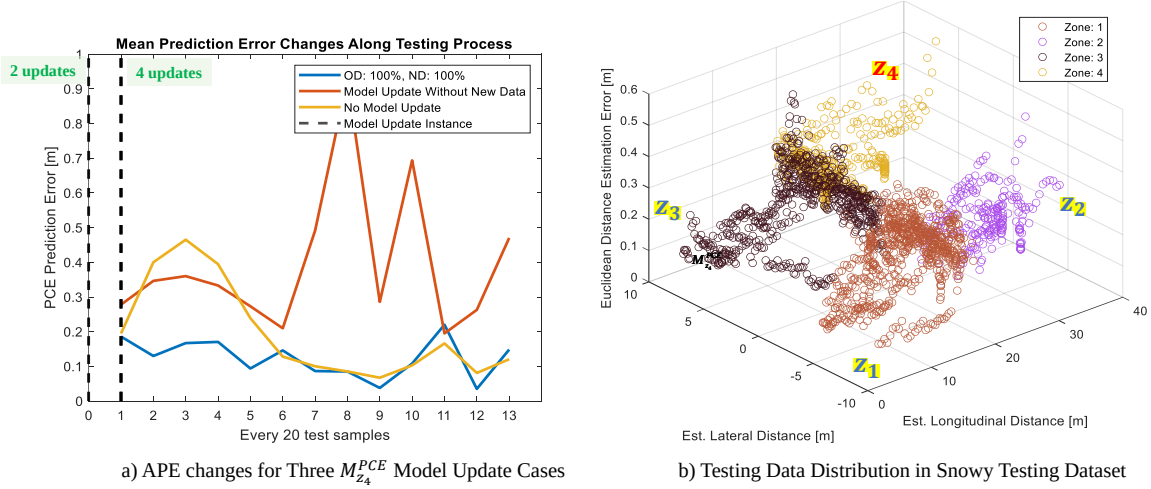


Figure 4.16: Real World Snowy Testing Dataset a) APE Results of $M_{z_4}^{PCE}$ and b) Data Distribution Visualization

Table 4.10: Comparison Results between PCE and Baseline with Snowy Testing Dataset

Evaluation Criteria	PCE	Baseline	Improvement
Overall RMSE	0.131m	0.193m	-0.062m
True object intersection rate	97.59%	82.46%	+15.13%

across diverse domains including navigation systems, aerospace engineering, and robotics. First, considering RMSE, the baseline model yields an RMSE of 0.1928m, whereas the PCE model demonstrates a significantly lower RMSE of 0.1308m, marking a reduction of 0.062m in error. In addition, another evaluation criteria called true object intersection rate is designed to assess the intersection between the true pedestrian’s representation (a circle with a radius of 0.2m) and the predicted variance by each method. A higher intersection rate signifies better alignment between predicted uncertainty and the true object’s location (See Figure 4.17). The baseline model achieves an intersection rate of 82.46%, while the PCE model outperforms it with an intersection rate of 97.59%, indicating a substantial improvement of 15.13%. The comparison demonstrates that the PCE model substantially outperforms the baseline Kalman filter in this aspect, particularly in snowy conditions, indicating its ability to provide more accurate uncertainty estimates.

Finally, runtime analysis confirms that the PCE uncertainty quantification is time-efficient, successfully meeting the real-time operational requirement of 10-20 Hz in a MAT-



Figure 4.17: Visualization of Predicted Variance and True Object Intersection

LAB environment with 16GB of RAM and an AMD Ryzen 9 5900X 12-core processor. The runtime performance could be further optimized by compiling the code in C/C++, which would enable faster execution and more efficient resource utilization.

4.5 Result Analysis and Discussion

This section delves into key observations obtained from the simulation and real-world experiments, aiming to elucidate the underlying factors behind these findings. Subsequently, potential future research opportunities regarding the proposed PCE uncertainty quantification method will be explored.

Notably, as depicted in Table 4.4, the PCE model outperforms the PCK model across both designated detection zones. Moreover, the computational efficiency of PCE surpasses that of PCK. Compared to PCK, PCE is more flexible in modeling nonlinear systems (i.e., OPU generation mechanism from object detection algorithm). While PCE can adeptly capture nonlinear relationships between inputs and system outputs using higher-order polynomial expansions, PCK is constrained by its assumption of linearity, potentially limiting its ability to represent complex system responses. Furthermore, PCE demonstrates enhanced efficacy in handling sparsely sampled data because it allows for the use of sparse sampling techniques. In contrast, PCK relies on the availability of densely sampled data and this drawback can lower its accuracy when the validation or testing data are sparsely sampled. Moreover, the longer processing time associated with PCK can be attributed to PCE’s superior efficiency in handling high-dimensional systems through the utilization of high-order polynomial basis functions.

Based on the comparison between $M_{z_i}^{PCE}$ and M_{entire}^{PCE} , it becomes evident that dividing the initialization dataset yields improved generalization performance. This strategy allows each initialized $M_{z_i}^{PCE}$ model to capture different aspects of the underlying system

responses, leading to enhanced quantification accuracy of OPU across various detection zones. Additionally, this approach facilitates efficient exploration of the high-dimensional parameter space, mitigating the risk of model overfitting observed in the histogram of M_{entire}^{PCE} , as depicted in Figure 4.9. In this histogram, the concentration of histogram bins in the middle indicates a disregard for marginal data, emphasizing the importance of segmenting the initialization dataset based on detection zones to foster better model performance.

In Table 4.3, the highest MAE and RMSE values are observed in testing scenario 7, indicating a notable performance disparity. This inconsistency may stem from encountering unfamiliar testing sample data, particularly as scenario 7 unfolds on a curved road with moving vehicles, whereas all initialization scenarios are based on a straight road. Consequently, the transition to a different service environment during testing can potentially diminish the accuracy of uncertainty quantification. This underscores the significance of model updating during online testing processes, as illustrated in Figure 4.12. By judiciously integrating both original and new initialization datasets, the PCE models can continuously assimilate new insights into environmental changes while retaining foundational knowledge derived from historical data. Consequently, during operation with novel conditions, online PCE model updating facilitates enhanced accuracy in OPU quantification.

Referring to Table 4.8, the disparities between real-world validation and testing outcomes primarily stem from encounters with unfamiliar samples. Notably, the model performance of $M_{z_4}^{PCE}$ in scenario 4 exhibits challenges during the testing phase, despite demonstrating accuracy during validation. Upon reviewing and contrasting the initialization and testing datasets collected under sunny weather conditions, it becomes apparent that many of the unseen samples in the testing dataset coincide with a sharply curved road segment featuring a bustling pedestrian presence on the sidewalk. These previously unseen samples in z_4 are circled in Figure 4.18a) and b) to demonstrate the differences.

In Table 4.9, the RMSE results of different model update cases are reviewed and compared to examine which case can improve the uncertainty quantification accuracy most effectively. The key advantage of updating the PCE model with both old and new initialization data is that it provides the most comprehensive and up-to-date representation of the system. By continuously integrating new observations, the model can adjust to changes in the underlying system behavior, improving its predictive accuracy and robustness. The problems with the other two approaches stem from their inability to account for changes in the system over time. Obviously, the "No Model Update" approach does not consider any new information or changes in the system over time. As a result, the model may become outdated and less accurate as the system evolves. While updating the model with historical data helps to incorporate some of the system's evolution, it may not capture changes or

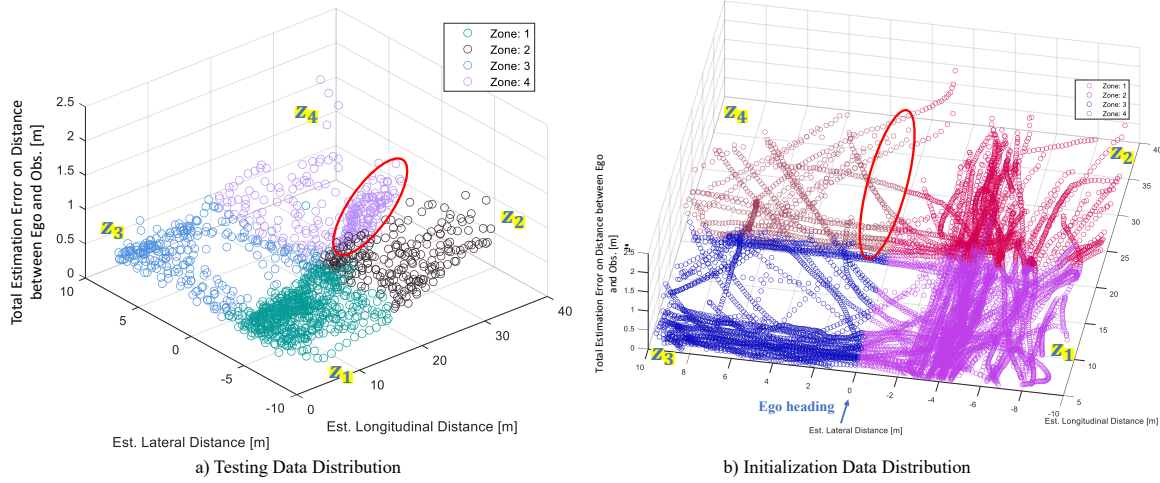


Figure 4.18: z_4 Data Distribution Comparison for a) Testing Dataset and b) Initialization Dataset with Sunny Weather

trends that occur after the initial data collection. The model may still lack adaptability to new conditions or variations in the system. In summary, updating the PCE model with both historical and newly collected initialization datasets is preferred because it ensures that the model remains accurate and adaptable to changes in the system over time. This approach provides the best balance between capturing past trends and incorporating new observations, leading to more reliable predictions and improved performance.

In the end of real-world experiment, PCE is compared with a constant acceleration Kalman filter to examine its OPU quantification performance. The reason why PCE outperforms the baseline is due to the nonlinear dynamics of the object's motion, especially when the pedestrian is occluded or leaving the scene. PCE can approximate complex nonlinear dynamics using polynomial expansions, providing a more accurate representation of the system's behavior. In contrast, constant acceleration Kalman filter relies on linear model assumptions, which may lead to inaccuracies when dealing with nonlinear systems or non-Gaussian noise.

Although the proposed PCE method has demonstrated promising performance in quantifying OPU across both simulations and real-world experiments, there are avenues for further enhancement. Firstly, while the current focus lies on pedestrian OPU, it's imperative to extend this analysis to encompass other traffic participants such as vehicles. Achieving this entails conducting additional studies to elucidate the transfer of uncertainty understanding from reference objects to dynamic entities. While our work assumes that the OPU of stop signs can adequately represent pedestrian OPU due to their similar size and

reflection rate, this assumption may not hold true for vehicles, necessitating further investigation. Secondly, the versatility of the proposed PCE method warrants its application in quantifying other forms of uncertainty to ascertain its generality. For instance, it can be utilized to quantify localization uncertainty based on lane boundary detection and HD map knowledge of lane markings. Thirdly, verifying the uncertainty correlation assumption between pedestrian and stop signs is also a valuable future task, even if we have observed promising uncertainty quantification performance with the help of stop sign information in the real-world experiments. Moreover, while our study treats weather changes as triggers for degradation in perception accuracy requiring model updates, future investigations could explore other operational condition changes such as variations in light conditions or road layouts to better assess the efficacy of model updates in improving uncertainty quantification.

Chapter 5

Uncertainty-aware Decision Making

Learning-based algorithms are crucial in enhancing the capabilities of autonomous driving vehicles. Unlike rule-based systems, which follow predefined instructions, learning-based algorithms acquire knowledge from data, introducing unique challenges related to uncertainty. Assessing these uncertainties and their impacts on other functional modules within autonomous vehicles remains a significant challenge [12, 171]. The effects of uncertainty in learning-based algorithms can be profound, potentially leading to incorrect or suboptimal decisions in critical situations. Therefore, it is essential to quantitatively evaluate how uncertainty permeates through an ADS and establish rigorous criticality measures [97]. This chapter introduces a method called Dynamic Bayesian Network and Influence Diagram (DBNID) to incorporate uncertainty propagation effects into the decision-making process, aiming to generate safer decisions for downstream path planning and low-level motion control modules. By doing so, the ultimate objective is to enhance the overall safety performance of autonomous driving. In the following sections, Section 5.1 will introduce the background of both Bayesian network and influence diagram. Afterwards, the uncertainty system propagation effect is investigated using Dynamic Bayesian Network (DBN) in Section 5.2. Considering the uncertainty propagation effect, Section 5.3 then formulates an uncertainty-aware decision-making method using Influence Diagram (ID). Section 5.4 then examines the proposed decision-making method with simulation and on-vehicle experiments, followed by the result discussion in Section 5.5.

5.1 Bayesian Network and Influence Diagram Fundamentals

In the following subsections, we delve into the fundamentals of Dynamic Bayesian Networks (DBNs) and Influence Diagrams (IDs), two advanced probabilistic modeling techniques employed in this work for decision making under uncertainty. Specifically, DBNs are an extension of Bayesian Networks (BNs) that incorporate the dimension of time, allowing for the modeling of temporal processes and the evolution of variables over sequential time steps. This makes DBNs particularly valuable for applications involving time series data and dynamic systems like ADS applications (See Section 5.1.1). Influence Diagrams (IDs), on the other hand, build upon BNs by integrating decision and utility nodes, providing an interpretable framework for making optimal decisions under uncertainty. These diagrams not only represent probabilistic dependencies but also factor in the preferences and outcomes associated with different decisions (See Section 5.1.2). Together, DBNs and IDs offer robust methodologies for handling complex, uncertain environments in a wide range of applications.

5.1.1 Bayesian Network (BN) and Dynamic Bayesian Network (DBN)

In this section, the fundamentals of BN and DBN will be introduced. BNs are probabilistic graphical models that represent a set of variables and their conditional dependencies using a directed acyclic graph. Each node in the graph represents a random variable, and edges indicate conditional dependencies between these variables, with associated conditional probability distributions defining these relationships. The formal definition of BNs consists of the following [5]:

- A set of variables and a set of directed edges between variables.
- Each variable has a finite set of mutually exclusive states.
- The variables with their directed edges form an directed acyclic graph (DAG).
- To each variable A with parents $B_1, \dots, B_n$, a conditional probability table $P(A|B_1, \dots, B_n)$ is attached. If A has no parents, then the table is reduced to $P(A)$.

Figure 5.1 shows an example of a BN with 7 variables and their corresponding dependencies, represented by different directed edges. In this example, the probabilities needed

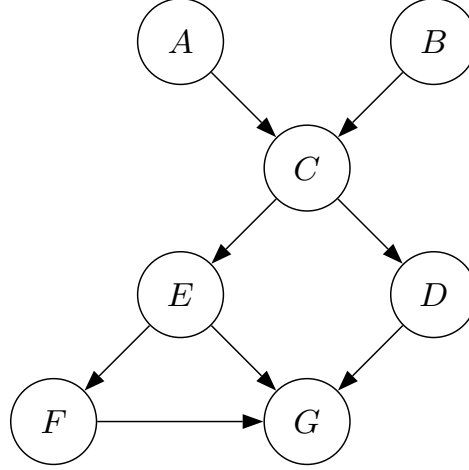


Figure 5.1: BN Example [5]

to be specified are $P(A)$, $P(B)$, $P(C|A, B)$, $P(E|C)$, $P(D|C)$, $P(F|E)$, and $P(G|D, E, F)$.

The chain rule can be applied for probability distributions over sets of variables. For a BN over a variable set of $\mathcal{U} = \{A_1, \dots, A_n\}$, then the joint probability distribution $P(\mathcal{U})$ given all conditional probability tables can be written as:

$$P(\mathcal{U}) = \prod_{i=1}^n P(A_i | pa(A_i)) \quad (5.1)$$

where $pa(A_i)$ are the parents variables of A_i in the BN. Sometimes, a variable in BNs can be observed as finding. If a discrete variable A has n states, a finding on A is an n -dimensional table of zeros and ones. Hence, with findings $\mathbf{e}_1, \dots, \mathbf{e}_m$, the chain rule can be modified as:

$$P(\mathcal{U}, \mathbf{e}) = \prod_{A \in \mathcal{U}} P(A | pa(\mathcal{U})) \cdot \prod_{i=1}^m \mathbf{e}_i \quad (5.2)$$

and for $A \in \mathcal{U}$, $P(A|\mathbf{e})$ can be written as:

$$P(A|\mathbf{e}) = \frac{\sum_{\mathcal{U} \setminus \{A\}} P(\mathcal{U}, \mathbf{e})}{P(\mathbf{e})} \quad (5.3)$$

When working with domains that evolve over time, discrete time stamps can be added to the BN model as time slice. The time slices are connected through temporal links to

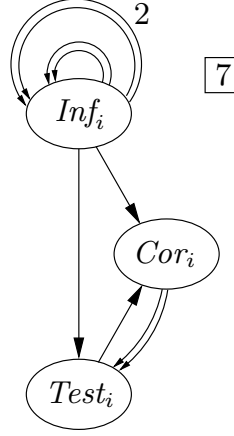


Figure 5.2: DBN Example [5]

each other and constitute a full model. If the temporal link among each time slice and their conditional probabilities are identical among all time slices, this type of BN model is called Dynamic Bayesian Network (DBN) model. DBNs represent temporal processes by connecting nodes across different time slices and are good at capturing how the state of variables evolves. This makes DBNs suitable for time series analysis such as monitoring the ADS functional modules' performance over a period or speech recognition. A compact representation of a DBN model is shown in Figure 5.2, where the single arrows represent directed edges within an individual time slice and the double arrows represent temporal links between time steps. The number 2 beside one double arrow specifies that temporal link jumps 2 time steps. No number attached means that temporal link jumps from slice i to $i + 1$. Number 7 means that there are 7 slices in this DBN model.

Building BN and DBN models can be approached through expert knowledge, data-driven methods, and hybrid methods. Expert knowledge involves domain experts manually specifying the network structure and parameters, leveraging deep insights but being time-consuming and subjective. Data-driven methods, such as constraint-based and score-based algorithms, use statistical analysis to uncover network structures and estimate parameters from large datasets, though they can be computationally intensive and prone to overfitting. Hybrid methods combine the strengths of both approaches, starting with expert-provided structures and refining them using data, offering a balanced solution but still requiring significant expert input.

After constructing the network model and obtaining the findings, users can then proceed to belief updating in BNs or DBNs. Although Equation (5.2) will be sufficient for this

purpose, it requires the joint probability table, and this table can increase exponentially with the number of variables. Thus, more efficient methods need to be investigated. A potential ϕ is defined for a conditional probability table $P(A|pa(A))$ as:

$$\phi : pa(A) \cup \{A\} \rightarrow [0 : 1] \quad (5.4)$$

Indeed, BN belief updating is basically calculating sums of products. A variable X is marginalized out of $\phi(X, Y, \dots, Z)$ by calculating the following summation:

$$\sum_X \phi(X, Y, \dots, Z) = \phi(x_1, Y, \dots, Z) + \dots + \phi(x_k, Y, \dots, Z) \quad (5.5)$$

Instead of using sum notation, projection notation is employed. Hence, $\sum_X \phi(X, Y, \dots, Z)$ in Equation (5.5) can be expressed as $\phi^{\downarrow X}(X, Y, \dots, Z)$, meaning the potential is projected down to X . If $\mathcal{W}$ is a set of variables, then $\phi^{\downarrow \mathcal{W}}$ denotes the result of marginalizing out all variables except variables in $\mathcal{W}$. In the context of BN or DBN belief updating, the task of products summation will be computing $(\prod \Phi)^{\downarrow A_i}$ for all A_i , where Φ is a set of potentials and A_i are variables in a network model. The specific procedure to marginalize out A_i is listed as below:

1. Remove all potential in Φ with A_i in their domain. Call the set of removed potentials Φ_{A_i} .
2. Calculate $\phi^{-A_i} = \sum_{A_i} \prod \Phi_{A_i}$.
3. Add ϕ^{-A_i} to Φ . Call the result Φ^{-A_i} , $\Phi^{-A_i} = \{\Phi \setminus \Phi_{A_i}, \phi^{-A_i}\}$.

The belief updating task is then solved by repeatedly eliminating the variables except for A_i . There are various mechanisms to perform either exact belief updating or approximate belief updating. The mechanism details can be found in [5] if interested.

After network belief updating, users can then estimate the conditional probabilities in the model if they can access a database of cases, which is called parameter estimation. For example, given the DBN model and an observed case, what is the conditional probability for veritable X to be at its first state. This probability will be important to make safe autonomous driving decisions. This can be done through Maximum Likelihood Estimation (MLE). Let $\mathcal{D}$ be a dataset of cases where each case is a configuration over all the variables in $\mathcal{U}$. Then for each case $\mathbf{d} \in \mathcal{D}$ and DBN model M^{DBN} , the likelihood of M^{DBN} given $\mathcal{D}$ is:

$$L(M|\mathcal{D}) = \prod_{\mathbf{d} \in \mathcal{D}} P(\mathbf{d}|M) \quad (5.6)$$

Often log is taken for Equation (5.6) so that it can be written in the form of log-likelihood as:

$$LL(M|\mathcal{D}) = \sum_{\mathbf{d} \in \mathcal{D}} \log_2 P(\mathbf{d}|M) \quad (5.7)$$

Based on the principle of maximum likelihood, if users want to estimate the conditional probability, a parameter estimate $\hat{\boldsymbol{\theta}}_{MLE}$ that maximizes the likelihood should be chosen:

$$\hat{\boldsymbol{\theta}}_{MLE} = \arg \max_{\boldsymbol{\theta}} LL(M_{\boldsymbol{\theta}}|\mathcal{D}) \quad (5.8)$$

where $M_{\boldsymbol{\theta}}$ represents a model with parameter $\boldsymbol{\theta}$.

In addition to MLE, another way to estimate parameters is Maximum a Posteriori (MAP), which aims to find the most probable parameters given the prior data. With the knowledge of likelihood function $P(\mathcal{D}|\boldsymbol{\theta})$ and prior data distribution $P(\boldsymbol{\theta})$, the posterior distribution $P(\boldsymbol{\theta}|\mathcal{D})$ and the optimization problem can be written in the following equations:

$$P(\boldsymbol{\theta}|\mathcal{D}) = \frac{P(\mathcal{D}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathcal{D})} \quad (5.9)$$

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg \max_{\boldsymbol{\theta}} \log P(\mathcal{D}|\boldsymbol{\theta}) + \log P(\boldsymbol{\theta}) \quad (5.10)$$

where $P(\mathcal{D})$ is the marginal likelihood and can be considered as a normalizing constant to ensure the posterior distribution sums to one.

Nevertheless, the previously introduced two parameter estimation methods require a complete dataset in which each case specifies a value for each of the variables. In practice, it is most likely that the dataset is incomplete. For example, some values may not be observable if the variables in a network are latent variables. In this case, one of the most popular solutions is the Expectation-Maximization (EM) algorithm. EM algorithm alternates between a so-called expectation step and a maximization step. In the expectation step, the dataset is “completed” by using the current parameters estimates $\hat{\boldsymbol{\theta}}$ to calculate expectations for the missing values. In the maximization step, the “completed” dataset is utilized to find a new maximum likelihood estimate $\hat{\boldsymbol{\theta}}'$ for the parameters. The algorithm continues this iteration either for a predefined number of iterations or until the algorithm converges. Assume the network model have variables $\mathcal{U} = \{A_1, \dots, A_n\}$, and let θ_{ijk} be the parameter corresponding to the conditional probability $P(A_i = k | pa(A_i) = j)$, where A_i is the conditional probability for variable k being in its j^{th} state given the configuration of its parent variables. Then the maximum likelihood estimate, $\hat{\theta}_{ijk}$, given a dataset $\mathcal{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_m\}$ can be determined based on the EM algorithm below.

Algorithm 6: EM Algorithm [172]

Input : Number of iterations up to the present time (t), maximum number of iterations ($t_{\max}$), convergence criteria (ε , $\varepsilon > 0$), number of variables (n), number of states in variable A_i ($|sp(A_i)|$), where $sp(\cdot)$ is a state number counting function, number of states in parent variables of A_i ($|sp(pa(A_i))|$)

Output : Maximum likelihood estimates of parameter θ_{ijk} ($\hat{\theta}_{ijk}$).

Initialize: Number of iterations: $t = 0$, initialized θ^0 : $\theta^0 = \{\theta_{ijk}\}$ be some initial estimates of the parameters, where
 $1 \leq i \leq n, 1 \leq k \leq |sp(A_i)| - 1, 1 \leq j \leq |sp(pa(A_i))| - 1$.

1-1 **for** $t = 0$ **to** ($t_{\max}$) **do**

1-2 E-step: for each $1 \leq i \leq n$, calculate the table of expected counts:
$$E_{\theta^t}[N(A_i, pa(A_i))|\mathcal{D}] = \sum_{\mathbf{d} \in \mathcal{D}} P(A_i, pa(A_i)|\mathbf{d}, \theta^t)$$

1-3 M-step: Use the expected counts to calculate a new maximum likelihood estimate for all θ_{ijk} based on Equation (5.8). The explicit form of θ_{ijk} can be written as:
$$\hat{\theta}_{ijk} = \frac{E_{\theta^t}[N(A_i=k, pa(A_i)=j)|\mathcal{D}]}{\sum_{h=1}^{|sp(A_i)|} E_{\theta^t}[N(A_i=h, pa(A_i)=j)|\mathcal{D}]}$$

1-4 $\theta^{t+1} = \hat{\theta}$

1-5 $t = t + 1$

1-6 **if** $|\log_2 P(\mathcal{D}|\theta^t) - \log_2 P(\mathcal{D}|\theta^{t-1})| \leq \varepsilon$ **then**

1-7 **break**

1-8 **end if**

1-9 **end for**

5.1.2 Influence Diagram (ID)

Influence Diagram (ID) is an advanced graphic representation of Bayesian networks, aiming to encode the probability model and develop the structure for solving a decision-making problem [5]. One key strength of ID is its ability to handle multiple decision alternatives by allowing an arbitrary number of states in a decision node. It can also include several decision nodes to form a decision sequence. Additionally, IDs manage uncertainties from both observed evidence and the network model itself. As an extension of BNs, IDs are computationally tractable for many practical applications, making them particularly advantageous in the safety design of ADS.

In an influence diagram, the chance variables are denoted as chance nodes. Apart from

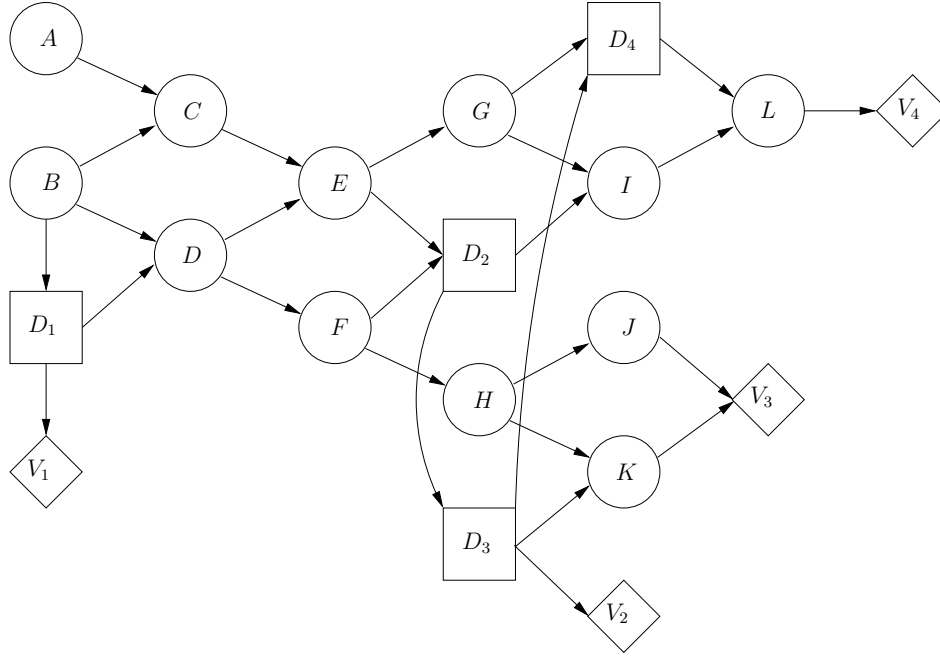


Figure 5.3: A Sample Influence Diagram [5]

the chance nodes, decision nodes and utility nodes are included into the influence diagrams to assist the decision-making tasks. Figure 5.3 illustrates a sample ID. In this example, the chance nodes are in circular shape, squares represent the decision nodes, and diamonds represent the utility nodes. An ID will have the following structural properties:

- There is a directed path comprising all decision nodes.
- The utility nodes have no children.
- The decision nodes and the chance nodes have a finite set of states.
- The utility nodes have no states.

In addition, an ID is realized when the following quantities are specified:

- A conditional probability table $P(A|Pa(A))$ is attached to each chance node A .
- A real-value function over $Pa(\mathcal{U})$ is attached to each utility node $\mathcal{U}$.

If there is more than one utility node, then the entire utility can be the sum or the product of the individual utilities. After an ID is created, it can be solved with different methods. Ultimately, an influence diagram should be solved to achieve the maximum Expected Utility (EU). The final decision alternative will have the highest utility. The EU of each decision alternative can be determined by calculating the parent nodes' posterior probability distributions of the utility nodes. Equation (5.11) below demonstrates the calculation details:

$$EU(X^d|e) = \sum_{x \in Pa(U)} P(x|X^d, e)U(x) \quad (5.11)$$

where X^d represents the selected decision from the decision nodes, x represents all state permutations of $Pa(U)$, e represents the observed evidence about certain chance nodes, and $U(x)$ is the utility value corresponding to the current $Pa(U)$.

More generally, to determine the best decisions and regardless of the existence of evidence, let ID have a set of nodes $U = U_C \cup U_D$ where all chance nodes are represented by U_C and all decision nodes are represented by U_D . In addition, the utility nodes are expressed as $U_V = \{V_i\}$. Let the temporal order of the variables be described as $\mathcal{I}_0 \prec D_1 \prec \mathcal{I}_1 \prec \dots \prec D_n \prec \mathcal{I}_n$ and $V = \sum_i V_i$, where $\mathcal{I}_0$ are the variable information seen before the first decision, D_1 . The similar logic applies $\{\mathcal{I}_1, \dots, \mathcal{I}_n\}$ and $\{D_2, \dots, D_n\}$ in this temporal order. Then the optimal policy for D_i , δ_i , can be written as:

$$\delta_i(\mathcal{I}_0, D_1, \dots, \mathcal{I}_{i-1}, D_i) = \arg \max_{D_i} \sum_{\mathcal{I}_i} \max_{D_{i+1}} \dots \max_{D_n} \sum_{\mathcal{I}_n} P(U_C|U_D)V \quad (5.12)$$

The expected utility from δ_i is:

$$\rho_i(\mathcal{I}_0, D_1, \dots, \mathcal{I}_{i-1}, D_i) = \frac{1}{P(\mathcal{I}_0, \dots, \mathcal{I}_{i-1}|D_1, \dots, D_{i-1})} \max_{D_i} \sum_{\mathcal{I}_i} \max_{D_{i+1}} \dots \max_{D_n} \sum_{\mathcal{I}_n} P(U_C|U_D)V \quad (5.13)$$

Eventually, following this optimal policy for each decision, the Maximum Expected Utility (MEU) can be computed by:

$$MEU = \sum_{\mathcal{I}_1} \max_{D_1} \sum_{\mathcal{I}_2} \max_{D_2} \dots \max_{D_n} \sum_{\mathcal{I}_n} P(U_C|U_D)V \quad (5.14)$$

Following this idea, different optimal policy solving algorithms are developed like variable elimination, node removal, arc reversal and so on. Please refer to [5] for more details.

5.2 Investigate Perception Uncertainty Propagation Effect with DBN

Based on the uncertainty quantification results for Object Position Uncertainty (OPU) from Chapter 4, this section proposes a DBN method to analyze the impact of OPU on object trajectory prediction and how these uncertainties propagate to risk assessment measures used in the decision-making module. This work focuses specifically on the OPU associated with True Positive (TP) detections, as TP cases are the most common during testing. Evaluating the position uncertainty associated with TP detection is crucial for incorporating this uncertainty into the decision-making module. Building on the STPA analysis work from Chapter 3 (see Figures 3.13 and 3.12), a branch of Figure 3.12 is highlighted in Figure 5.4, with the selected UCA propagation traces used as the expert knowledge foundation to construct the causal relationships within the proposed DBN model. Furthermore, this approach demonstrates the proposed methodology’s flexibility, allowing users to select other relevant STPA traces from their qualitative ADS safety analysis and apply the techniques from Chapters 4 and 5 to quantify and incorporate specific learning algorithm uncertainties into their decision-making modules.

As shown in Figure 5.5, the DBN model consists of six nodes: Object Position Uncertainty (OPU), Low Prediction Confidence (LPC), Closest Object Trajectory Confidence (COTC), Threat Measure Credibility (TMC), Ego Velocity (EGOVEL), and Delta s Distance (DS). The nodes OPU, LPC, EGOVEL, and DS are observable and assumed quantifiable. Besides, OPU and LPC can jointly influence COTC results, which is crucial for subsequent TMC evaluation in decision making. EGOVEL and DS, measurable through on-vehicle sensors or algorithms, can help capture the changing characteristics of TMC and aid online DBN inference. However, COTC and TMC are latent variables, which

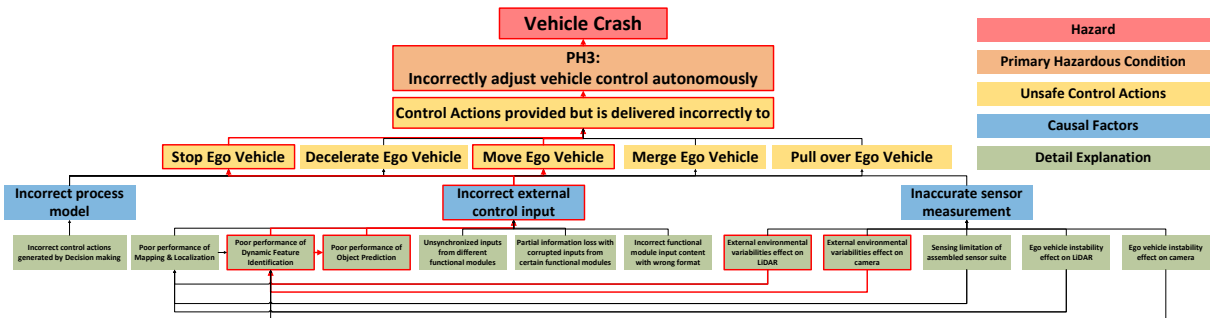


Figure 5.4: Highlighted STPA Traceability Diagram for DBN Formulation

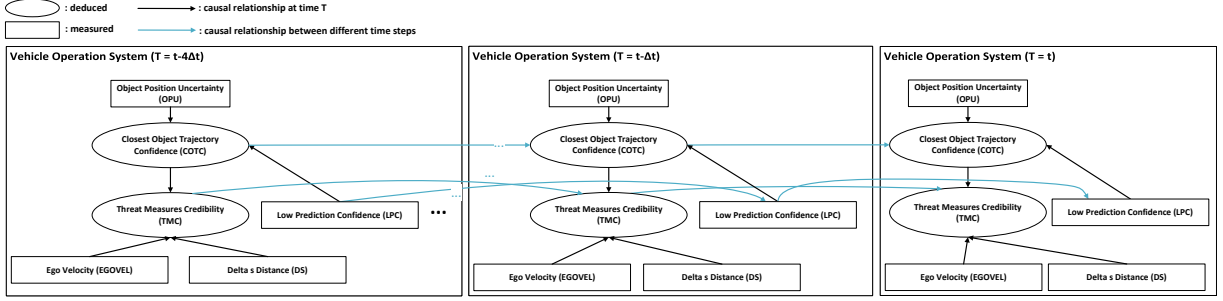


Figure 5.5: Block Diagram of the DBN Model for Assessing Uncertainty Propagation

require offline reference information to calculate actual trajectory confidence and threat measurement accuracy. To update these hidden node probabilities, an EM algorithm is implemented. This DBN model spans five time steps, including the current moment and the past four time steps. In the later simulation and real-world experiments, the time step is set at 0.1s. Nevertheless, the window size is flexible and can be determined experimentally based on different applications. The temporal relationships of COTC and TMC are considered to accurately capture variable changes over time, aiding EM algorithm inference. Although false positive and negative (FP, FN) results from the object detection algorithm are also important, the perception algorithms are designed to have low FP and FN rates. FP and FN cases can also be included in the DBN model in the future with additional nodes. Subsequent paragraphs will provide detailed explanations of each node and the use of the EM algorithm in the DBN model.

Object Position Uncertainty (OPU) represents the distance deviations between perceived and true object positions in 2D space (i.e., longitudinal and lateral directions). Its magnitude is assumed to be proportional to the Euclidean distances from the ego vehicle to the obstacles. This uncertainty is already quantified in the previous chapter and there are also various methods developed to quantify uncertainties in object detection [5, 11, 14, 15]. Based on the data distribution studies in 4.3.1, let the longitudinal and lateral distance between ego vehicle and the object be x and y , respectively. Then the simulated OPU, e_x and e_y , are modelled using normal distribution with distance dependent means, $\mu_x(x)$ and $\mu_y(y)$, and standard deviations σ_x and σ_y in Equation (5.15).

$$f(\mathbf{e}) = \frac{1}{2\pi\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(\mathbf{e} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{e} - \boldsymbol{\mu})\right)$$

$$\mathbf{e} = \begin{pmatrix} e_x \\ e_y \end{pmatrix}, \boldsymbol{\mu} = \begin{pmatrix} \mu_x(x) \\ \mu_y(y) \end{pmatrix}, \Sigma = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix} \quad (5.15)$$

where $\det(\cdot)$ is the determinant calculation function, $\exp(\cdot)$ represents exponential function, T denotes vector transpose action, and ρ is the correlation coefficient. Furthermore, the overall OPU, e , can also be obtained through L2 norm calculation of e_x and e_y . In this DBN model, the overall OPU is discretized into 5 levels with the higher level indicating the greater object position uncertainty.

Low Prediction Confidence (LPC) is expressed as a percentage and indicates whether the confidence score of the predicted object trajectory is low or not. LPC is discretized into 2 levels: $LPC = \text{True}$ indicates confidence $\geq 50\%$; otherwise, $LPC = \text{False}$. The prediction confidence is estimated by sampling the Average Displacement Errors (ADE) from the past data, which are calculated between the predicted and observed object trajectories. The past prediction accuracy can be used as a reference in prediction confidence estimation since the object movement will not change dramatically given the prediction horizons. Furthermore, the ADE samples collected from the past data are based on both short- and long-term prediction horizons to provide a comprehensive assessment of the prediction performance. The ADE for each observed object trajectory prediction can be calculated using Equation (5.16) below:

$$ADE_{obsv,\tau} = \frac{1}{T_{pred,\tau}} \sum_{t=1}^{T_{pred,\tau}} \sqrt{(x_t - x_t^{obsv})^2 + (y_t - y_t^{obsv})^2} \quad (5.16)$$

where x_t and y_t are the object positions along the predicted trajectories. x_t^{obsv} and y_t^{obsv} are the object positions along the observed trajectories. T_{pred} is the prediction horizon. τ stands for different prediction horizons and $\tau \in T = \{pred_{long}, pred_{short}\}$. In addition, during the entire simulation time length (T_{total}), there will be n_{sample} of ADE samples if there are $n_{t,\tau}$ predictions at each time step t .

$$n_{sample} = \sum_{t=1}^{T_{total}} \sum_{\tau}^T n_{t,\tau} \quad (5.17)$$

After collecting the ADE samples, they will then be evaluated according to three histogram analysis criteria. For accurate prediction performance, it is essential for the collected ADE samples to cluster around a mean of zero with minimal variance. Therefore, the three criteria are devised to ascertain that the ADE samples neither exhibit a uniform distribution (C_1) nor multimodal distribution (C_2), and comply with the specified accuracy

threshold (C_3):

$$\underbrace{C_{1,\tau} = KS(\mathcal{D}_{ADE,\tau}, \mathcal{U}_{ADE,\tau})}_{\text{Not uniformly distributed}} \quad \underbrace{C_2 = \frac{ml}{l_{max}}}_{\text{Not a multimodal distribution}} \quad \underbrace{C_{3,\tau} = \max\left(1 - \frac{\mu_{ADE_{obsv},\tau}}{ADE_{thre,\tau}}, 0\right)}_{\text{Predict trajectory accurately}} \quad (5.18)$$

where $\mathcal{D}_{ADE,\tau}$ represents the collected ADE samples and $\mathcal{U}_{ADE,\tau}$ is the corresponding uniform distribution considering the max and min ADE samples in $\mathcal{D}_{ADE,\tau}$. $KS(\cdot)$ represents Kolmogorov-Smirnov test function used to compare how similar $\mathcal{D}_{ADE,\tau}$ is to a uniform distribution. m represents the top m histogram bins and each has a bin width at l . l_{max} is the maximum bin distance among those m bins. $\mu_{ADE_{obsv},\tau}$ represents the mean of ADE samples computed based on the observed object trajectories, and $ADE_{thre,\tau}$ is the minimum accuracy threshold for the collected ADE samples. The visualization of these three evaluation criteria is shown in Figure 5.6.

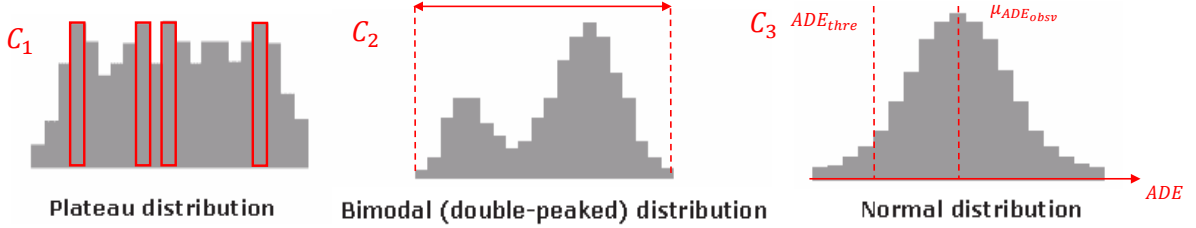


Figure 5.6: Visualization of Three LPC Evaluation Criteria

If the number of collected ADE samples surpasses a threshold ($n_{sample} > n_{sample,thre}$), the ADE samples evaluation through histogram analysis criteria can then be performed as below:

$$\begin{aligned} P_{uni} &= \sum_{\tau}^T w_{1,\tau} C_{1,\tau} \\ P_{multi} &= \sum_{\tau}^T w_{2,\tau} C_{2,\tau} \\ P_{acc} &= \sum_{\tau}^T w_{3,\tau} C_{3,\tau} \end{aligned} \quad (5.19)$$

where $w_{i,\tau}$ are the weights for evaluation results $C_{i,\tau}$ ($i \in \{1, 2, 3\}$), which consider the past predicted trajectories in both short and long terms. P_{uni} , P_{multi} , and P_{acc} are the 3 overall evaluation results based on the three histogram analysis criteria and will be used in LPC

calculation shown as below:

$$\begin{aligned} P(LPC = False) &= \max(\min(P_{uni}, P_{multi}, P_{acc}), 0) \\ P(LPC = True) &= 1 - P(LPC = False) \end{aligned} \quad (5.20)$$

Closest Object Trajectory Confidence (COTC) signifies the estimated confidence score of the predicted trajectories for the close-to-collision objects perceived at the front of ADS based on both object detection and prediction results. COTC is critical for autonomous vehicles as it guides driving maneuvers based on potential collision scenarios. Depending on the traffic, there can be individual or multiple close-to-collision objects to consider. The reference COTC ($COTC_{ref}$) calculated in Equation (5.21) is based on sampling the average displacement errors of the closest object ($ADE_{closest}$), which are computed from predicted and reference trajectories of the closest objects.

$$\begin{aligned} COTC_{ref} &= \min(P_{acc,COTC}, P_{conc,COTC}) \\ P_{acc,COTC} &= \sum_{\tau}^T w_{4,\tau} C_{4,\tau} \quad P_{conc,COTC} = \sum_{\tau}^T w_{5,\tau} C_{5,\tau} \\ C_{4,\tau} &= \max\left(1 - \frac{\mu_{ADE_{closest},\tau}}{\mu_{thres,\tau}}, 0\right) \\ C_{5,\tau} &= \max\left(1 - \frac{\sigma_{ADE_{closest},\tau}}{\sigma_{thres,\tau}}, 0\right) \end{aligned} \quad (5.21)$$

where $C_{4,\tau}$ and $C_{5,\tau}$ are the accuracy and standard deviation criteria for comparing the prediction performance of the closest object's trajectory with all trajectory predictions. $P_{acc,COTC}$ and $P_{conc,COTC}$ are overall evaluation results considering short- and long-term history along with corresponding weights, $w_{4,\tau}$ and $w_{5,\tau}$. $\mu_{ADE_{closest},\tau}$ and $\sigma_{ADE_{closest},\tau}$ represent the mean and standard deviation of ADE samples over short- and long-term history and can be computed as below:

$$\begin{aligned} \mu_{ADE_{closest},\tau} &= \frac{\sum_{i=1}^{T_{pred,\tau}} k_i \frac{1}{i} \sum_{j=0}^i ADE_{closest,t+j}}{\sum_{i=1}^{T_{pred,\tau}} k_i} \\ \sigma_{ADE_{closest},\tau} &= \sqrt{\frac{\sum_{i=1}^{T_{pred,\tau}} k_i \text{var}(\{ADE_{closest,t}, \dots, ADE_{closest,t+i}\})}{\sum_{i=1}^{T_{pred,\tau}} k_i}} \end{aligned} \quad (5.22)$$

where k_i is the weight assigned to the ADE results calculated with different numbers of prediction time steps. Based on different needs, The increment of i can be arbitrary. It does not have to be 1 at each time.

However, since $COTC_{ref}$ relies on the reference trajectories, $COTC_t$ will not be measurable online and will be a latent variable in the DBN model. Therefore, $COTC_t$ will be derived through an EM algorithm during the DBN inference in Equation (5.23) based on OPU_t , LPC_t , and $COTC_{t-1}$.

$$COTC_t = f(OPU_t, LPC_t, COTC_{t-1}) \quad (5.23)$$

where t represents the time step. COTC is discretized into 3 states: Accurate, Warning, and Inaccurate. The higher confidence score means a higher probability to be classified into Accurate state.

Ego velocity (EGOVEL) and Delta s Distance (DS) are the other two measurable nodes in this DBN model. Specifically, EGOVEL represents the current ego vehicle speed and DS is the Frenet frame distance between ego vehicle and the highest risk front obstacle. These two measure nodes are added to help infer the TMC results. The credibility of threat measures like TTC often relies on these ego vehicle related parameters. In addition, without these two nodes, TMC results can only be deduced through another latent variable COTC, which sometimes cannot fully capture the TMC changing characteristics during ADS operation.

Threat Measure Credibility (TMC) represents the confidence in implemented threat measures like Time-to-collision (TTC). For example, TMC evaluates the uncertainties in TTC while considering the existing perception and prediction uncertainties (OPU, LPC), and their impact on COTC. According to Equation (5.24), TMC_{ref} can be estimated by sampling the differences between calculated and reference TTC.

$$TMC_{ref} = \sum_{\tau}^{T_{pred,\tau}} w_{6,\tau} C_{6,\tau}$$

$$C_{6,\tau} = \begin{cases} h(g(V_{ego,t}, COTC_t), TMC_{t-\Delta t(\tau)}), & \text{if } TTC_t < TTC_{thre} \text{ or } d_{s,t} < d_{s,thre} \\ \min(\alpha C_{6,\tau,t-\Delta t(\tau)}, C_{6,max}), & \text{else} \end{cases} \quad (5.24)$$

where $w_{6,\tau}$ is the weight for short- and long-term evaluation results, $C_{6,\tau}$. $V_{ego,t}$, $d_{s,t}$, and TTC_t are ego velocity, delta s distance to the obstacles, and TTC at time t . Besides, TTC_{thre} and $d_{s,thre}$ are pre-defined thresholds for checking TTC_t and $d_{s,t}$. $g(\cdot)$ is a fitted cubic spline extrapolation function based on pre-defined dataset $\mathcal{D}_{TMC,ref}$, where $\mathcal{D}_{TMC,ref}$ is composed of n samples of $\{V_{ego,n}, d_{s,n}, COTC_n, C_{6,n}\}$. The visualization of a sample $g(\cdot)$ can be seen in Figure 5.7. $h(\cdot)$ is a signal smoothing function considering $g(\cdot)$ at time t and $TMC_{t-\Delta t(\tau)}$ estimated at time $t - t(\tau)$. α is a recovering factor and $\alpha > 1$. $C_{6,max}$ is the maximum cap of $C_{6,\tau}$ score. However, since this process relies on TMC_{ref} , TMC is also a

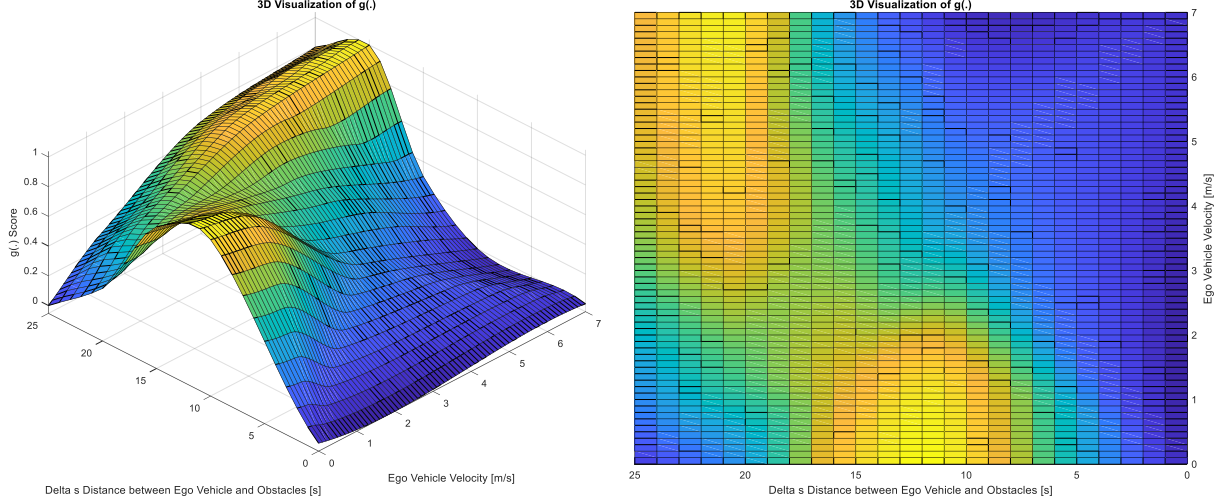


Figure 5.7: Visualization of a Sample $g(\cdot)$

hidden node in the DBN model. Without reference TMC, the EM algorithm used in the DBN inference process relies on $EGOVEL_t$, DS_t , $COTC_t$, and TMC_{t-t} , to infer TMC_t as below:

$$TMC_t = f(EGOVEL_t, DS_t, TMC_{t-t}, COTC_t), \text{ for } TMC_i \text{ where } i \in [t-4, \dots, t] \quad (5.25)$$

where $f(\cdot)$ represents Algorithm 6, which is employed for online DBN inference.

Expectation Maximization (EM) algorithm is a general algorithm for estimating maximum likelihood parameters θ when working with incomplete data. In this work, EM is applied to infer the hidden nodes COTC and TMC in the DBN model. The algorithm consists of two steps: the expectation step, which calculates the missing value expectations using current parameter estimates ($\hat{\theta}$), and the maximization step, which finds new parameter estimates ($\hat{\theta}'$) using the completed data set. The estimate is then used iteratively until convergence. The terms relevant to this EM algorithm are introduced as follows. Variable set, $\mathcal{U}$, is defined as:

$$\mathcal{U} = \{OPU_t, LPC_t, COTC_t, TMC_t\}, t \in [1, \dots, 5] \quad (5.26)$$

where t represents the time step. In addition, the conditional probability parameter is expressed as θ_{ijk} . For instance, a conditional probability parameter can be $\theta = P(TMC_5 = H | TMC_4 = H, COTC_5 = A)$, where H is *High* and A is *Accurate*. Then the maximum likelihood estimation of all θ_{ijk} is expressed as $\hat{\theta}$. According to the MLE method presented

in Section 5.1.1, $\hat{\theta}$ and θ_{ijk} can be calculated as below:

$$\begin{aligned}
\hat{\theta} &= \arg \max_{\theta} L(M_{\theta} | \mathcal{D}) \\
\theta_{ijk} &= P(X_i = k \mid pa(X_i) = j) \\
1 &\leq i \leq n \\
1 &\leq k \leq |sp(X_i)| - 1 \\
1 &\leq j \leq |sp(pa(X_i))| - 1
\end{aligned} \tag{5.27}$$

where X_i is the variable in $\mathcal{U}$. Moreover, for a dataset, $\mathcal{D} = \{\mathbf{d}_1, \dots, \mathbf{d}_s\}$ with s cases. An example case with missing data can be $\{OPU_5 = L1, LPC_5 = False, COTC_5 = d_{miss,1}, TMC_5 = d_{miss,2}, \dots, TMC_1 = H\}$, $L1$ is *Level 1* and d_{miss} represents the missing values in this case. Based on the previous setup, the optimization problem in Equation (5.27) can then be solved by EM Algorithm 6 and both normal and latent variables in the DBN can be estimated for the belief update at time t .

5.3 Formulate Uncertainty-aware Decision Making with ID

In this section, we introduce the proposed Influence Diagram (ID) for handling uncertainty propagation across functional modules and conducting a dual-tiered decision-making process (see Figure 5.8). The chance node variables are the TTC assessment results at the current moment, along with TMC and LPC outcomes from the previous DBN model. The continuous TTC assessment results are discretized into five levels in this ID model. Level 1 means the TTC is equal to or greater than its maximum pre-defined threshold. In contrast, level 5 means the TTC is within the smallest pre-defined interval, representing the most emergent situation. In the model, they are shown as elliptical nodes. Additionally, there are two decision nodes in the ID for deciding high-level driving maneuver and which driving precaution measures are required based on the estimated uncertainty propagation effect. They are shown as rounded rectangles in the model. A utility node in the ID contains a utility function for evaluating the decision alternatives based on the inputs from chance nodes. It is shown as red diamonds in this ID model. The first utility node (U_1) evaluates different diving maneuvers ($D_1 = Stop, Go$) given the current TTC and TMC. The second utility node (U_2) then scores the adoption of various driving precaution measures (D_2) by considering TTC, TMC and LPC.

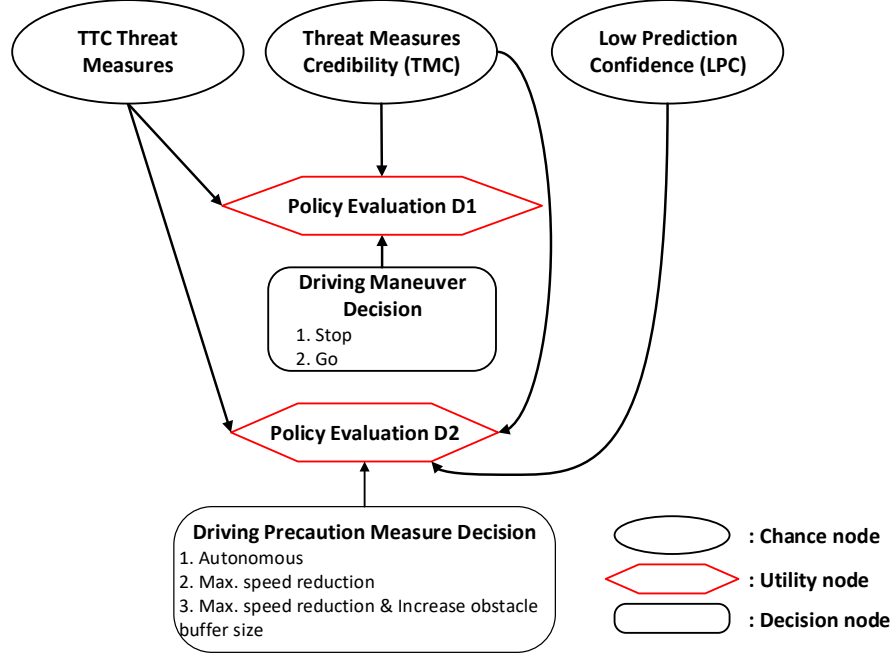


Figure 5.8: Block Diagram of the Proposed ID Model

In this ID model, the high-level driving maneuvers, D_1 , focuses on ego vehicle stop or go, which are the most basic driving commands for operating a vehicle. On top of that, considering learning algorithms' uncertainties and their propagation effect on threat measures, a few driving precaution measures are designed for the second decision node, D_2 . There can be no driving precaution measure (i.e., normal autonomous driving). Nevertheless, given the uncertain scenarios, one measure is limiting the ego vehicle's maximum speed to make the ADS perform more conservatively as well as reduce the severity of any potential collisions. Furthermore, under more uncertain and risky scenarios, D_2 can be increasing the buffer size of the detected objects along with maximum speed reduction to obtain more safety margin for the ego vehicle. This dual-tiered decision-making process can assist in generating decisions on both behavior level for functional requirements and driving precaution level for safety considerations. It aims to improve the existing decision logic by firstly providing a range of driving precaution measures with quantifiable confidence scores under uncertain scenarios, where a rigid safety fallback strategy, like emergency braking, may not be the most appropriate response.

The utility functions of each utility node are initially formulated by cubic spline interpolation with a few pre-defined, desired utility scenarios. Specifically, 4 or more samples

are needed for the interpolation. For U_1 , a sample can be $\{TMC = a, TTC = b, u_1 = c\}$, where $0 \leq a \leq 1, 0 \leq b \leq 1, 0 \leq c \leq 100$ are numeric values of the corresponding chance nodes. Let the pre-defined utility data samples contain data points in the form of (x_i, y_j, z_{ij}) , where there are i samples on x direction and j samples on y direction. Then the bicubic polynomial used to interpolate the surface function, $S(x, y)$, as the utility function can be expressed as:

$$S(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 c_{ij} x^i y^j \quad (5.28)$$

where c_{ij} are the coefficients in the bicubic spline equation. Based on the 4 samples (at least), Equation (5.28) and along with the following equations on the first and second derivatives, c_{ij} can be calculated with 16 equations in total.

$$\begin{aligned} \frac{\partial S(x, y)}{\partial x} &= \sum_{i=1}^3 \sum_{j=0}^3 c_{ij} i x^{i-1} y^j \\ \frac{\partial S(x, y)}{\partial y} &= \sum_{i=0}^3 \sum_{j=1}^3 c_{ij} j x^i y^{j-1} \\ \frac{\partial^2 S(x, y)}{\partial x \partial y} &= \sum_{i=1}^3 \sum_{j=1}^3 c_{ij} i j x^{i-1} y^{j-1} \end{aligned} \quad (5.29)$$

According to the interpolated surface function, the initial U_1 is shown in Figure 5.9. Specifically, Figure 5.9(a) presents the interpolation results for U_1 when the first decision variable, D_1 , is set to "Stop," illustrating utility values across different values of TTC and TMC. Besides, the color gradients indicating utility levels and yellow color means higher utility score. In addition, Figure 5.9(b) depicts an opposite surface plot for D_1 set to "Go," allowing for a direct comparison with the "Stop" scenario. To better visualize the utility function surface plot in this figure, the axes of TTC and TMC are reversed. Figure 5.9(c) then discretize the TTC into 5 levels and display 3 of them for each driving maneuver decision with numerical TMC values on 3 axes. Among these scatter plots, blue color indicates a higher utility score. These scatter plots highlight how U_1 can adapt to varying levels of urgency or collision risk.

On the other hand, since U_2 requires four inputs to determine the utility score for D_2 , there will be one more dimension for its utility scatter plots (See Figure 5.10). This figure demonstrates the initial U_2 function scores with $LPC = True$ for the three driving precaution measures under different TMC and TTC cases. The top, middle, and bottom row represent the decision utility score on autonomous, maximum speed reduction, and maximum speed reduction with obstacle buffer size increase, respectively. From left to right, the increasing TTC levels indicate the decreasing amount of TTC remaining and

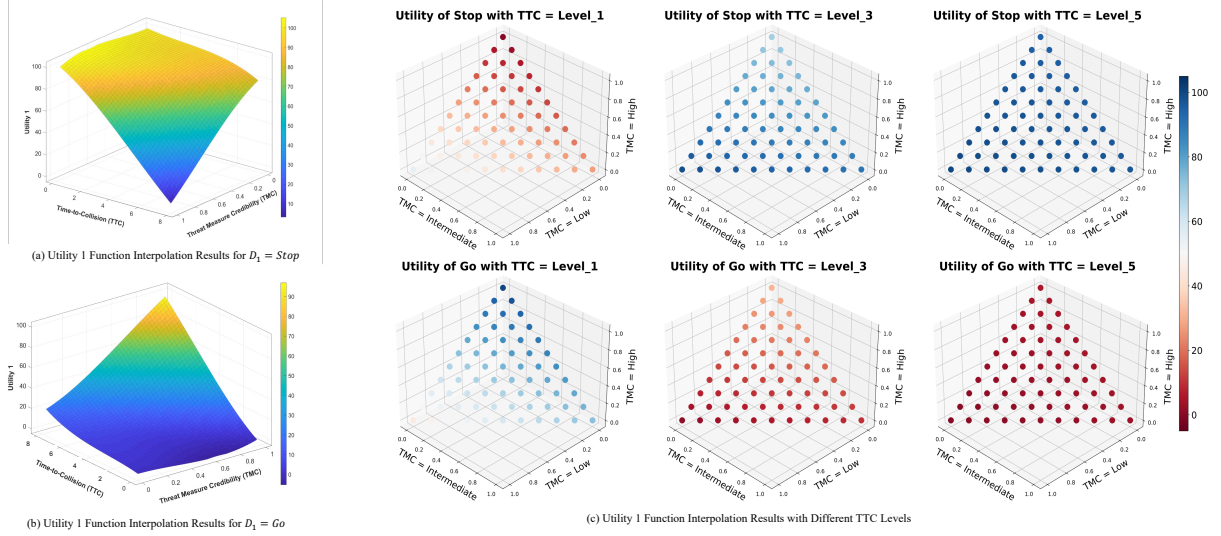


Figure 5.9: Utility Function 1 Visualization. (a) U_1 Interpolation Results for $D_1 = \text{Stop}$, (b) U_1 Interpolation Results for $D_1 = \text{Go}$, (c) U_1 Visualization with Discretized TTC Levels

increasing risk of collision. Same as Figure 5.9. Blue color represents the higher utility in the scatter plots. Thus, maintaining autonomous driving mode is least favorite in the initial U_2 function when the object trajectory prediction confidence score is low (i.e., $LPC = \text{True}$), regardless to TTC or TMC changes. Furthermore, based on TTC levels and TMC differences, the last two driving precaution measures are preferred. Specifically, when TTC at level 1 or 2, maximum ego speed reduction is mostly preferred. When TTC reaches level 4 and 5, maximum speed reduction and obstacle buffer size increase obtain a higher utility score, resulting in even more cautious ego vehicle behaviors with respect to the riskier situation.

After constructing different nodes, the ID model is solved by computing the expected utilities of all possible decision alternatives at each fold of decision process. Eventually, the decision alternatives in a policy with the highest expected utility will be selected. In this work, following the optimal ID decision policy logic in Section 5.1.2, a policy evaluation algorithm is used to solve the proposed ID [173]. Essentially, it transforms the ID into a Bayesian network and then repeatedly performs network inference to find a complete list of expected utilities of all possible decision strategies in the network. To do so, the decision nodes in the ID are transformed into chance nodes. Suppose that there are n decision nodes; the decision order is first determined as $D_{ID} = \{D_1, \dots, D_n\}$. We then transform the utility functions in the ID to probability functions by defining utility node

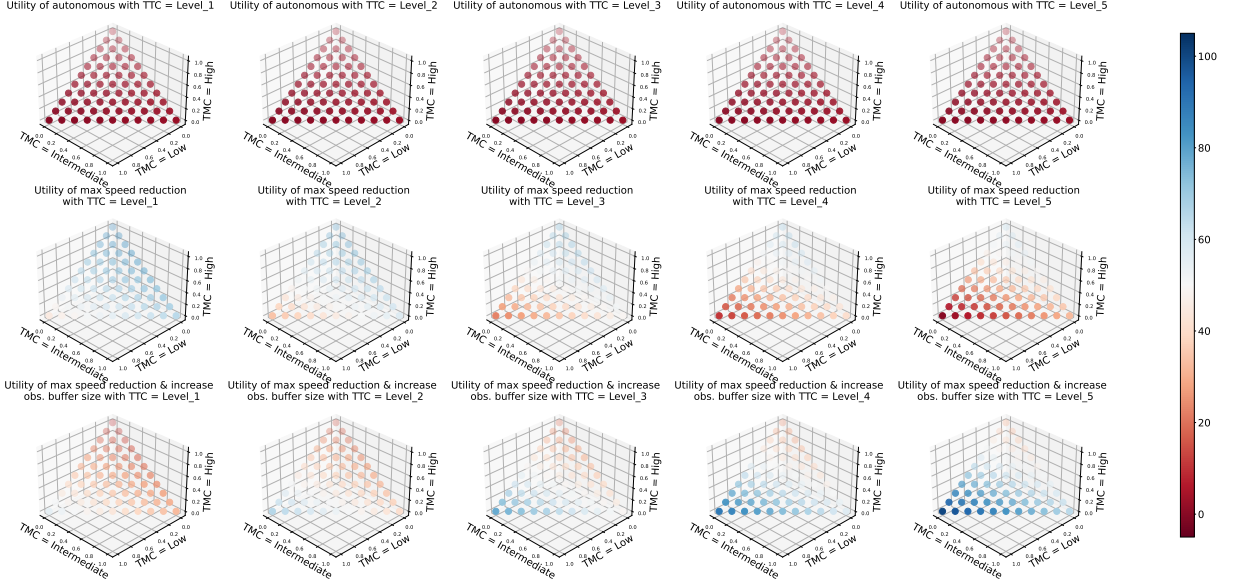


Figure 5.10: Utility Function 2 Visualization with $LPC = True$ and Discretized TTC Levels. (Top Row) U_2 Interpolation Results for $D_2 = Autonomous$, (Middle Row) U_2 Interpolation Results for $D_2 = Max Speed Reduction$, (Bottom Row) U_2 Interpolation Results for $D_2 = Max Speed Reduction \ \& \ Increase Obstacle Buffer Size$

to be binary-valued with possible values T and F . Let Π_U be all nodes in ID with arcs to the utility node U . Then the probability function relating Π_U to U is defined as Equation (5.30), which linearly maps the value of utility node U to a range between 0 and 1.

$$P(U = T | \Pi_U) = \frac{U(\Pi_U) - \min_{\Pi_U} U(\Pi_U)}{\max_{\Pi_U} U(\Pi_U) - \min_{\Pi_U} U(\Pi_U)} \quad (5.30)$$

Based on Section 5.1.2, if there is only one decision node in the ID model, it can be solved by taking the decision which results in the MEV of the corresponding utility node, using Equation (5.31)

$$MEV(D, E) = \max_D \sum_{\Pi'_U} U(\Pi_U) P(\Pi'_U | D, E) \quad (5.31)$$

where D represents the only decision node in the ID and E is the evidence available for the ID model. Π'_U represents all the chance node with arcs to the utility node U . The $U(\Pi_U)$ in Equation (5.31) can then be substitute by Equation (5.30) and Equation (5.31) can be

rewritten as:

$$\begin{aligned}
MEV(D, E) &= k_1 \max_D [P(V = T|D, E)] - k_2 \\
k_1 &= \max_{\Pi_U} U(\Pi_U) - \min_{\Pi_U} U(\Pi_U) \\
k_2 &= -\min_{\Pi_U} U(\Pi_U)
\end{aligned} \tag{5.32}$$

For an ID model with multiple decision nodes, Equation (5.32) needs to be applied for each decision according to the decision order $D_{ID} = \{D_1, \dots, D_n\}$. After a decision is made, that decision is added to the evidence for making the next decision. The general equation for multiple decision nodes can be written as a recursive equation:

$$\begin{aligned}
MEV(D_{ID}, E) &= k_1 f(D_{ID}, E) - k_2 \\
f(D_{ID}, E) &= \max_{h(D_{ID})} \left[\sum_{\Pi'_{h(t(D_{ID}))}} f(t(D_{ID}), E \cup \Pi_{h(t(D_{ID}))}) P(\Pi'_{h(t(D_{ID}))} | h(D_{ID}), E) \right]
\end{aligned} \tag{5.33}$$

where $h(D_{ID})$ returns the first element of D_{ID} , and $t(D_{ID})$ returns the list that remains when the first element of D_{ID} is removed. Dynamic programming can be used to solve this recursive equation. For the proposed ID model in this work, $f(D_{ID}, E)$ can be written as Equation (5.34) explicitly.

$$f(D_{ID}, E) = \max_{D_1} \left[\sum_{\Pi'_{D_2}} f(D_2, E \cup \Pi_{D_2}) P(\Pi'_{D_2} | D_1, E) \right] \tag{5.34}$$

As a result, the output D_1 and D_2 will be the optimal decisions with the highest utilities given the current TTC assessment results, TMC and LPC results from the DBN model.

To accommodate the changing environment and the potential data distribution shift during operation, the initial utility functions of the ID model need to be updated accordingly. Similar to the concept of boosting, the main idea behind this update algorithm is to select a more conservative driving precaution measures for the encountered failure cases and update the initial values of the cubic spline polynomials again with these failure cases. As a result, the utility functions with the updated cubic spline polynomial parameters can dynamically allocate the weights assigned to different decision candidates in each decision node based on the failure history. The detailed steps is shown in Algorithm 7.

Algorithm 7: ID Utility Function Update Algorithm

Input : Testing dataset (D^{test}), ID model (M^{ID}), failure criteria (C_{fail}), decision candidates for D_1 and D_2 ($D_1 = \{d_{1,1}, d_{1,2}\}, D_2 = \{d_{2,1}, d_{2,2}, d_{2,3}\}$), failure number convergence criteria ($\varepsilon, \varepsilon > 0$), original cubic spline interpolation dataset for U_1 and U_2 (D_{U_1}, D_{U_2})

Output : Updated M^{ID} with new utility functions ($M^{ID,new}$).

Initialize: Testing result list: $R = \{\}$, failure case list: $R_{fail} = \{\}$, failure testing cases list: $D^{test,fail} = \{\}$, failure testing case number after updating utility functions in M^{ID} : $n_{fail}^{new} = 0$.

1-1 Test M^{ID} with (D^{test}) and save testing results into R .

1-2 Compare testing results R with failure criteria C_{fail} and identify failure cases.
Save the identified failure cases into R_{fail} and failure test cases into $D^{test,fail}$.

1-3 **for each** r_{fail} **in** R_{fail} **do**

1-4 | Obtain TMC, TTC, LPC, d_1 and d_2 from r_{fail} .

1-5 | Check if d_1 and d_2 can be changed to more conservative decisions.

1-6 | **if** d_1 and d_2 can be more conservative **then**

1-7 | | Change them to more conservative decision candidates and record $\{TMC, TTC, LPC, d_1^{new}, d_2^{new}\}$.

1-8 | **else**

1-9 | | Record original $\{TMC, TTC, LPC, d_1, d_2\}$.

1-10 | **end if**

1-11 | Add $\{TMC, TTC, LPC, d_1^{new}, d_2^{new}\}$ or $\{TMC, TTC, LPC, d_1, d_2\}$ to D_{U_1} and D_{U_2} correspondingly and update U_1 and U_2 with Equation (5.28) and (5.29).

1-12 | Test the ($M^{ID,new}$) with $D^{test,fail}$ and compare with failure criteria C_{fail} .
Count the failure results number, n_{fail}^{new} .

1-13 | **if** $n_{fail}^{new} \leq \varepsilon$ **then**

1-14 | | **break**

1-15 | **end if**

1-16 **end for**

5.4 Simulations and Real-world Experimental Results

The proposed DBNID uncertainty-aware decision-making method is evaluated in both simulation and real-world environments to assess its performance. Section 5.4.1 describes the simulation setup, while Section 5.4.2 presents the simulation results of DBNID with

and without model updates across various testing scenarios, including a performance comparison with an FSM baseline model. Subsequently, Section 5.4.3 details the real-world experiment setup for DBNID on the WATonoBus platform, and Section 5.4.4 evaluates its performance using recorded rosbag data, again compared with an FSM baseline model.

5.4.1 Simulation Setup

Both the proposed DBN model and ID model are initially drafted in the GeNie modeler, chosen for its intuitive graphical interface and robust features. More importantly, GeNie facilitates seamless integration with other programming languages such as Python, JavaScript, etc. In this paper, Python wrapper library was employed for the GeNie models. The wrapped DBN and ID models are then called in a WATonoBus simulation platform built in Simulink (See Fig. 5.11). The details of WATonoBus can be found in Section 3.3.1. Furthermore, the simulation scenarios are created through MATLAB Driving Scenario Designer (DSD) based on the qualitative ADS safety analysis and the combinatorial edge case generation mechanism. During simulation, essential data from the generated simulation driving scenarios, such as perception and prediction results, are exchanged between the Simulink model and the DBNID algorithm. The resulting driving decisions, encompass-

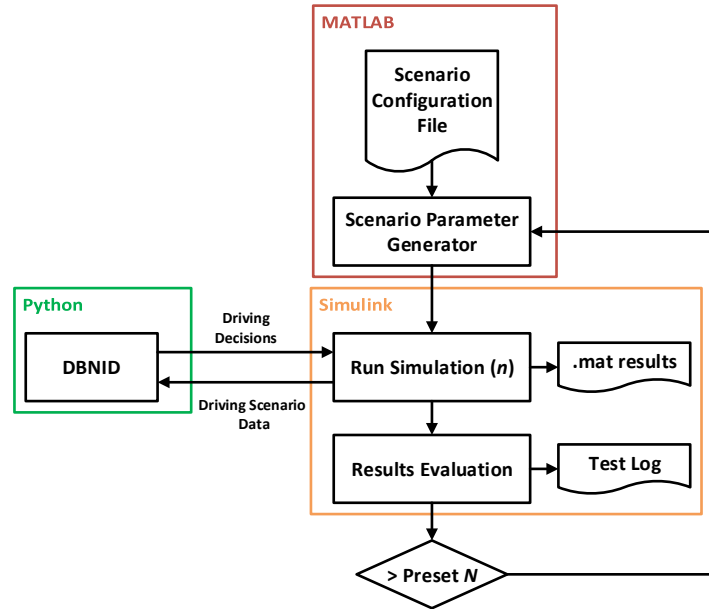


Figure 5.11: DBNID Simulation Data Workflow

ing high-level driving maneuvers and driving precaution measures, are relayed back to the Simulink model for the simulation in the subsequent time step. For each simulated testing scenario, the raw DBNID testing results are saved as .mat files and will then be evaluated against several criteria. The failure cases will be then documented in a test log file for later design revision. Considering the randomness of the continuous parameters each generated testing scenario, one scenario configuration will be evaluated for N times before the next scenario.

To enhance the efficiency of network inference in DBN using the EM algorithm, an initialization dataset is gathered to obtain prior knowledge on pedestrian behaviors. During this process, both $COTC_{ref}$ and TMC_{ref} are accessible. The initialization dataset collection covers four typical scenarios and an additional four edge cases of pedestrian crossings. The detailed initialization scenario descriptions are listed in Table 5.1.

Table 5.1: Simulated Initialization Scenarios for DBN

	Description
1	A single pedestrian crosses the path in front of the vehicle.
2	A pedestrian slows down before crossing the driving lane and then proceeds to cross the lane at a normal speed.
3	Multiple pedestrians cross in front of the vehicle but at different distances. They are equally spaced and cross at close distances to the ego vehicle.
4	Multiple pedestrians cross in front of the vehicle at different distances. They are equally spaced and cross at close distances to the ego vehicle but from different directions.
5	A single pedestrian runs across the path in front of the vehicle.
6	A single jaywalking pedestrian crosses at a far distance from the vehicle.
7	A pedestrian walking on the sidewalk parallel to the vehicle suddenly starts crossing the driving lane.
8	A pedestrian starts to cross the road but then turns back halfway.

Subsequently, the proposed DBNID method is evaluated using 9 testing scenarios, encompassing a total of 99 new edge cases, as listed in Table 3.16. In these simulation scenarios, the true OPU is manually designed as introduced in Section 4.4.1. Given the magnitude of the simulated OPU, it is filtered by a $\pm 0.1\text{m}$ threshold, allowing the subsequent uncertainty propagation and decision-making process to focus more on major object

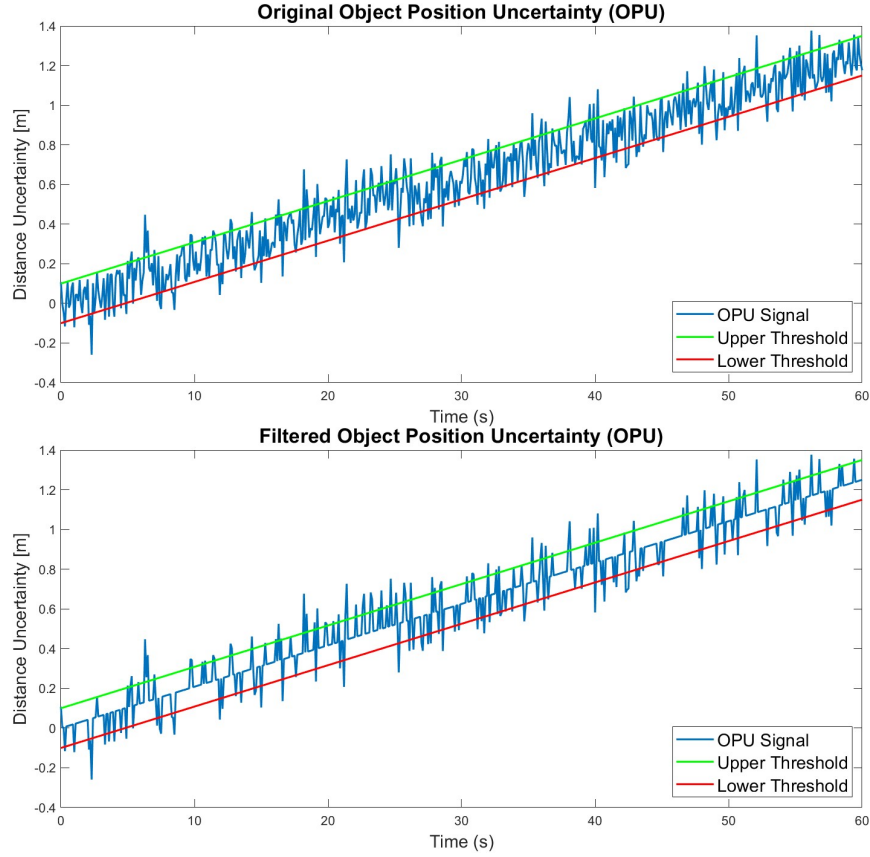


Figure 5.12: (Top) Original OPU and (Bottom) Filtered OPU that are Linearly Proportional to the Distance between Ego Vehicle and Object

position deviations. For visualization purposes, the original and filtered OPU, which have a linear proportional relationship to the distance between the ego vehicle and the object, are plotted in Figure 5.12. In both plots, an object is moving away from the ego vehicle in the simulation, resulting in a linearly increasing OPU over time.

5.4.2 Simulation Results

The safety performance of DBNID is evaluated against a baseline approach with different ego vehicle speeds. The baseline model, an FSM decision making method, focuses solely on high-level driving behavior decisions without considering uncertainties and the associated driving precaution measures. FSM provides a structured and modular framework

for handling various driving scenarios and has been successfully employed in prior studies [110, 112, 113, 114, 174]. Moreover, the selected FSM model has undergone rigorous testing through simulations and on-vehicle experiments in the WATonoBus project. Its decision-making performance has received endorsement as part of the Ministry of Transportation Ontario’s Autonomous Vehicle Pilot Program.

The comparison between the proposed method and the baseline is based on two key evaluation criteria: the number of instances of safe stop violation (N_{STV}) and minimum Time-to-Collision (TTC) between ego and obstacle (TTC_{min}). N_{STV} refers to the count of time steps during which the ego vehicle should either have been already stopped or in close proximity to stopping. If the safety driver is on-board, it represents the time steps during which the driver must closely monitor and be prepared to take over the ego vehicle control. Quantitatively, during the WATonoBus service on campus ring road, a safe stop violation occurs if the distance to the front obstacle is less than 10m and the ego vehicle’s velocity exceeds 5km/h. In testing scenarios, a lower N_{STV} is more desirable, indicating that the ego vehicle does not violate or violates the safe stop criteria for a shorter duration. Table 5.2 lists the N_{STV} values for both DBNID and baseline across different testing scenarios at 20km/h and 30km/h, respectively. Testing scenarios with no performance differences have been omitted from this table.

Table 5.2: N_{STV} Comparison between DBNID and Baseline Methods

EgoVehicle Operates at 20km/h				
Scenario #	Test Scenario	Baseline	DBNID	Improvement after Update
1	Returning pedestrian	28	25	3
4	Single pedestrian crosses the road, behaving unexpectedly.	30	20	10
5	Running to cross, then walking in parallel on sidewalk, and suddenly running to cross	15	4	9
6	Running to cross, waiting, and suddenly crossing the road	54	9	45
8	Multiple pedestrians walk on sidewalk in parallel to ego and then cross the road.	22	12	10

EgoVehicle Operates at 30km/h				
Scenario #	Test Scenario	Baseline	DBNID	Improvement
1	Returning pedestrian	44	34	10
4	Single pedestrian crosses the road, behaving unexpectedly.	24	20	4
5	Running to cross, then walking in parallel on sidewalk, and suddenly running to cross	30	30	0
6	Running to cross, waiting, and suddenly crossing the road	81	1	80
8	Multiple pedestrians walk on sidewalk in parallel to ego and then cross the road.	18	10	8

The proposed method consistently outperforms the baseline, particularly in handling unpredictable and suddenly changing pedestrian road-crossing behaviors in test scenarios 5 and 6, with the ego vehicle operating at 20 km/h. The corresponding N_{STV} reductions are notable, with decreases of 9 and 45 time steps, representing reductions of 73.3% and 83.3%, respectively. Furthermore, DBNID demonstrates superior performance in scenarios involving multiple obstacles. In scenario 8, there is a significant N_{STV} reduction from 22 to 12 time steps, indicating a 45.5% improvement. This performance enhancement holds true even as the ego vehicle speed increases to 30 km/h. Notably, in test scenario 6, the difference is particularly striking, with DBNID achieving an 80-time-step reduction.

TTC_{min} is defined as the minimum TTC between the ego vehicle and the closest front obstacle during the simulation. This calculation considers the safety buffers of both entities. For example, $TTC=0$ can represent either an actual collision between the ego vehicle and the obstacle or an overlap in the safety buffer regions of the two objects. Table 5.3 provides a comparison of TTC_{min} results between DBNID and the baseline at 20km/h and 30km/h.

Table 5.3: TTC_{min} Comparison between DBNID and Baseline Methods

EgoVehicle Operates at 20km/h

Scenario #	Test Scenario	Baseline [s]	DBNID [s]	Gain [s]
1	Running pedestrian	2.31	2.43	+0.12
2	Walking on sidewalk in parallel to ego and then crossing	2.89	3.04	+0.15
3	Returning pedestrian	1.93	1.92	-0.01
4	Single pedestrian crosses the road, behaving unexpectedly.	0.84	1.82	+0.98
5	Running to cross, then walking in parallel on sidewalk, and suddenly running to cross	2.66	2.77	+0.11
6	Running to cross, waiting, and suddenly crossing the road	0.1	1.33	+1.32
7	Multiple pedestrians cross the road. One of them is behaving unexpectedly.	1.49	1.83	+0.34
8	Multiple pedestrians walk on sidewalk in parallel to ego and then cross the road.	1.08	1.05	-0.03
9	1 running pedestrian with multiple normal crossing pedestrians.	2.28	2.65	+0.37
EgoVehicle Operates at 30km/h				
Scenario #	Test Scenario	Baseline [s]	DBNID [s]	Gain [s]
1	Running pedestrian	1.91	2.01	+0.1
2	Walking on sidewalk in parallel to ego and then crossing	1.80	1.97	+0.17
3	Returning pedestrian	2.10	2.48	+0.38
4	Single pedestrian crosses the road, behaving unexpectedly.	1.34	1.48	+0.14

5	Running to cross, then walking in parallel on sidewalk, and suddenly running to cross	0.98	1.23	+0.25
6	Running to cross, waiting, and suddenly crossing the road	2.36	2.84	+0.48
7	Multiple pedestrians cross the road. One of them is behaving unexpectedly.	1.17	0.76	-0.41
8	Multiple pedestrians walk on sidewalk in parallel to ego and then cross the road.	0.87	1.84	+0.97
9	1 running pedestrian with multiple normal crossing pedestrians.	0.5	1.2	+0.70

The TTC_{min} comparison results reveal an overall performance improvement with DBNID outperforming the baseline. In specific testing scenarios, DBNID demonstrates noticeable enhancement. For instance, when the ego vehicle is driving at 20 km/h, the improvements in testing scenarios 4 and 6 are 0.98s and 1.32s, corresponding to improvements of 116.7% and 123%, respectively. Given that the baseline’s performance in these scenarios is 0.84 seconds and 0.1 seconds, the gains from DBNID are substantial in preventing collisions. Furthermore, in scenarios involving multiple obstacles, such as scenarios 7 and 9, the TTC_{min} improvements with DBNID are also significant, reaching 22.8% and 16.2%, respectively. Although the baseline shows slightly longer times in scenarios 3 and 8, with differences of only 0.01 to 0.03 seconds, this suggests a comparable performance between DBNID and the baseline. These marginal discrepancies may arise from uncertainty fluctuations during testing but have minimal impact on the vehicle’s stopping reaction. Moreover, when the ego vehicle drives faster at 30 km/h, notable improvements from DBNID are also observed in testing scenarios 8 and 9, with improvements of 0.97s and 0.7s, respectively.

Furthermore, this paragraph presents scenario 4 in detail to illustrate the performance disparities over the entire simulation between the proposed DBNID method and the baseline from various angles. In this scenario, a pedestrian crosses the road unexpectedly and then reverses direction after crossing (refer to Figure 5.13). The blue line illustrates the pedestrian’s true trajectory, with red segments indicating areas of heightened OPU. Figure 5.14 depicts the resulting longitudinal velocity of the ego vehicle, its distance to the nearest

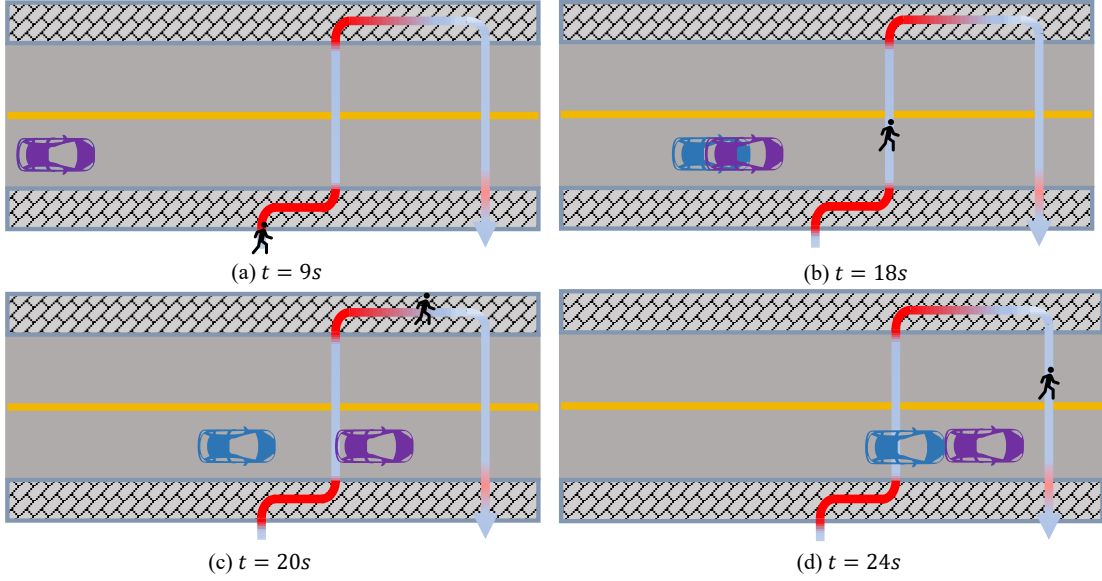


Figure 5.13: Unexpected pedestrian road crossing and turning back case. The blue vehicle denotes the DBNID ego vehicle and the purple vehicle denotes the baseline ego vehicle. The true pedestrian crossing trajectory is shown with red segments representing high OPU periods. Sidewalk has been filled with diagonal brick pattern.

front obstacle, and decision scores. Both DBNID and baseline methods output D_1 for the high-level behavior decisions and the score for stopping the vehicle is plotted. However, the DBNID method additionally offers insights for selecting driving precaution measures, D_2 . As depicted in the plot, the ego vehicle proceeds normally until 14 seconds, detecting the approaching pedestrian, which triggers a rapid increase in the stopping maneuver score. Both methods effectively decelerate as the pedestrian walks alongside and then crosses the road for the first time. Nevertheless, given the uncertain object position detections and the unexpected crossing behavior, limiting ego vehicle maximum speed in D_2 accrues a higher score while the ego vehicle is decelerating. This precautionary measure restricts the DBNID method's longitudinal velocity to 10 km/h from 18 to 21 seconds, whereas the baseline-guided ego vehicle begins accelerating to 20 km/h. Consequently, when the pedestrian initiates the second crossing, the DBNID method allows sufficient reaction time for the ego vehicle to slow down by capping its maximum speed. In contrast, the baseline method can only transition the stopping maneuver score to 100 from 0 to initiate the stopping maneuver, resulting in 5 time steps of safe stop violation. Furthermore, after handling the unexpected pedestrian crossing, the DBNID method can resume normal autonomous driving (i.e., default setting) in D_2 without being overly conservative for the remainder of

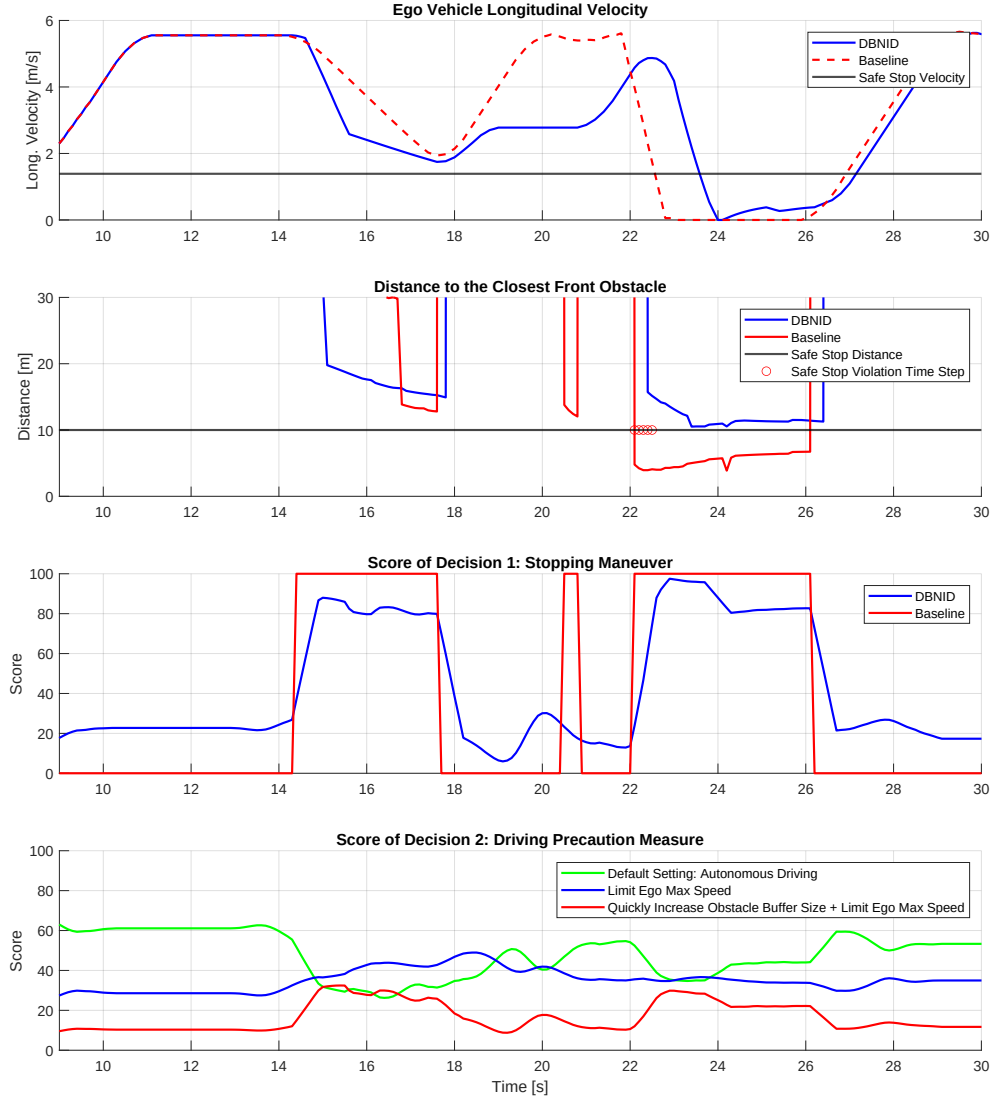


Figure 5.14: Result Comparison between Proposed DBNID Method and Baseline Method

the simulation.

Although DBNID achieves better performance with the previous two evaluation criteria, it can still violate the safety stop criteria in a few testing scenarios (See Table 5.2). This reflects the data discrepancies between the initialization and testing processes. Therefore, the utility functions in the ID model need to be updated according to the previous failure

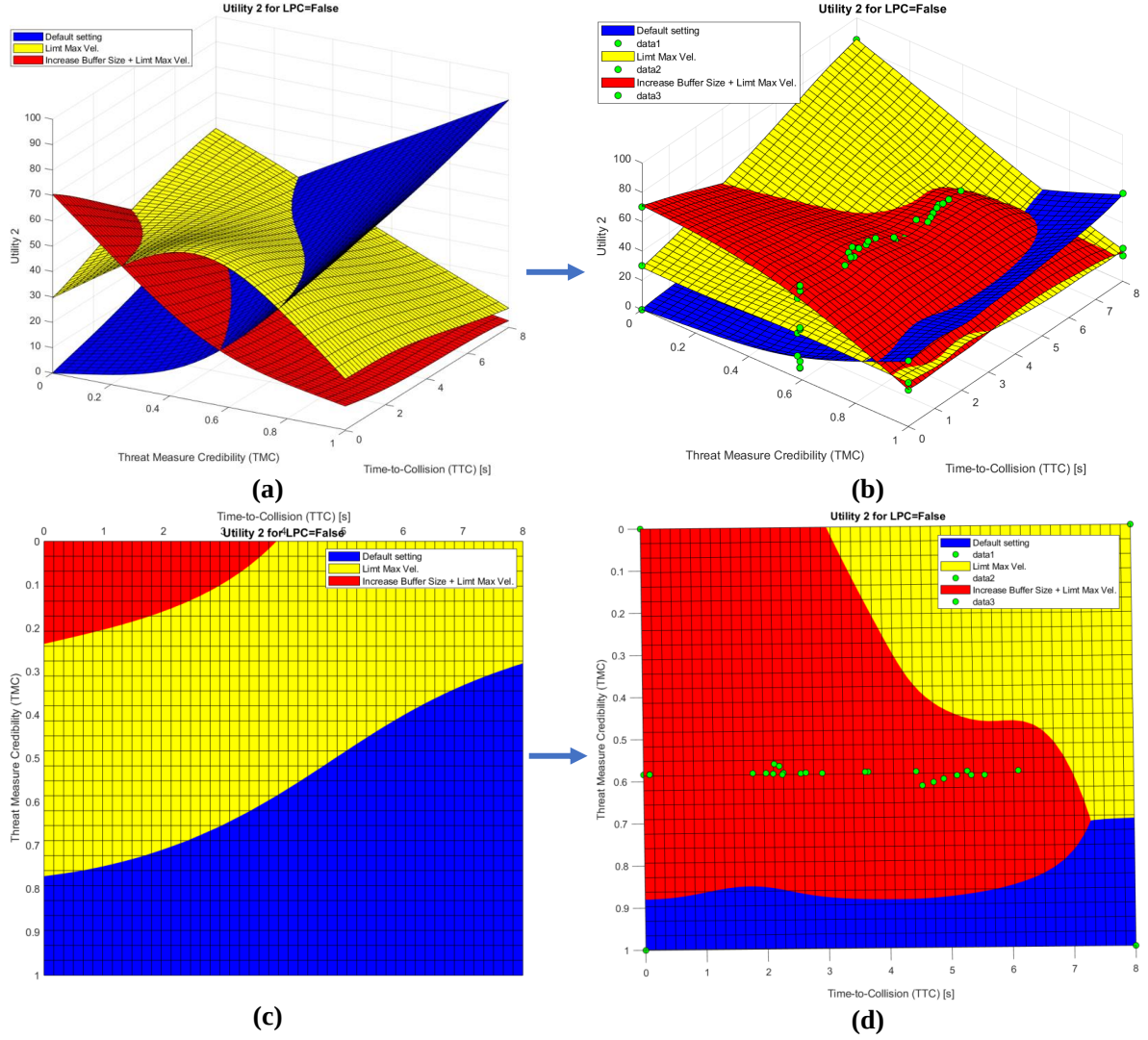


Figure 5.15: (a) 3D View of Original U_2 (b) 3D View of Updated U_2 (c) Top View of Original U_2 (d) Top View of Updated U_2 , when $LPC = False$

cases identified in the testing process. In this case, after reviewing the raw testing results with D_1 and D_2 , U_2 is identified that needs to be updated since D_1 is already *Stop* at those safety stop violation time steps, but D_2 still remains at *Autonomous* mode, ignoring the uncertainties and the possibly approaching objects. Specifically, the failure cases are found with $LPC = False$. Hence, the collected failure cases are used to update U_2 when $LPC = False$. Originally, the utility surface functions for the three decision candidates

in D_2 are plotted in Figure 5.15 (a) and (c) from 3D and top view perspectives. After identifying the failure cases encountered during the previous testing process (shown as the green dots in Figure 5.15 (b) and (d)), U_2 is updated correspondingly. Compared the utility function top views before and after update, it is obvious that the overall D_2 logic becomes more conservative since the *Maximum speed reduction & Increase obstacle buffer size* has taken the largest area in U_2 .

The updated DBNID is then examined with the testing scenarios listed in Table 5.2 at 20km/h to evaluate the effectiveness of the update. Table 5.4 presents a comparative analysis of N_{STV} and TTC_{min} before and after DBNID update, revealing substantial improvements with the implemented update. For N_{STV} , the improvement is obvious. For 4 out of 5 failure scenarios, the utility function update has helped solve all safety stop violation cases. The only remaining scenario is scenario 1, which is too harsh for the ego vehicle to react on time. Therefore, in this case, it is suggested to directly implement related safety fallback strategy like emergency stop. Furthermore, scenario 4, characterized by a single pedestrian unexpectedly crossing the road, saw a notable increase of TTC_{min} from 1.82 seconds to 5.08 seconds, demonstrating an improvement of 3.26 seconds. Similarly, in Scenarios 1, 5 and 6, the improvements are all around 1.6 to 1.7 seconds. These results collectively demonstrate the effectiveness of the updates in enhancing the safety performance for DBNID when encountering unfamiliar testing data during operation.

Table 5.4: N_{STV} and TTC_{min} Comparison Before and After DBNID update

N_{STV} Results Comparison				
Scenario #	Test Scenario	Before Update	After Update	Improvement after Update
1	Returning pedestrian	25	24	1
4	Single pedestrian crosses the road, behaving unexpectedly.	20	0	20
5	Running to cross, then walking in parallel on sidewalk, and suddenly running to cross	4	0	4
6	Running to cross, waiting, and suddenly crossing the road	9	0	9

8	Multiple pedestrians walk on sidewalk in parallel to ego and then cross the road.	12	0	12
TTC_{min} Results Comparison				
Scenario #	Test Scenario	Before Update	After Update	Gain
1	Returning pedestrian	1.75	3.37	+1.62
4	Single pedestrian crosses the road, behaving unexpectedly.	1.82	5.08	+3.26
5	Running to cross, then walking in parallel on sidewalk, and suddenly running to cross	2.63	4.29	+1.66
6	Running to cross, waiting, and suddenly crossing the road	1.33	3.09	+1.76
8	Multiple pedestrians walk on sidewalk in parallel to ego and then cross the road.	1.07	1.71	+0.06

In addition to assessing the safety performance of DBNID, it is crucial to evaluate the estimation accuracies of $COTC_t$ and TMC_t to validate the network inference capability of the DBN model concerning these two hidden nodes. These evaluations gauge how effectively the proposed DBN model propagates the influence of perception and prediction uncertainties (OPU and LPC) to the decision-making level. The estimation accuracies, outlined in Table 5.5, affirm the precise reflection of uncertainty influence propagation onto the two hidden nodes. During validation, COTC achieves an estimation accuracy of 90.49%, maintaining this level during testing, while TMC demonstrates an accuracy of 96.02% at validation, which slightly decreases by 5.29% during testing due to unseen scenarios and limited prior knowledge for network inference. The variance in estimation accuracy between COTC and TMC can be attributed to the two consecutive hidden nodes in the DBN model (Refer to Figure. 5.5) and the potential error propagation from the upper hidden node to the lower one. Overall, the results demonstrate promising estimation accuracies of the two hidden nodes in the proposed DBN model. Future endeavors will focus on minimizing error propagation and further enhancing the estimation accuracy of these hidden nodes.

Table 5.5: COTC and TMC Estimation Accuracy at Initialization and Testing Stages

	COTC Estimation Accuracy	TMC Estimation Accuracy
Validation	90.49%	96.02%
Testing	90.60%	90.73%

5.4.3 Real-world Experimental Setup

DBNID real-world experiment utilizes the same rosbag data as the one in Chapter 4. It is obtained from the WATonoBus platform. The initialization scenarios and the first set of testing scenarios are collected under sunny weather at 10Hz and 1Hz respectively (see Figure 4.13a, b). The second set of testing scenarios are collected under snowy weather at 10Hz (see Figure 4.13c). The perception and prediction results in these rosbag file are inputs to the DBNID model, along with the real-world OPU uncertainty quantification results obtained through PCE in Chapter 4 (see Figure 5.16). At this stage, DBNID required inputs are extracted from rosbag files and PCE uncertainty quantification work and saved as .xlsx files. However, integrating PCE uncertainty quantification and DBNID uncertainty-aware decision-making algorithm onto WATonoBus can be easily implemented and is scheduled as a future task.

Like DBNID simulation, the initialization and testing scenarios are extracted from

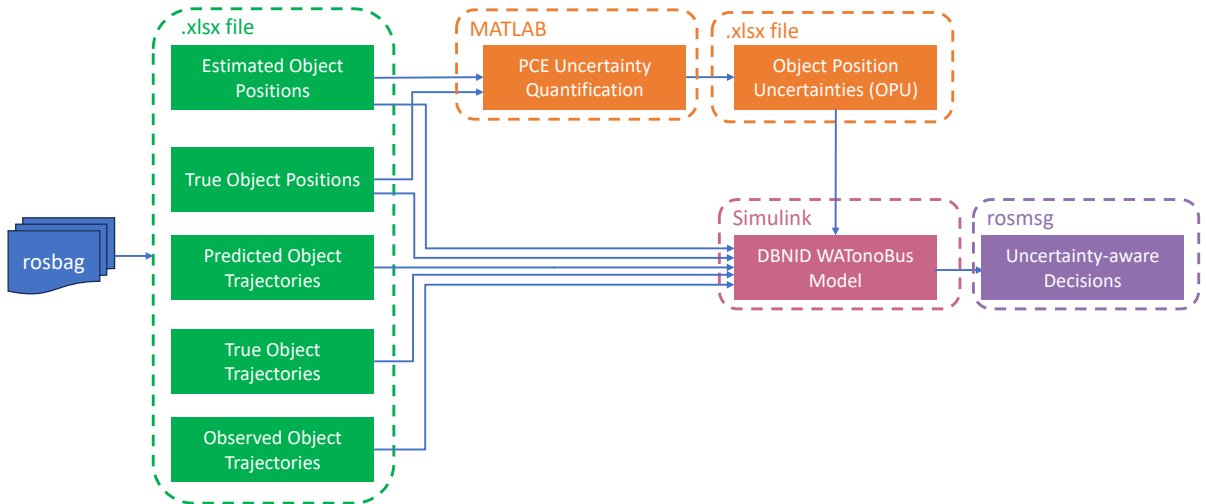


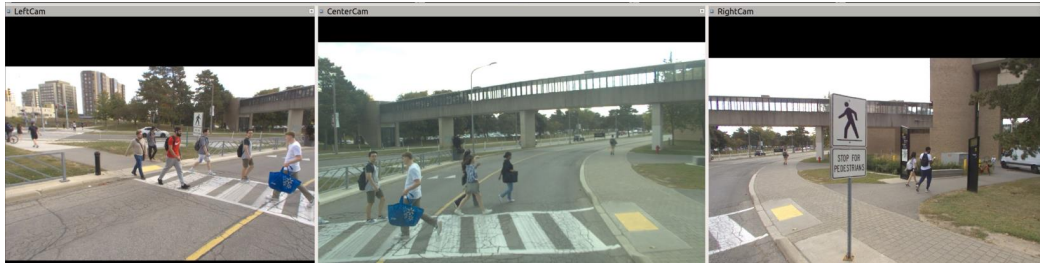
Figure 5.16: Real-world Dataflow of DBNID

the obtained rosbag files. Specifically, 10 initialization scenarios, 7 testing scenarios under sunny weather, and 6 testing scenarios under snowy weather are collected along the campus ring road. Their detailed descriptions are listed in Table 5.6 below and the camera views of a few scenarios are shown in Figure 5.17.

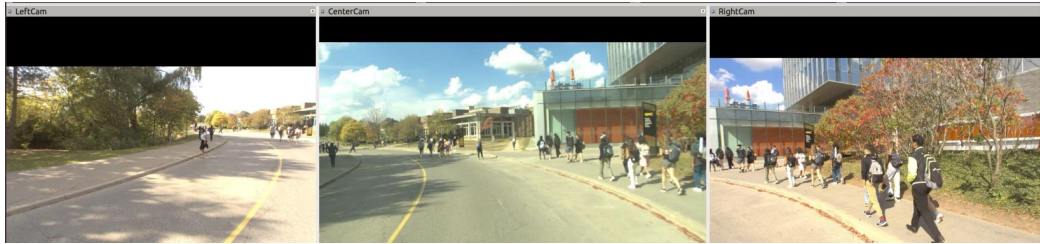
Table 5.6: Simulated Initialization Scenarios for DBN

	Initialization Scenario Description
1	Encounter crossing and sidewalk pedestrians after the E3 intersection.
2	Encounter a sidewalk pedestrian after the E3 intersection.
3	Encounter pedestrians at the crosswalk.
4	Encounter a sidewalk pedestrian after the crosswalk.
5	Stop and wait at the Seagram intersection.
6	Pass through the other end of the Seagram intersection.
7	Encounter a sidewalk pedestrian at the Seagram bus stop.
8	Encounter a jaywalking pedestrian who then returns at the BC Matthews bus stop.
9	Encounter a returning pedestrian at the Seagram bus stop.
10	Encounter a sidewalk pedestrian before the E3 intersection.
	Sunny Testing Scenario Description
1	Encounter a sidewalk pedestrian after the EV3 intersection.
2	Encounter a sidewalk pedestrian at the Ira Needles Hall bus stop region (Case 1).
3	Encounter a sidewalk pedestrian at the Ira Needles Hall bus stop region (Case 2).
4	Encounter a crossing pedestrian at the Ira Needles Hall bus intersection.
5	Encounter numerous crossing and sidewalk pedestrians after the Ira Needles Hall bus intersection.
6	Encounter a crossing pedestrian at the BC Matthews Hall intersection.
7	Encounter a crossing pedestrian at the DC train station intersection.
	Snowy Testing Scenario Description

1	Encounter a sidewalk pedestrian after the Seagram bus stop region.
2	Encounter a sidewalk pedestrian at the Ira Needles Hall intersection.
3	Encounter a sidewalk pedestrian at the opposite end of the Ira Needles Hall intersection.
4	Encounter a crossing pedestrian after the Ira Needles Hall intersection.
5	Encounter a sidewalk pedestrian after passing the Ira Needles Hall intersection.
6	Encounter a crossing pedestrian at the DC train station intersection.



a) Init. Scenario 3 - Encounter pedestrians at the crosswalk.



b) Sunny Test Scenario 5 - Encounter numerous crossing and sidewalk pedestrians after the Ira Needles Hall bus intersection.



c) Snowy Test Scenario 4 - Encounter a crossing pedestrian after the Ira Needles Hall intersection.

Figure 5.17: Real-world Dataflow of DBNID

5.4.4 Real-world Experimental Results

The real-world experiment performance of DBNID is first evaluated with N_{STV} and TTC_{min} for the two sets of testing scenarios. Considering the traffic condition along campus ring road is usually not very busy, there are not many situations requiring performing safety stop for the pedestrians. Therefore, for the safety stop violation criteria, only scenarios with $N_{STV} \neq 0$ are tabulated in Table 5.7. In addition, the second half of Table 5.7 lists the scenarios and their TTC_{min} results that are not exceed the maximum TTC_{min} threshold. In this work, the maximum TTC_{min} threshold is set at 30s. If $TTC_{min} > 30$, it means the ego vehicle does not need to decelerate during the entire scenario.

Table 5.7: N_{STV} and TTC_{min} Testing Results for DBNID Real-world Experiments

N_{STV} Results Comparison			
Testing Scenario Description	Baseline	DBNID	Improve- ment after Update
Sunny Test Scenario 5 - Encounter numerous crossing and sidewalk pedestrians after the Ira Needles Hall bus intersection.	11	1	10
TTC_{min} Results Comparison			
Test Scenario Description	Baseline [s]	DBNID [s]	Gain [s]
Sunny Test Scenario 2 - Encounter a sidewalk pedestrian at the Ira Needles Hall bus stop region (Case 1).	4.84s	30s	+25.16s
Sunny Test Scenario 3 - Encounter a sidewalk pedestrian at the Ira Needles Hall bus stop region (Case 2).	1.36s	1.37s	+0.01s
Sunny Test Scenario 4 - Encounter a crossing pedestrian at the Ira Needles Hall bus intersection.	2.56s	2.89s	+0.33s
Sunny Test Scenario 5 - Encounter numerous crossing and sidewalk pedestrians after the Ira Needles Hall bus intersection.	1.33s	1.59s	+0.25s

Sunny Test Scenario 6 - Encounter a crossing pedestrian at the BC Matthews Hall intersection.	4.94s	5.65s	+0.70s
Snowy Test Scenario 2 - Encounter side-walk ped at Ira Needles Hall intersection	0.1s	0.29s	+0.19s
Snowy Test Scenario 6 - Encounter crossing ped at DC train station intersection	4.59s	9.71s	+5.13s

Based on Table 5.7, the ego vehicle does not encounter safety stop violation issue in most of testing scenarios since the traffic condition is not busy and the vehicle speed was at around 20km/h. Nevertheless, for the only testing scenario that $N_{STV} \neq 0$, DBNID has reduced the safety stop violation time steps from 11 times to 1 time, compared to the baseline model. In addition, significant improvements are observed from TTC_{min} results. Specifically, for sunny test scenario 2, DBNID totally avoids the need for stopping ego vehicle (i.e., $TTC_{min} > 30s$) while the baseline still needs to stop and has a TTC_{min} at 4.48s. Another notable improvement can be found in snowy test scenario 6. The TTC_{min} of DBNID is 5.13s longer than that of the baseline model, which reduces the collision risk dramatically. Overall, among all testing scenarios, DBNID exhibits consistently better TTC_{min} performance than the baseline model.

In addition to N_{STV} and TTC_{min} , the estimation accuracy of $COTC_t$ and TMC_t of DBN with real-world rosbag data are also computed to examine whether these two hidden node can be deduced correctly, which set a foundation for the uncertainty-aware decision making in the ID model. Table 5.8 highlights the major findings regarding the estimation accuracy of COTC and TMC across different conditions.

Table 5.8: Real-world Experiment COTC and TMC Estimation Accuracy

	COTC Estimation Accuracy	TMC Estimation Accuracy
Validation	79.3%	97.6%
Sunny Testing Scenarios	74.9%	88.4%
Snowy Testing Scenarios	94.1%	100%

During the validation phase, the estimation accuracies of COTC and TMC are achieved at 79.3% and 97.6%, respectively. In sunny testing scenarios, the accuracy of COTC slightly decreases to 74.9%, while TMC shows a more pronounced drop to 88.4%. Conversely, in snowy testing scenarios, both hidden nodes exhibit higher estimation accuracies, with COTC reaching an estimation accuracy of 94.1% and TMC achieving 100%. These results suggest that while both models perform well under varying conditions, the estimation accuracy of TMC consistently outperforms COTC, and both hidden nodes exhibit enhanced accuracy in snowy conditions compared to sunny scenarios.

Finally, through runtime examination, it is confirmed that running DBNID with and without ID updates are both time-efficient, meeting the real-time operation requirement of 10-20 Hz in a MATLAB environment with 16GB RAM and an AMD Ryzen 9 5900X 12-Core processor. The runtime performance could be further improved by compiling the code with C/C++.

5.5 Result Analysis and Discussion

One reason to choose the proposed DBNID method is the capability handling uncertainties from different sources. In this study, the scope of uncertainties is confined to the perception and prediction results for pedestrians and their motion trajectories. The objective of the DBN's uncertainty propagation assessment is to gauge how these uncertainties impact the reliability of the selected threat measures such as TTC. While the influence of perception uncertainties can sometimes be directly integrated into TTC calculation (i.e., $\pm xm$ from OPU results in $\pm ts$ in TTC), quantifying the impact of prediction uncertainties on TTC is intricate. This complexity arises from the intertwined relationship with perception uncertainties, making it challenging to establish a precise mathematical formulation. Consequently, there is a lack of complete understanding about prediction uncertainties' influence on TTC. This confluence of uncertainty influence is a common occurrence in autonomous software systems. Hence, the DBN model incorporates both perception and prediction uncertainties (intertwined with perception uncertainties) to analyze their propagation impact on $COTC_t$ and eventually TMC_t . Its aim is to determine the probability:

$$P(TMC, COTC | OPU, LPC) \quad (5.35)$$

instead of formulating the equation:

$$TMC = f(OPU, LPC) = f(OPU, g(OPU)) \quad (5.36)$$

where $LPC = g(OPU)$ and $g(\cdot)$ is a black box function.

Both the simulation and real-world experiment results affirm that the proposed DBNID method yields more dependable decisions, enhancing the overall safety of the ego vehicle. The improvements in N_{STV} and TTC_{min} significantly reduce the likelihood of collisions caused by uncertain perception or prediction results. This improvement is attributed to the continuous adaptation of high-level behavior decision (D_1) utility driven by the DBN uncertainty propagation assessment results. Additionally, the incorporation of the driving precaution measure decision (D_2) with its quantitative utilities further contributes to these enhancements. Notable improvement is evident in the N_{STV} comparison for simulation testing scenario 6 when the ego vehicle is driving at 20 km/h. The associated TTC_{min} of this scenario also reflects the safety improvement of DBNID. For simulation testing scenario 6, the baseline has $TTC_{min} = 0.1s$, which can be seen as a collision, while DBNID still achieves a 1.33s TTC_{min} result. Furthermore, during real-world experiments, significant improvements can also be observed from sunny test scenario 2 and snowy test scenario 6. Although the TTC_{min} results of both DBNID and the baseline in these two scenarios are not close to a collision, the improvement brought by DBNID relieves the ego vehicle from performing a stopping maneuver while deceleration is still required according to the baseline TTC_{min} results.

To better understand the performance improvement brought by DBNID, simulation testing scenario 4 is selected as a case study for in-depth analysis. As demonstrated, the continuous D_1 from the DBNID method enables the ego vehicle to avoid harsh acceleration and braking, addressing issues related to fluctuating maneuver decisions. Furthermore, D_2 helps the ego vehicle adopt a precautionary driving style by accounting for uncertainties, allowing it to navigate challenging scenarios more adeptly when facing uncertain object positions or unexpected object movements. Notably, DBNID does not merely opt for the most conservative decisions to enhance safety performance. According to Figure 5.14, after approximately 19 seconds, the decision-making status of D_2 resumes to *Autonomous*, indicating that DBNID assesses the current uncertainties as low and no longer impacting ADS driving safety. Consequently, an increase in velocity is observed in the longitudinal velocity plot. After the crossing pedestrian exits the region of interest at the end of this scenario, a significant velocity increase is also observed with $D_2 = \textit{Autonomous}$, demonstrating DBNID’s ability to balance safety with efficient driving dynamics.

To accommodate new data encountered during the testing phase, the utility functions in the ID model are constantly updated based on identified past failure cases. Specifically, as demonstrated in Figures 5.15b and 5.15d, the past failure cases are labeled as green dots and plotted alongside the utility functions of three D_2 decision candidates. The plot indicates that the original utility functions only allowed the ego vehicle to choose *Autonomous* or *Max speed reduction*, which were insufficient for safely stopping the ve-

hicle in the simulation. By updating the D_2 utility functions accordingly, the decision *Max speed reduction & Obstacle buffer size increase* is assigned the highest utility score in these situations, making it the selected driving precaution measure. This adjustment leads to a noticeable improvement, as shown in Table 5.4. However, simulation testing scenario 1 does not show significant improvement before and after the ID update. Further investigation reveals that scenario 1 is designed to be a particularly challenging situation for the ego vehicle to stop in time, necessitating a more direct safety fallback strategy rather than driving precaution measures. Although only the results of past failure scenarios are presented in Table 5.4 to compare DBNID improvements before and after ID updates, other scenarios that originally DBNID can successfully handle were also tested. The safety performance in these scenarios is either improved or maintained at the same level as the original, indicating that the ID updates in DBNID based on identified past failure cases do not result in overfitting.

Another important perspective to examine is the estimation accuracy of the two DBN hidden nodes, $COTC_t$ and TMC_t , as it represents the uncertainty propagation effect from the upstream module and forms the foundation for subsequent uncertainty-aware decision-making. Based on results from both simulation and real-world experiments, these two hidden nodes can be estimated accurately using the EM algorithm. The general estimation accuracy for both $COTC_t$ and TMC_t is around 90%, with two exceptions for $COTC_t$ during validation and sunny testing scenarios in real-world experiments. The accuracy drops in these cases are likely due to the overcrowded pedestrians on the ring road encountered during the rosbag data recording. Unlike the snowy testing scenarios, which have fewer pedestrians, the overcrowded conditions challenge object detection and tracking, leading to inconsistent trajectory predictions for single pedestrians. Consequently, the confidence score for COTC may not be accurately predicted. In contrast, with fewer pedestrians in snowy testing scenarios, the $COTC_t$ estimation accuracy returns to a very high level.

Apart from the DBNID progress in simulation and real-world experiments, future work of this uncertainty-aware decision-making technique will focus on several key areas to enhance its performance and robustness. Firstly, new decision strategies should be developed to handle extreme cases that the current ID update cannot address. This will involve identifying and implementing additional safety measures and fallback strategies to ensure the ego vehicle can manage even the most challenging scenarios. Secondly, we aim to integrate the DBNID system onto the WATonoBus platform to facilitate real-world online testing and validation. This integration will enable us to assess the system’s performance in dynamic, real-time environments and gather valuable insights. Additionally, we will collect more challenging rosbag data from real-world experiments to further test and refine the proposed model. This data will help us simulate a broader range of scenarios and

improve the system’s robustness. Lastly, we plan to integrate other types of uncertainties into the decision-making process, such as environmental uncertainties and system-level uncertainties, to create a more comprehensive and resilient decision-making framework. This holistic approach will ensure the DBNID method can adapt to various real-world complexities, enhancing its overall effectiveness and safety.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

This research introduces a novel approach to enhancing ADS safety by incorporating learning-based algorithm uncertainties and conducting a comprehensive safety analysis that considers both system and interaction malfunctions. It develops a real-time, adaptive uncertainty quantification method and a dual-tiered decision-making algorithm that balances safety with more flexible responses beyond conservative emergency stops. The contributions include implementing and verifying these methods on WATonoBus, constructing a qualitative uncertainty-hazard relationship, and achieving improved performance in uncertainty quantification and decision-making compared to baseline methods. To achieve this, three sequential objectives are defined: identifying, quantifying ADS learning algorithm uncertainties, and incorporating them within the decision-making process.

To address the first objective of identifying uncertainties, a sequential qualitative safety analysis using HAZOP and STPA methods was developed. HAZOP identifies malfunctions and hazards in individual functional modules, while STPA uncovers loss scenarios triggered by UCAs across multiple modules. This combined approach effectively addresses the complexity of ADS systems. By generating comprehensive test scenarios based on identified uncertainties, this analysis provides critical insights into the vulnerabilities of current ADS designs, guiding subsequent uncertainty quantification efforts. The effectiveness of this approach was demonstrated through its application to the WATonoBus project. Through ASIL risk assessment and test case generation, 99 test cases were developed across 9 atomic scenarios. These test cases form the basis for subsequent uncertainty-aware decision-making research.

The impact of this work lies in its ability to systematically associate identified hazards with specific learning algorithm uncertainties, thereby offering a clear pathway to understanding how these uncertainties propagate through the system and affect overall safety. This analysis equips engineers and researchers with actionable insights to design more robust testing protocols. The generated test cases are particularly valuable, as they enable the replication of real-world scenarios where these uncertainties could lead to failures, thus providing a practical tool for improving the resilience of ADS designs.

The second objective, quantifying identified uncertainties, was achieved by implementing PCE with HD map knowledge. This method models probability distributions in complex systems using polynomial basis functions, offering strengths in interpretability and flexibility for online updates—crucial for addressing data shift issues in ADS applications. The PCE method was applied to quantify pedestrian position uncertainty in WATonoBus’s perception module. Extensive simulations and real-world experiments showed that PCE performs better than other methods like bPCE and PCK in both accuracy and computation time. Its ability to adapt to dynamic conditions further validates its reliability.

The significance lies in its ability to quantify how uncertainties in input data, such as pedestrian positions, can propagate through perception systems and affect downstream decision-making processes. In addition, by applying this method to the WATonoBus project, the research demonstrated how PCE can improve the accuracy of uncertainty quantification in real-world ADS operations. Furthermore, the ability of PCE to continuously update with new data also addresses the critical issue of data shift, which is a common challenge in ADS applications. Therefore, this work also provides a scalable solution that can be adapted to various functional modules, eventually enhancing ADS safety and robustness.

The third objective, integrating quantified uncertainty into decision-making, was accomplished through DBNID method. DBNID traces uncertainty propagation from the quantified OPU to downstream risk assessment measures like TTC. The method allows for dual-tiered decision-making, adjusting driving maneuvers and precaution measures based on real-time uncertainty updates. DBNID demonstrated high interpretability and flexibility, outperforming the FSM baseline algorithm used in WATonoBus. By continuously monitoring perception and prediction uncertainties, DBNID enabled more cautious driving in uncertain scenarios, with further improvements achieved through utility function updates during simulations. Real-world experiments confirmed the effectiveness of DBNID, achieving comparable safety performance with satisfactory real-time response.

The importance of this work lies in its potential to enhance the safety of ADS by ensuring that decision-making processes are informed by the most current and accurate

understanding of environmental uncertainties. The method’s high interpretability and ability to dynamically update utility functions based on past failures make it a powerful tool for improving ADS reliability. Through rigorous simulations and real-world experiments, DBNID has demonstrated its capacity to instill a more cautious and context-aware driving behavior in uncertain scenarios. This contribution is crucial for building public trust in ADS and accelerating their adoption in real-world applications.

6.2 Future Works

The research presented in this thesis aims to enhance the safety and interpretability of high-level ADS decision-making through the integration of uncertainty quantification and propagation techniques. However, several areas merit further investigation to explore the potential of the proposed approach.

Refinement on Continuous Configuration Parameter Modeling in Test Case Generation

During the test case generation process following the ADS safety qualitative analysis, an atomic scenario is broken down into discrete and continuous configuration parameters for combinatorial testing. In this thesis, the continuous configuration parameters are modeled as Gaussian distributions with means and variances determined based on field test knowledge. While this simple modeling technique efficiently represents the main characteristics of the interested continuous parameters, it may not account for the joint influence of other factors. For example, variations in road surface conditions can impact the maximum acceleration of the actor vehicles. Therefore, further refinement of continuous configuration parameters is necessary to make simulated test cases more realistic and to ensure that the obtained testing results are more accurate and applicable to real-world scenarios.

Incorporating Diverse Uncertainties into Uncertainty Quantification and Decision-making Framework

To create a more comprehensive and resilient framework on learning algorithm uncertainty quantification and corresponding decision making, future research should integrate various types of uncertainties into both PCE and DBNID processes. This requires conducting comprehensive studies to understand the transfer of uncertainty from reference objects,

like stop signs, to dynamic entities such as moving vehicles. Given that the assumptions made for pedestrians may not apply to vehicles due to differences in size, reflection rates, and behavior, further research is needed to develop accurate models for vehicle OPU. Additionally, expanding the scope to other uncertainties, such as localization uncertainty based on lane boundary detection and HD map knowledge, can validate the general applicability and robustness of the proposed method, ensuring the decision-making process to be more adaptable to real-world complexities.

Enhancing Decision Strategies in DBNID for Extreme Scenarios

Future work should focus on developing new decision strategies within the DBNID framework to handle extreme and unforeseen scenarios that the current ID update mechanism cannot address. This includes identifying additional safety measures and fallback strategies to ensure the ego vehicle can safely navigate even the most challenging conditions. Integrating the DBNID system onto the WATonoBus platform will facilitate real-world online testing and validation, providing insights into its performance in dynamic, real-time environments. Collecting and analyzing more challenging real-world data, such as rosbag recordings, will help refine these strategies and improve the system's robustness against a broader range of scenarios.

Integration with WATonoBus and Performing Real-World Validation

Integrating DBNID onto platforms like WATonoBus for real-world online testing and validation represents a crucial step towards bridging the gap between simulation and practical implementation. By deploying the DBNID system in real-world settings, researchers can observe its performance under diverse environmental conditions, traffic scenarios, and unexpected events. This live testing environment provides invaluable opportunities to assess the system's robustness, identify potential weaknesses, and fine-tune decision-making algorithms in response to real-time feedback.

References

- [1] Shamus P Smith and Michael D Harrison. Improving hazard classification through the reuse of descriptive arguments. In *International Conference on Software Reuse*, pages 255–268. Springer, 2002.
- [2] Reyhaneh Sadeghi and Floris Goerlandt. A proposed validation framework for the system theoretic process analysis (stpa) technique. *Safety science*, 162:106080, 2023.
- [3] Torben Stolte, Gerrit Bagschik, Andreas Reschka, and Markus Maurer. Hazard analysis and risk assessment for an automated unmanned protective vehicle. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1848–1855. IEEE, 2017.
- [4] Renran Tian, Stanley Chien, Yaobin Chen, and Rini Sherony. Pedestrian moving patterns during potential conflicts with 110 on-road driving vehicles. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pages 2036–2040. SAGE Publications Sage CA: Los Angeles, CA, 2019.
- [5] Finn V Jensen and Thomas Dyhre Nielsen. *Bayesian networks and decision graphs*, volume 2. Springer, 2007.
- [6] Eric Thorn, Shawn C Kimmel, Michelle Chaka, Booz Allen Hamilton, et al. A framework for automated driving system testable cases and scenarios. Technical report, United States. Department of Transportation. National Highway Traffic Safety ..., 2018.
- [7] ISO 26262. Road vehicles — functional safety — part 1: Vocabulary. Technical report, International Organization for Standardization, Geneva, CH, 2018.
- [8] SAE-J3016. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. Technical report, SAE International, 2021.

- [9] James M Anderson, Kalra Nidhi, Karlyn D Stanley, Paul Sorensen, Constantine Samaras, and Oluwatobi A Oluwatola. *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation, 2014.
- [10] ISO 21448:2022. Road vehicles — safety of the intended functionality. Technical report, International Organization for Standardization, Geneva, CH, 2022.
- [11] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.
- [12] Rick Salay and Krzysztof Czarnecki. Using machine learning safely in automotive software: An assessment and adaption of software process requirements in iso 26262. *arXiv preprint arXiv:1808.01614*, 2018.
- [13] Joshua Hassol, David Perlman, Lora Chajka-Cadin, Jingsi Shaw, et al. Understanding surveys of public sentiment regarding automated vehicles: Summary of results to date and implications of past research on the dynamics of consumer adoption. Technical report, United States. Department of Transportation, 2019.
- [14] Philip Koopman and Michael Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017.
- [15] Jan Erik Stellet, Jan Schumacher, Wolfgang Branz, and J Marius Zöllner. Uncertainty propagation in criticality measures for driver assistance. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1187–1194. IEEE, 2015.
- [16] Chen Sun, Ruihe Zhang, Yukun Lu, Yaodong Cui, Zejian Deng, Dongpu Cao, and Amir Khajepour. Toward ensuring safety for autonomous driving perception: Standardization progress, research advances, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [17] Ali Harakeh, Michael Smart, and Steven L Waslander. Bayesod: A bayesian approach for uncertainty estimation in deep object detectors. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 87–93. IEEE, 2020.
- [18] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

- [19] Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1:187–210, 2018.
- [20] Keshav Bimbraw. Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology. In *2015 12th international conference on informatics in control, automation and robotics (ICINCO)*, volume 1, pages 191–198. IEEE, 2015.
- [21] Rolf Johansson and Jonas Nilsson. The need for an environment perception block to address all asil levels simultaneously. In *2016 IEEE intelligent vehicles symposium (IV)*, pages 1–4. IEEE, 2016.
- [22] US department of transportation. Av 4.0. Technical report, US department of transportation, 2019.
- [23] European Union. Regulation (eu) 2019/2144 on type-approval requirements for motor vehicles and their trailers, and systems, components and separate technical units intended for such vehicles, as regards their general safety and the protection of vehicle occupants and vulnerable road users. Technical report, European Union, 2019.
- [24] Philip Koopman. *The UL 4600 Guidebook: What to Include in an Autonomous Vehicle Safety Case*. Carnegie Mellon University, 2022.
- [25] ISO 17361:2017. Intelligent transport systems – lane departure warning systems – performance requirements and test procedures. Technical report, International Organization for Standardization, Geneva, CH, 2017.
- [26] ISO 15622:2018. Intelligent transport systems – adaptive cruise control systems – performance requirements and test procedures. Technical report, International Organization for Standardization, Geneva, CH, 2018.
- [27] ISO 20035:2019. Intelligent transport systems – cooperative adaptive cruise control systems (cacc) – performance requirements and test procedures. Technical report, International Organization for Standardization, Geneva, CH, 2019.
- [28] ISO 22737:2021. Intelligent transport systems – low-speed automated driving (lsad) systems for predefined routes – performance requirements, system requirements and performance test procedures. Technical report, International Organization for Standardization, Geneva, CH, 2021.

- [29] ISO/DIS 23374-1. Intelligent transport systems – automated valet parking systems (avps) – part 1: System framework, requirements for automated driving, and communication interface. Technical report, International Organization for Standardization, Geneva, CH, 2023.
- [30] ISO 19237:2017. Intelligent transport systems – pedestrian detection and collision mitigation systems (pdcms) – performance requirements and test procedures. Technical report, International Organization for Standardization, Geneva, CH, 2017.
- [31] ISO 22078:2020. Intelligent transport systems – bicyclist detection and collision mitigation systems (bdcms) – performance requirements and test procedures. Technical report, International Organization for Standardization, Geneva, CH, 2020.
- [32] Philip Koopman and Michael Wagner. Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety*, 4(1):15–24, 2016.
- [33] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.
- [34] Tchoya Florence Koné, Eric Bonjour, Eric Levrat, Frédérique Mayer, and Stéphane Géronimi. Safety demonstration of autonomous vehicles: A review and future research questions. In *Complex Systems Design & Management: Proceedings of the Tenth International Conference on Complex Systems Design & Management, CSD&M Paris 2019*, pages 176–188. Springer, 2020.
- [35] Masao Ito. The uncertainty that the autonomous car faces and predictability analysis for evaluation. In *Systems, Software and Services Process Improvement: 26th European Conference, EuroSPI 2019, Edinburgh, UK, September 18–20, 2019, Proceedings 26*, pages 83–95. Springer, 2019.
- [36] Michael Kläs and Lena Sembach. Uncertainty wrappers for data-driven models: increase the transparency of ai/ml-based models through enrichment with dependable situation-aware uncertainty estimates. In *Computer Safety, Reliability, and Security: SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings 38*, pages 358–364. Springer, 2019.
- [37] Michael Kläs. Towards identifying and managing sources of uncertainty in ai and machine learning models-an overview. *arXiv preprint arXiv:1811.11669*, 2018.

- [38] Oliver Willers, Sebastian Sudholt, Shervin Raafatnia, and Stephanie Abrecht. Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks. In *Computer Safety, Reliability, and Security. SAFECOMP 2020 Workshops: DECSoS 2020, DepDevOps 2020, USDAI 2020, and WAISE 2020, Lisbon, Portugal, September 15, 2020, Proceedings 39*, pages 336–350. Springer, 2020.
- [39] Liangliang Sun, Yan-Fu Li, and Enrico Zio. Comparison of the hazop, fmea, fram, and stpa methods for the hazard analysis of automatic emergency brake systems. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 8(3):031104, 2022.
- [40] Paul Aoanan. *A Systematic Approach to Hazard and Operability Study (HAZOP)*. PhD thesis, McMaster University, 2021.
- [41] Yi Qi, Yi Dong, Siddartha Khastgir, Paul Jennings, Xingyu Zhao, and Xiaowei Huang. Stpa for learning-enabled systems: a survey and a new practice. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1381–1388. IEEE, 2023.
- [42] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. Survey on scenario-based safety assessment of automated vehicles. *IEEE access*, 8:87456–87477, 2020.
- [43] Thomas Demeillers, Nicolas Dowding, Florian Fillol, Roman Mouchel, Elsa Serna, Baptiste Valentin, and Benoit Vinière. Pegasus project technical report. Technical report, Université de Toulouse, 2021.
- [44] Antona-Makoshi Jacobo, Uchida Nobuyuki, Yamazaki Kunio, Ozawa Koichiro, Kitahara Eiichi, and Taniguchi Satoshi. Development of a safety assurance process for autonomous vehicles in japan. In *Proceedings of ESV Conference*, 2019.
- [45] Sebastian Geyer, Marcel Baltzer, Benjamin Franz, Stephan Hakuli, Michaela Kauer, Martin Kienle, Sonja Meier, Thomas Weißgerber, Klaus Bengler, Ralph Bruder, et al. Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance. *IET Intelligent Transport Systems*, 8(3):183–189, 2014.
- [46] Gerrit Bagschik, Till Menzel, and Markus Maurer. Ontology based scene creation for the development of automated vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1813–1820. IEEE, 2018.

- [47] Wei Chen and Leila Kloul. An ontology-based approach to generate the advanced driver assistance use cases of highway traffic. In *10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management.*, 2018.
- [48] Florian Klück, Yihao Li, Mihai Nica, Jianbo Tao, and Franz Wotawa. Using ontologies for test suites generation for automated and autonomous driving functions. In *2018 IEEE International symposium on software reliability engineering workshops (ISSREW)*, pages 118–123. IEEE, 2018.
- [49] Jacob Langner, Johannes Bach, Lennart Ries, Stefan Otten, Marc Holzäpfel, and Eric Sax. Estimating the uniqueness of test scenarios derived from recorded real-world-driving-data using autoencoders. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1860–1866. IEEE, 2018.
- [50] Robert Krajewski, Tobias Moers, Dominik Nerger, and Lutz Eckstein. Data-driven maneuver modeling using generative adversarial networks and variational autoencoders for safety validation of highly automated vehicles. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2383–2390. IEEE, 2018.
- [51] Ian Rhys Jenkins, Ludvig Oliver Gee, Alessia Knauss, Hang Yin, and Jan Schroeder. Accident scenario generation with recurrent neural networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3340–3345. IEEE, 2018.
- [52] Stefan Jesenski, Jan Erik Stellet, Florian Schiegg, and J Marius Zöllner. Generation of scenes in intersections for the validation of highly automated driving functions. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 502–509. IEEE, 2019.
- [53] Bingjian Yue, Shuming Shi, Shuo Wang, and Nan Lin. Low-cost urban test scenario generation using microscopic traffic simulation. *IEEE Access*, 8:123398–123407, 2020.
- [54] Elias Rocklage, Heiko Kraft, Abdullah Karatas, and Jörg Seewig. Automated scenario generation for regression testing of autonomous vehicles. In *2017 IEEE 20th international conference on intelligent transportation systems (itsc)*, pages 476–483. IEEE, 2017.
- [55] Galen E Mullins, Paul G Stankiewicz, and Satyandra K Gupta. Automated generation of diverse and challenging scenarios for test and evaluation of autonomous

- vehicles. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 1443–1450. IEEE, 2017.
- [56] Rodrigo Queiroz, Thorsten Berger, and Krzysztof Czarnecki. Geoscenario: An open dsl for autonomous driving scenario representation. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 287–294. IEEE, 2019.
 - [57] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019.
 - [58] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don’t know? *arXiv preprint arXiv:1810.09136*, 2018.
 - [59] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.
 - [60] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.
 - [61] M Jordan, J Kleinberg, and B Schölkopf. Information science and statistics. (*No Title*), 2006.
 - [62] Yarín Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.
 - [63] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.
 - [64] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.
 - [65] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

- [66] Davis Josiah, Zhu Jason, and Oldfather Jeremy. Quantifying uncertainty in deep learning systems - aws prescriptive guidance. Technical report, AWS, 2020.
- [67] Yarin Gal et al. Uncertainty in deep learning. *PhD thesis*, 2016.
- [68] Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 41–50, 2019.
- [69] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020.
- [70] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.
- [71] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015.
- [72] Chunyuan Li, Changyou Chen, Yunchen Pu, Ricardo Henao, and Lawrence Carin. Communication-efficient stochastic gradient mcmc for neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4173–4180, 2019.
- [73] Liang Peng, Hong Wang, and Jun Li. Uncertainty evaluation of object detection algorithms for autonomous vehicles. *Automotive Innovation*, 4(3):241–252, 2021.
- [74] Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 2019.
- [75] Tanya Nair, Doina Precup, Douglas L Arnold, and Tal Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical image analysis*, 59:101557, 2020.
- [76] Alain Jungo, Richard McKinley, Raphael Meier, Urspeter Knecht, Luis Vera, Julián Pérez-Beteta, David Molina-García, Víctor M Pérez-García, Roland Wiest, and

- Mauricio Reyes. Towards uncertainty-assisted brain tumor segmentation and survival prediction. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers 3*, pages 474–485. Springer, 2018.
- [77] Adrian Tousignant, Paul Lemaître, Doina Precup, Douglas L Arnold, and Tal Arbel. Prediction of disease progression in multiple sclerosis patients using deep learning analysis of mri data. In *International conference on medical imaging with deep learning*, pages 483–492. PMLR, 2019.
 - [78] Alireza Norouzi, Ali Emami, Kayvan Najarian, Nader Karimi, SM Reza Soroushmehr, et al. Exploiting uncertainty of deep neural networks for improving segmentation accuracy in mri images. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2322–2326. IEEE, 2019.
 - [79] Abhijit Guha Roy, Sailesh Conjeti, Nassir Navab, Christian Wachinger, Alzheimer’s Disease Neuroimaging Initiative, et al. Bayesian quicknat: Model uncertainty in deep whole-brain segmentation for structure-wise quality control. *NeuroImage*, 195:11–22, 2019.
 - [80] Thomas Vandal, Max Livingston, Camen Piho, and Sam Zimmerman. Prediction and uncertainty quantification of daily airport flight delays. In *International conference on predictive applications and APIs*, pages 45–51. PMLR, 2018.
 - [81] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.
 - [82] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
 - [83] Christopher M Bishop. Mixture density networks. Technical report, Aston University, 1994.
 - [84] Andrew Foong, David Burt, Yingzhen Li, and Richard Turner. On the expressiveness of approximate inference in bayesian neural networks. *Advances in Neural Information Processing Systems*, 33:15897–15908, 2020.

- [85] Aditya Siddhant and Zachary C Lipton. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. *arXiv preprint arXiv:1808.05697*, 2018.
- [86] Noble Kennamer, Alexander Ihler, and David Kirkby. Empirical study of mc-dropout in various astronomical observing conditions. In *CVPR workshops*, pages 17–20, 2019.
- [87] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.
- [88] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.
- [89] Gregory Kahn, Adam Villafior, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.
- [90] Björn Lütjens, Michael Everett, and Jonathan P How. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8662–8668. IEEE, 2019.
- [91] Mingliang Yang, Kun Jiang, Junze Wen, Liang Peng, Yanding Yang, Hong Wang, Mengmeng Yang, Xinyu Jiao, and Diange Yang. Real-time evaluation of perception uncertainty and validity verification of autonomous driving. *Sensors*, 23(5):2867, 2023.
- [92] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.
- [93] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020.
- [94] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.
- [95] Mario Berk, Olaf Schubert, Hans-Martin Kroll, Boris Buschardt, and Daniel Straub. Exploiting redundancy for reliability analysis of sensor perception in automated driving vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(12):5073–5085, 2019.

- [96] Xiupeng Shi, Yiik Diew Wong, Chen Chai, and Michael Zhi-Feng Li. An automated machine learning (automl) method of risk prediction for decision-making of autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(11):7145–7154, 2020.
- [97] Chen Sun, Yaodong Cui, Ngoc-Dung DJao, Reza Valiollahi Mehrizi, Mohammad Pirani, and Amir Khajepour. Medium-fidelity evaluation and modeling for perception systems of intelligent and connected vehicles. *IEEE Transactions on Intelligent Vehicles*, 2023.
- [98] ASURVEY OF EMERGING TRENDS. Run-time monitoring of machine learning for robotic perception: Asurvey of emerging trends. *arXiv preprint*, 2021.
- [99] Nataraj Jammalamadaka, Andrew Zisserman, Marcin Eichner, Vittorio Ferrari, and CV Jawahar. Has my algorithm succeeded? an evaluator for human pose estimators. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III 12*, pages 114–128. Springer, 2012.
- [100] Peng Zhang, Jiuling Wang, Ali Farhadi, Martial Hebert, and Devi Parikh. Predicting failures of vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3566–3573, 2014.
- [101] Wei Zhou, Julie Stephany Berrio, Stewart Worrall, and Eduardo Nebot. Automated evaluation of semantic segmentation robustness for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):1951–1963, 2019.
- [102] Heng Yang and Ioannis Patras. Mirror, mirror on the wall, tell me, is the error small? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4685–4693, 2015.
- [103] Kai Yang, Xiaolin Tang, Jun Li, Hong Wang, Guichuan Zhong, Jiaxin Chen, and Dongpu Cao. Uncertainties in onboard algorithms for autonomous vehicles: Challenges, mitigation, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 2023.
- [104] Vibhu Gautam, Youcef Gheraibia, Rob Alexander, and Richard David Hawkins. Runtime decision making under uncertainty in autonomous vehicles. In *Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI 2021)*. CEUR Workshop Proceedings, 2021.

- [105] Boris Ivanovic, Yifeng Lin, Shubham Shrivastava, Punarjay Chakravarty, and Marco Pavone. Propagating state uncertainty through trajectory forecasting. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2351–2358. IEEE, 2022.
- [106] Iwo Kurzidem, Ahmad Saad, and Philipp Schleiß. A systematic approach to analyzing perception architectures in autonomous vehicles. In *Model-Based Safety and Assessment: 7th International Symposium, IMBSA 2020, Lisbon, Portugal, September 14–16, 2020, Proceedings 7*, pages 149–162. Springer, 2020.
- [107] Chen Chen, Xiaomin Liu, Hsiao-Hwa Chen, Meilian Li, and Liqiang Zhao. A rear-end collision risk evaluation and control scheme using a bayesian network model. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):264–284, 2018.
- [108] Iago Pacheco Gomes and Denis Fernando Wolf. Health monitoring system for autonomous vehicles using dynamic bayesian networks for diagnosis and prognosis. *Journal of Intelligent & Robotic Systems*, 101(1):19, 2021.
- [109] Matthias Schreier, Volker Willert, and Jürgen Adamy. An integrated approach to maneuver-based trajectory prediction and criticality assessment in arbitrary road environments. *IEEE Transactions on Intelligent Transportation Systems*, 17(10):2751–2766, 2016.
- [110] Samyeul Noh and Kyoungwhan An. Decision-making framework for automated driving in highway environments. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):58–71, 2017.
- [111] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.
- [112] Arda Kurt and Ümit Özgüner. Hierarchical finite state machines for autonomous mobile systems. *Control Engineering Practice*, 21(2):184–194, 2013.
- [113] Jie Ji, Amir Khajepour, Wael William Melek, and Yanjun Huang. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. *IEEE Transactions on Vehicular Technology*, 66(2):952–964, 2016.
- [114] Yongli Chen, Shen Li, Xiaolin Tang, Kai Yang, Dongpu Cao, and Xianke Lin. Interaction-aware decision making for autonomous vehicles. *IEEE Transactions on Transportation Electrification*, 2023.

- [115] Martin Buehler, Karl Iagnemma, and Sanjiv Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*, volume 56. Springer Science & Business Media, 2009.
- [116] Sebastian Brechtel, Tobias Gindele, and Rüdiger Dillmann. Probabilistic mdp-behavior planning for cars. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1537–1542. IEEE, 2011.
- [117] Simon Ulbrich and Markus Maurer. Probabilistic online pomdp decision making for lane changes in fully automated driving. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 2063–2067. IEEE, 2013.
- [118] Carl-Johan Hoel, Krister Wolff, and Leo Laine. Tactical decision-making in autonomous driving by reinforcement learning with uncertainty estimation. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1563–1569. IEEE, 2020.
- [119] Maxime Bouton, Jesper Karlsson, Alireza Nakhaei, Kikuo Fujimura, Mykel J Kochenderfer, and Jana Tumova. Reinforcement learning with probabilistic guarantees for autonomous driving. *arXiv preprint arXiv:1904.07189*, 2019.
- [120] Yu Fan Chen, Michael Everett, Miao Liu, and Jonathan P How. Socially aware motion planning with deep reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1343–1350. IEEE, 2017.
- [121] Michael Everett, Yu Fan Chen, and Jonathan P How. Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9:10357–10377, 2021.
- [122] Meha Kaushik, Vignesh Prasad, K Madhava Krishna, and Balaraman Ravindran. Overtaking maneuvers in simulated highway driving using deep reinforcement learning. In *2018 IEEE intelligent vehicles symposium (iv)*, pages 1885–1890. IEEE, 2018.
- [123] Alexis C Madrigal. The atlantic, May 2014.
- [124] Jiaxin Liu, Hong Wang, Liang Peng, Zhong Cao, Diange Yang, and Jun Li. Pnnuad: Perception neural networks uncertainty aware decision-making for autonomous vehicle. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24355–24368, 2022.
- [125] Xiaolin Tang, Kai Yang, Hong Wang, Jiahang Wu, Yechen Qin, Wenhao Yu, and Dongpu Cao. Prediction-uncertainty-aware decision-making for autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 7(4):849–862, 2022.

- [126] Weitao Zhou, Zhong Cao, Yunkang Xu, Nanshan Deng, Xiaoyu Liu, Kun Jiang, and Diange Yang. Long-tail prediction uncertainty aware trajectory planning for self-driving vehicles. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1275–1282. IEEE, 2022.
- [127] Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. *international joint conferences on artificial intelligence*, 2017.
- [128] Carl-Johan Hoel, Katherine Driggs-Campbell, Krister Wolff, Leo Laine, and Mykel J Kochenderfer. Combining planning and deep reinforcement learning in tactical decision making for autonomous driving. *IEEE transactions on intelligent vehicles*, 5(2):294–305, 2019.
- [129] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.
- [130] Constantin Hubmann, Jens Schulz, Marvin Becker, Daniel Althoff, and Christoph Stiller. Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. *IEEE transactions on intelligent vehicles*, 3(1):5–17, 2018.
- [131] Hanna Kurniawati and Vinay Yadav. An online pomdp solver for uncertainty planning in dynamic environment. In *Robotics Research: The 16th International Symposium ISRR*, pages 611–629. Springer, 2016.
- [132] Maxime Bouton, Alireza Nakhaei, Kikuo Fujimura, and Mykel J Kochenderfer. Scalable decision making with sensor occlusions for autonomous driving. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 2076–2081. IEEE, 2018.
- [133] Man Cheol Kim. Reliability block diagram with general gates and its application to system reliability analysis. *Annals of Nuclear Energy*, 38(11):2456–2461, 2011.
- [134] Robin Schubert and Gerd Wanielik. A unified bayesian approach for object and situation assessment. *IEEE Intelligent Transportation Systems Magazine*, 3(2):6–19, 2011.

- [135] Robin Schubert. Evaluating the utility of driving: Toward automated decision making under uncertainty. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):354–364, 2011.
- [136] Dimia Iberraken, Lounis Adouane, and Dieumet Denis. Multi-level bayesian decision-making for safe and flexible autonomous navigation in highway environment. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3984–3990. IEEE, 2018.
- [137] Christopher Becker, Larry Yount, Shane Rozen-Levy, John Brewer, et al. Functional safety assessment of an automated lane centering system. Technical report, United States. Department of Transportation. National Highway Traffic Safety ... , 2018.
- [138] Changhai Nie and Hareton Leung. A survey of combinatorial testing. *ACM Computing Surveys (CSUR)*, 43:1–29, 2011.
- [139] Dongbin Xiu. *Numerical methods for stochastic computations: a spectral method approach*. Princeton university press, 2010.
- [140] Dongbin Xiu and Jan S Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.
- [141] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.
- [142] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.
- [143] Bruno Sudret. *Uncertainty propagation and sensitivity analysis in mechanical models Contributions to structural reliability and stochastic spectral methods*. PhD thesis, ETH Zurich, 2008.
- [144] S. Marelli, N. Lüthen, and B. Sudret. UQLab user manual – Polynomial chaos expansions. Technical report, Chair of Risk, Safety and Uncertainty Quantification, ETH Zurich, Switzerland, 2022. Report UQLab-V2.0-104.
- [145] Marc Berveiller, Bruno Sudret, and Maurice Lemaire. Stochastic finite element: a non intrusive approach by regression. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*, 15(1-3):81–92, 2006.

- [146] Giovanni Migliorati, Fabio Nobile, Erik von Schwerin, and Raúl Tempone. Approximation of quantities of interest in stochastic pdes by the random discrete l^2 projection on polynomial spaces. *SIAM Journal on Scientific Computing*, 35(3):A1440–A1460, 2013.
- [147] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.
- [148] Trevor Hastie, Jonathan Taylor, Robert Tibshirani, and Guenther Walther. Forward stagewise regression and the monotone lasso. *arxiv preprint*, 2007.
- [149] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 2004.
- [150] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of computational Physics*, 230(6):2345–2367, 2011.
- [151] Stefano Marelli and Bruno Sudret. An active-learning algorithm that combines sparse polynomial chaos expansions and bootstrap for structural reliability analysis. *Structural Safety*, 75:67–74, 2018.
- [152] Roland Schobi, Bruno Sudret, and Joe Wiart. Polynomial-chaos-based kriging. *International Journal for Uncertainty Quantification*, 5(2), 2015.
- [153] Pierrick Kersaudy, Bruno Sudret, Nadège Varsier, Odile Picon, and Joe Wiart. A new surrogate modeling technique combining kriging and polynomial chaos expansions—application to uncertainty analysis in computational dosimetry. *Journal of Computational Physics*, 286:103–117, 2015.
- [154] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.
- [155] Sundas Iftikhar, Zuping Zhang, Muhammad Asim, Ammar Muthanna, Andrey Koucheryavy, and Ahmed A Abd El-Latif. Deep learning-based pedestrian detection in autonomous vehicles: Substantial issues and challenges. *Electronics*, 11(21):3551, 2022.
- [156] Long Chen, Shaobo Lin, Xiankai Lu, Dongpu Cao, Hangbin Wu, Chi Guo, Chun Liu, and Fei-Yue Wang. Deep neural network based vehicle and pedestrian detection

- for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 22(6):3234–3246, 2021.
- [157] Jessica Van Brummelen, Marie O’Brien, Dominique Gruyer, and Hodayoun Najjaran. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation research part C: emerging technologies*, 89:384–406, 2018.
 - [158] Juan Carrillo and Steven Waslander. Urbannet: Leveraging urban maps for long range 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3799–3806. IEEE, 2021.
 - [159] Rafael Peixoto Derenzi Vivacqua, Massimo Bertozzi, Pietro Cerri, Felipe Nascimento Martins, and Raquel Frizera Vassallo. Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(2):582–597, 2017.
 - [160] Ehsan Javanmardi, Mahdi Javanmardi, Yanlei Gu, and Shunsuke Kamijo. Factors to evaluate capability of map for vehicle localization. *IEEE Access*, 6:49850–49867, 2018.
 - [161] Zhiqiang Jian, Songyi Zhang, Shitao Chen, Xin Lv, and Nanning Zheng. High-definition map combined local motion planning and obstacle avoidance for autonomous driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2180–2186. IEEE, 2019.
 - [162] Alejandro Diaz et al. *Path planning and map monitoring for self-driving vehicles based on HD maps*. PhD thesis, Universidad de Alcalá, 2021.
 - [163] Gamal Elghazaly, Raphaël Frank, Scott Harvey, and Stefan Safko. High-definition maps: Comprehensive survey, challenges and future perspectives. *IEEE Open Journal of Intelligent Transportation Systems*, 2023.
 - [164] Guoqin Zang, Shéhérazade Azouigui, Sébastien Saudrais, Olivier Peyricot, and Mathieu Hébert. Quantitative study of vehicle-pedestrian interactions: Towards pedestrian-adapted lighting communication functions for autonomous vehicles. *Electronic Imaging*, 33:1–8, 2021.
 - [165] Julian FP Kooij, Nicolas Schneider, and Darius M Gavrilă. Analysis of pedestrian dynamics from a vehicle perspective. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1445–1450. IEEE, 2014.

- [166] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 26(2):211–243, 1964.
- [167] Jeffrey C Lagarias, James A Reeds, Margaret H Wright, and Paul E Wright. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on optimization*, 9(1):112–147, 1998.
- [168] Queensland Government. Stopping distances on wet and dry roads. Technical report, Queensland Government, 2016.
- [169] Jonathan P Powell and Roberto Palacín. Passenger stability within moving railway vehicles: limits on maximum longitudinal acceleration. *Urban Rail Transit*, 1:95–103, 2015.
- [170] Stefano Marelli and Bruno Sudret. Uqlab: A framework for uncertainty quantification in matlab. In *Vulnerability, uncertainty, and risk: quantification, mitigation, and management*, pages 2554–2563. 2014.
- [171] Chen Sun, Zejian Deng, Wenbo Chu, Shen Li, and Dongpu Cao. Acclimatizing the operational design domain for autonomous driving systems. *IEEE Intelligent Transportation Systems Magazine*, 14(2):10–24, 2021.
- [172] Xiao-Li Meng and David Van Dyk. The em algorithm—an old folk-song sung to a fast new tune. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 59(3):511–567, 1997.
- [173] Gregory F Cooper. A method for using belief networks as influence diagrams. *arXiv preprint arXiv:1304.2346*, 2013.
- [174] Weida Wang, Tianqi Qie, Chao Yang, Wenjie Liu, Changle Xiang, and Kun Huang. An intelligent lane-changing behavior prediction and decision-making strategy for an autonomous vehicle. *IEEE transactions on industrial electronics*, 69(3):2927–2937, 2021.

APPENDICES

Appendix A

HAZOP Analysis Tables

According to Table 3.7 in Section 3.3.3, 35 malfunctions are identified from the identified 9 functions. They are listed as below:

Table A.1: Causal Factors for the Implemented Sub-controllers

ID	Malfunction	Hazardous Conditions
<i>Function 1: Sense surrounding environment</i>		
F1-1	No sensor data	- Unable to adjust vehicle control - Unable to detect nearby dynamic objects
F1-2	Partially functional sensors	- Incorrectly adjust vehicle control - Omit certain nearby dynamic objects
<i>Function 2: Localize ego vehicle</i>		
F2-1	Ego vehicle cannot be localized	- Unable to adjust vehicle control - Unable to estimate distance to dynamic objects - Unable to understand dynamic objects' intention
F2-2	Ego vehicle localization accuracy is low	- Incorrectly adjust vehicle control - Incorrect estimation on distance to dynamic objects - Incorrect understanding of dynamic objects' intention
F2-3	Ego vehicle localization is inefficient	- Slow reaction to surrounding environment

<i>Function 3: Detect and localize dynamic traffic participants and static traffic features</i>		
F3-1	Traffic objects cannot be detected and localized	- Unable to adjust vehicle control - Unable to detect nearby dynamic objects - Unable to estimate distance to dynamic objects
F3-2	False negative cases in traffic object localization	- Incorrectly adjust vehicle control - Omit certain nearby dynamic objects
F3-3	False positive cases in traffic object localization	- Incorrectly adjust vehicle control - Incorrect estimation on distance to dynamic objects - Incorrect understanding of dynamic objects' intention
F3-4	Traffic objects localization is inefficient	- Slow reaction to surrounding environment
<i>Function 4: Predict dynamic traffic participants' movement trajectories</i>		
F4-1	Dynamic traffic objects' trajectories cannot be predicted	- Incorrectly adjust vehicle control - Incorrect estimation on distance to dynamic objects - Unable to understand dynamic objects' intention
F4-2	Dynamic traffic objects' trajectories are predicted incorrectly	- Incorrectly adjust vehicle control - Incorrect estimation on distance to dynamic objects - Incorrect understanding of dynamic objects' intention
F4-3	Dynamic traffic objects' trajectories are predicted inefficiently	- Slow reaction to surrounding environment
<i>Function 5: Instruct ego vehicle high-level autonomous driving motions</i>		
F5-1	Incorrect ego vehicle motion instructions	- Incorrectly adjust vehicle control
F5-2	Late ego vehicle motion instructions	- Slow reaction to surrounding environment
<i>Function 6: Identify traffic conflicts and related obstacles</i>		
F6-1	Potential traffic obstacles cannot be identified	- Incorrectly adjust vehicle control

F6-2	False negative cases in potential traffic obstacles identification	- Incorrectly adjust vehicle control - Omit certain nearby dynamic objects
F6-3	False positive cases in potential traffic obstacles identification	- Incorrectly adjust vehicle control - Incorrect estimation on distance to dynamic objects - Incorrect understanding of dynamic objects' intention
F6-4	Inefficient traffic obstacles identification	- Slow reaction to surrounding environment
<i>Function 7: Switch vehicle control authorities</i>		
F7-1	Unable to implement required vehicle control mode	- Unable to adjust vehicle control - Unable to proceed safety fallback strategy
F7-2	Implement required vehicle control mode inefficiently	- Slow reaction to surrounding environment
<i>Function 8: Plan local path for ego vehicle based on high-level motion instructions</i>		
F8-1	Unable to plan the path	- Unable to adjust vehicle control - Omit certain nearby dynamic objects
F8-2	Incorrectly plan the path	- Incorrectly adjust vehicle control
F8-2	Inefficient path planning	- Slow reaction to surrounding environment
<i>Function 9: Execute vehicle control on longitudinal and lateral directions</i>		
F9-1	Cannot execute low-level vehicle control	- Unable adjust vehicle control
F9-2	Low-level vehicle control is executed more than intended	- Incorrectly adjust vehicle control
F9-3	Low-level vehicle control is executed less than intended	- Incorrectly adjust vehicle control
F9-4	Inefficient low-level vehicle control exestuation	- Slow reaction to surrounding environment

Appendix B

STPA Analysis Tables

The generated unsafe control actions for each sub-controller identified in Section 3.3.2 are tabulated with the potentially triggered primary vehicle-level hazardous conditions:

Table B.1: Decision-making Module Control Actions

Context for Control Request	Guidewords for Unsafe Control Actions		
General vehicle motion request at certain traffic scenarios	Provided but is delivered incorrectly	Provided but too late	Provided but too late
Stop Ego Vehicle	PH3	PH3, PH5	PH3, PH5
Deaccelerate Ego Vehicle	PH3	PH3, PH5	PH3, PH5
Move Ego Vehicle	PH3 (Except for Stopping)	PH5	PH5
Pull Over Ego Vehicle	PH3 (Except for Stopping)	PH3, PH5	PH3, PH5
Merge Ego Vehicle	PH3 (Except for Stopping)	PH5	PH5

Table B.2: Motion Planning and Vehicle Control Modules Control Actions

Context for Control Request	Guidewords for Unsafe Control Actions					
Longitudinal position adjustment request based on the planned trajectories	Lateral position adjustment request based on the planned trajectories	Provided but not needed	Provided but more than intended	Provided but less than intended	Provided but too late	Not provided but needed
None	None	PH3	Not apply	Not apply	Not apply	Not apply
None	Lateral adjustment toward trajectory with steering angle modification	PH3	PH3	PH3	PH3, PH5	PH1
Longitudinal adjustment forward gas/brake pedal modification	None	PH3	PH3	PH3	PH3, PH5	PH1
Longitudinal adjustment forward gas/brake pedal modification	Lateral adjustment toward trajectory with steering angle modification	Not apply	PH3	PH3	PH3, PH5	PH1

Table B.3: Safety Strategy Selection Module Control Actions

Context for Control Request	Guidewords for Unsafe Control Actions		
Safety state request based on vehicle status monitoring results	Provided but is delivered incorrectly	Provided but too late	Not provided but needed
Normal autonomous driving	PH3, PH4	PH3, PH5	PH2
Constrained autonomous driving	PH3, PH4	PH3, PH5	PH2
Manual control	PH3, PH4	PH3, PH5	PH2

Table B.4: Control Mode Arbiter Module Control Actions

Context for Control Request	Guidewords for Unsafe Control Actions		
Vehicle operational mode confirmation request before implementing vehicle control	Provided but is delivered incorrectly	Provided but too late	Not provided but needed
Engage autonomous driving mode	PH1, PH2	PH1, PH5	PH1, PH5
Engage manual control mode	PH1, PH2	PH1, PH5	PH1, PH5

Appendix C

WATonoBus Operation Conditions Generation

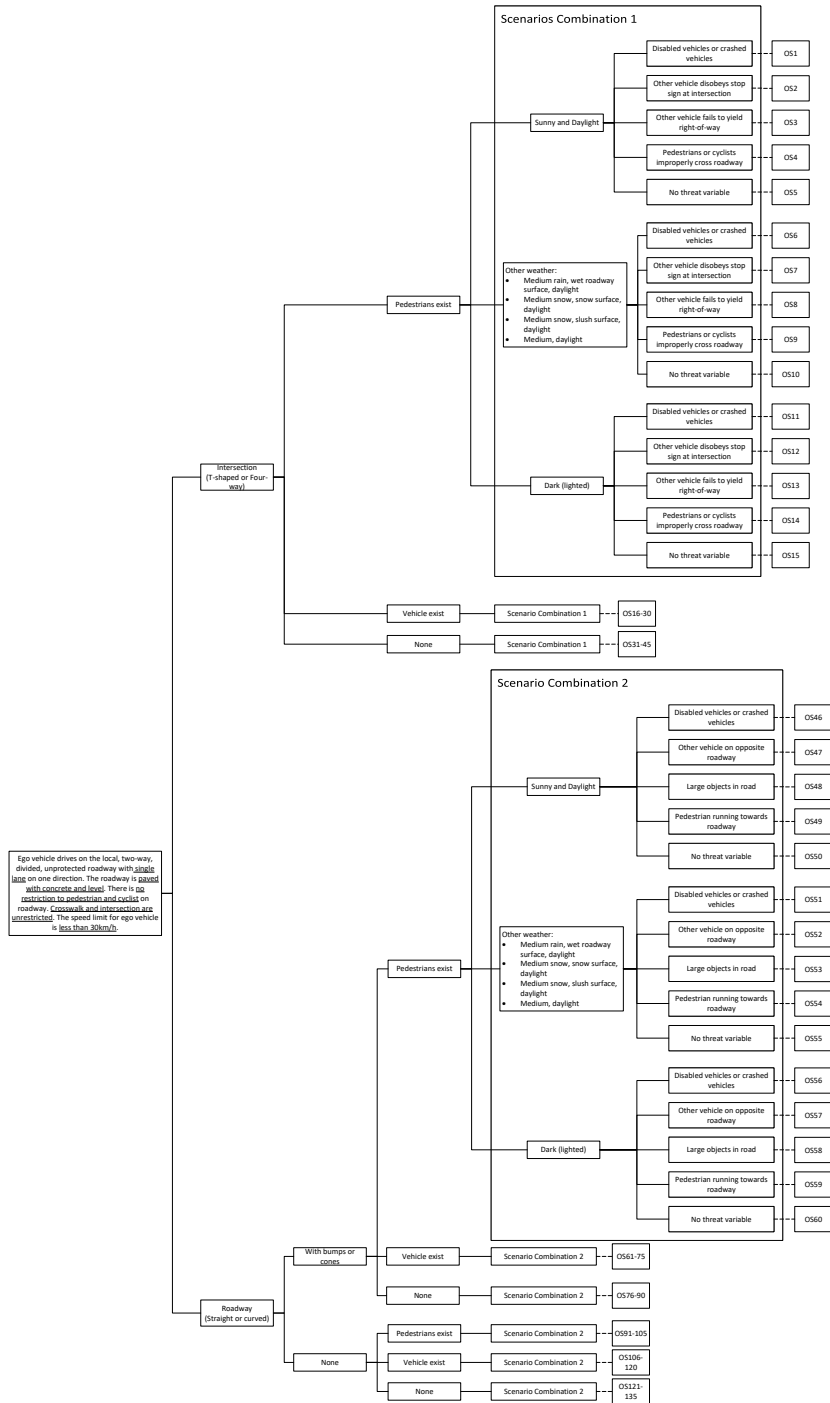


Figure C.1: Operation Conditions Summary of WATonoBus

Appendix D

ASIL Assessment Table

Table D.1: ASIL Assessment Table

Exposure Class				
E0	E1	E2	E3	E4
Incredible	Very low probability	Low probability	Medium probability	High probability
Severity Assessment				
S0	S1	S2	S3	S4
No injures	Light and moderate injuries	Severe and life-threatening injuries (survival probable)	Life-threatening injuries (survival uncertain), fatal injuries	N/A
Controllability Assessment				
C0	C1	C2	C3	C4
Control-lable in general	Simply controllable	Normally controllable	Difficult to control or uncontrollable	N/A

Appendix E

WATonoBus Primary Hazardous Condition ASIL Assessment

Table E.1: ASIL Assessment for PH1

Unable to adjust vehicle control autonomously							
OS	S	Judgement	E	Judgement	C	Judgement	ASIL
OS 1, 6, 11	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 2, 7, 12	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM
OS 3, 8, 13	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	A
OS 4, 9, 14	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 5, 10, 15	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	B
OS 16, 21, 26	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 17, 22, 27	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM
OS 18, 23, 28	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	QM
OS 19, 24, 29	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 20, 25, 30	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM

OS 31, 36, 41	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 32, 37, 42	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM
OS 33, 38, 43	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	QM
OS 34, 39, 44	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 35, 40, 45	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM
OS 46, 51, 56, 91, 96, 101	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 47, 52, 57, 92, 97, 102	S3	Ego vehicle collides with existing pedestrians. Ego vehicle collides with opposite vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 48, 53, 58, 93, 98, 103	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM
OS 49, 54, 59, 94, 99, 104	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 50, 55, 60, 95, 100, 105	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	B
OS 61, 66, 71, 106, 111, 116	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 62, 67, 72, 107, 112, 117	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 63, 68, 73, 108, 113, 118	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM
OS 64, 69, 74, 109, 114, 119	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 65, 70, 75, 110, 115, 120	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM
OS 76, 81, 86, 121, 126, 131	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 77, 82, 87, 122, 127, 132	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 78, 83, 88, 123, 128, 133	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM

OS 79, 84, 89, 124, 129, 134	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 80, 85, 90, 125, 130, 135	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM

Table E.2: ASIL Assessment for PH2

Unable to proceed safety fallback strategy							
OS	S	Judgement	E	Judgement	C	Judgement	ASIL
OS 1, 6, 11	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C3	Human operator can not take over vehicle control during emergency.	B
OS 2, 7, 12	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering aggressive vehicles at intersection	C3	Human operator can not take over vehicle control during emergency.	B
OS 3, 8, 13	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C3	Human operator can not take over vehicle control during emergency.	C
OS 4, 9, 14	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C3	Human operator can not take over vehicle control during emergency.	C
OS 5, 10, 15	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C3	Human operator can not take over vehicle control during emergency.	D
OS 16, 21, 26	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C3	Human operator can not take over vehicle control during emergency.	A
OS 17, 22, 27	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C3	Human operator can not take over vehicle control during emergency.	A
OS 18, 23, 28	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C3	Human operator can not take over vehicle control during emergency.	B
OS 19, 24, 29	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C3	Human operator can not take over vehicle control during emergency.	C
OS 20, 25, 30	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C3	Human operator can not take over vehicle control during emergency.	B
OS 31, 36, 41	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C3	Human operator can not take over vehicle control during emergency.	A
OS 32, 37, 42	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C3	Human operator can not take over vehicle control during emergency.	A
OS 33, 38, 43	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C3	Human operator can not take over vehicle control during emergency.	A

OS 34, 39, 44	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C3	Human operator can not take over vehicle control during emergency.	C
OS 35, 40, 45	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C3	Human operator can not take over vehicle control during emergency.	B
OS 46, 51, 56, 91, 96, 101	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C3	Human operator can not take over vehicle control during emergency.	B
OS 47, 52, 57, 92, 97, 102	S3	Ego vehicle collides with existing pedestrians. Ego vehicle collides with opposite vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C3	Human operator can not take over vehicle control during emergency.	B
OS 48, 53, 58, 93, 98, 103	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering large obstacles on campus roadway	C3	Human operator can not take over vehicle control during emergency.	B
OS 49, 54, 59, 94, 99, 104	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C3	Human operator can not take over vehicle control during emergency.	C
OS 50, 55, 60, 95, 100, 105	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C3	Human operator can not take over vehicle control during emergency.	D
OS 61, 66, 71, 106, 111, 116	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C3	Human operator can not take over vehicle control during emergency.	A
OS 62, 67, 72, 107, 112, 117	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C3	Human operator can not take over vehicle control during emergency.	B
OS 63, 68, 73, 108, 113, 118	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C3	Human operator can not take over vehicle control during emergency.	QM
OS 64, 69, 74, 109, 114, 119	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C3	Human operator can not take over vehicle control during emergency.	C
OS 65, 70, 75, 110, 115, 120	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C3	Human operator can not take over vehicle control during emergency.	B
OS 76, 81, 86, 121, 126, 131	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C3	Human operator can not take over vehicle control during emergency.	A
OS 77, 82, 87, 122, 127, 132	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C3	Human operator can not take over vehicle control during emergency.	B
OS 78, 83, 88, 123, 128, 133	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C3	Human operator can not take over vehicle control during emergency.	QM
OS 79, 84, 89, 124, 129, 134	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C3	Human operator can not take over vehicle control during emergency.	C
OS 80, 85, 90, 125, 130, 135	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C3	Human operator can not take over vehicle control during emergency.	B

Table E.3: ASIL Assessment for PH3

Incorrectly adjust vehicle control autonomously							
OS	S	Judgement	E	Judgement	C	Judgement	ASIL
OS 1, 6, 11	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 2, 7, 12	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM
OS 3, 8, 13	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	A
OS 4, 9, 14	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 5, 10, 15	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	B
OS 16, 21, 26	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 17, 22, 27	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM
OS 18, 23, 28	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	QM
OS 19, 24, 29	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 20, 25, 30	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM
OS 31, 36, 41	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 32, 37, 42	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM
OS 33, 38, 43	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	QM
OS 34, 39, 44	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 35, 40, 45	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM
OS 46, 51, 56, 91, 96, 101	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 47, 52, 57, 92, 97, 102	S3	Ego vehicle collides with existing pedestrians. Ego vehicle collides with opposite vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 48, 53, 58, 93, 98, 103	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM

OS 49, 54, 59, 94, 99, 104	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 50, 55, 60, 95, 100, 105	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	B
OS 61, 66, 71, 106, 111, 116	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 62, 67, 72, 107, 112, 117	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 63, 68, 73, 108, 113, 118	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM
OS 64, 69, 74, 109, 114, 119	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 65, 70, 75, 110, 115, 120	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM
OS 76, 81, 86, 121, 126, 131	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 77, 82, 87, 122, 127, 132	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 78, 83, 88, 123, 128, 133	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM
OS 79, 84, 89, 124, 129, 134	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 80, 85, 90, 125, 130, 135	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM

Table E.4: ASIL Assessment for PH4

Incorrectly implement safety strategies							
OS	S	Judgement	E	Judgement	C	Judgement	ASIL
OS 1, 6, 11	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 2, 7, 12	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering aggressive vehicles at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 3, 8, 13	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C2	Human operator can take over control after noticing ineffective safety strategies.	B
OS 4, 9, 14	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	B

OS 5, 10, 15	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	C
OS 16, 21, 26	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	QM
OS 17, 22, 27	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	QM
OS 18, 23, 28	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 19, 24, 29	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	B
OS 20, 25, 30	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	B
OS 31, 36, 41	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	QM
OS 32, 37, 42	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	QM
OS 33, 38, 43	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 34, 39, 44	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	B
OS 35, 40, 45	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 46, 51, 56, 91, 96, 101	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 47, 52, 57, 92, 97, 102	S3	Ego vehicle collides with existing pedestrians. Ego vehicle collides with opposite vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 48, 53, 58, 93, 98, 103	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering large obstacles on campus roadway	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 49, 54, 59, 94, 99, 104	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	B
OS 50, 55, 60, 95, 100, 105	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	C
OS 61, 66, 71, 106, 111, 116	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	QM

OS 62, 67, 72, 107, 112, 117	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 63, 68, 73, 108, 113, 118	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C2	Human operator can take over control after noticing ineffective safety strategies.	QM
OS 64, 69, 74, 109, 114, 119	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	B
OS 65, 70, 75, 110, 115, 120	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 76, 81, 86, 121, 126, 131	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	QM
OS 77, 82, 87, 122, 127, 132	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	A
OS 78, 83, 88, 123, 128, 133	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C2	Human operator can take over control after noticing ineffective safety strategies.	QM
OS 79, 84, 89, 124, 129, 134	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C2	Human operator can take over control after noticing ineffective safety strategies.	B
OS 80, 85, 90, 125, 130, 135	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C2	Human operator can take over control after noticing ineffective safety strategies.	A

Table E.5: ASIL Assessment for PH5

Slow reaction to surrounding environment							
OS	S	Judgement	E	Judgement	C	Judgement	ASIL
OS 1, 6, 11	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 2, 7, 12	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM
OS 3, 8, 13	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	A
OS 4, 9, 14	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 5, 10, 15	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	B
OS 16, 21, 26	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 17, 22, 27	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM

OS 18, 23, 28	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	QM
OS 19, 24, 29	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 20, 25, 30	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM
OS 31, 36, 41	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 32, 37, 42	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering aggressive vehicles at intersection	C1	Human operator can take over control immediately.	QM
OS 33, 38, 43	S2	Ego vehicle collides with same direction vehicles at low velocity	E3	Medium probability of encountering vehicle that misunderstand the right-of-way	C1	Human operator can take over control immediately.	QM
OS 34, 39, 44	S3	Ego vehicle collides with existing pedestrians	E3	Medium probability of encountering pedestrian/cyclist improper crossing at intersection	C1	Human operator can take over control immediately.	A
OS 35, 40, 45	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM
OS 46, 51, 56, 91, 96, 101	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 47, 52, 57, 92, 97, 102	S3	Ego vehicle collides with existing pedestrians. Ego vehicle collides with opposite vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 48, 53, 58, 93, 98, 103	S3	Ego vehicle collides with existing pedestrians.	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM
OS 49, 54, 59, 94, 99, 104	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 50, 55, 60, 95, 100, 105	S3	Ego vehicle collides with existing pedestrians.	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	B
OS 61, 66, 71, 106, 111, 116	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 62, 67, 72, 107, 112, 117	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 63, 68, 73, 108, 113, 118	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM
OS 64, 69, 74, 109, 114, 119	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 65, 70, 75, 110, 115, 120	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM

OS 76, 81, 86, 121, 126, 131	S2	Ego vehicle collides with same direction vehicles at low velocity	E2	Low probability of encountering disabled or crashed vehicles on campus	C1	Human operator can take over control immediately.	QM
OS 77, 82, 87, 122, 127, 132	S3	Ego vehicle collides with opposite direction vehicle with low velocity	E2	Low probability of encountering opposite vehicle in wrong lane on campus	C1	Human operator can take over control immediately.	QM
OS 78, 83, 88, 123, 128, 133	S1	Ego vehicle collides with large objects on roadway or nearby infrastructure	E2	Low probability of encountering large obstacles on campus roadway	C1	Human operator can take over control immediately.	QM
OS 79, 84, 89, 124, 129, 134	S3	Ego vehicle collides with existing pedestrians.	E3	Medium probability of encountering pedestrians rushing to cross roadway on campus	C1	Human operator can take over control immediately.	A
OS 80, 85, 90, 125, 130, 135	S1	Ego vehicle deviate from trajectory and collide with nearby infrastructure	E4	High probability of encountering no threat variable at intersection	C1	Human operator can take over control immediately.	QM