

THÈSE DE DOCTORAT
DOCTORATE THESIS

MULTI-RISK-AWARE
AUTONOMOUS NAVIGATION
UNDER UNCERTAINTY INDUCED
BY PERSONAL LIGHT ELECTRIC
VEHICLES

EMMANUEL ALAO

*Université de technologie de Compiègne
Heudiasyc, UMR CNRS 7253*

MEMBRES DU JURY

RAPPORTEURS	Emilia SILVAS	Associate Professor, Eindhoven University of Technology (TU/e Netherlands)
	Maan EL BADAoui EL NAJJAR	Professor, Université de Lille
EXAMINATEURS	Michel BASSET	Professor, Université Haute-Alsace
	Philippe BONNIFAIT	Professor, Université de Technologie de Compiègne
	Jean-Christophe POPIEUL	Professor, Université Polytechnique Hauts-de-France
	Anne SPALANZANI	Professor, Université de Grenoble Alpes
CO-DIRECTEURS	Lounis ADOUANE	Professor, Université de Technologie de Compiègne
	Philippe MARTINET	Research Director, Centre Inria d'Université Côte d'Azur

5 DECEMBER 2025

SPÉCIALITÉ: Automatic control and Robotics
ÉCOLE DOCTORALE: ED71 « Sciences pour l'Ingénieur »



ABSTRACT

Modern transportation is being transformed by autonomous vehicles (AVs), which promise enhanced safety, efficiency, and convenience. These vehicles rely on complex algorithms to perceive their environment, make decisions, and control motion in real time. A critical component is the risk assessment and management unit, which analyzes traffic scenarios for potential hazards and determines the safest control outputs. This process requires predicting and simulating multiple road scenarios, leading to various hypotheses and algorithms for estimating the future states of surrounding road users.

The emergence of Personal Light Electric Vehicles (PLEVs), such as electric scooters and bikes, has increased uncertainty on the road. These vehicles exhibit multi-modal behavior—moving like pedestrians at low speeds and accelerating rapidly—and their small size complicates detection. Two main challenges arise: uncertainty in predicting the states (position, velocity, size) of surrounding agents, and managing multi-modal motion predictions without overly conservative control actions. This PhD thesis, conducted under the ANNAPOLIS (AutoNOMous Navigation Among Personal mObiLity devIceS) project, addresses these challenges to ensure safe and comfortable AV navigation in dynamic and constrained urban environments. The research aims to design and validate a Multi-Risk Assessment and Multi-level Motion Optimization & Control (MiRA-MOC) architecture as a comprehensive multi-risk assessment and management system.

Specifically, the thesis addresses autonomous driving challenges posed by uncertainties in the dynamic, possibly erratic motion of surrounding agents, including PLEVs. It introduces a decision-making and control strategy based on a multi-level motion optimization framework that handles motion uncertainties using a Fusion of stochastic Predictive Inter-Distance Profiles (F-sPIDP). This continuous multi-risk assessment metric projects uncertainty in agents' motion onto the predicted inter-distance between the AV and its surroundings.

To address the first challenge, F-sPIDP extends the Predictive Inter-Distance Profile (PIDP) concept to a stochastic version (sPIDP) that captures uncertainty in predicted states. The second challenge in multi-modal prediction is handled by fusing multiple sPIDPs. An optimal trajectory is then selected from possible maneuvers using a combination of safe global trajectory sampling and F-sPIDP. Finally, local control optimization computes control actions that minimize collision risk while respecting vehicle dynamics. The proposed strategy has been validated through extensive simulations and real-vehicle experiments.

RÉSUMÉ

Le transport moderne est transformé par les véhicules autonomes (VA), qui promettent une meilleure sécurité, une efficacité accrue et une plus grande commodité. Ces véhicules utilisent des algorithmes complexes pour observer leur environnement, prendre des décisions et contrôler leur mobilité en temps réel. Un élément clé de ces algorithmes est constitué des unités d'évaluation et de gestion du risque, qui analysent les scénarios de circulation afin de détecter les dangers potentiels, notamment vis-à-vis des autres usagers, et de calculer les meilleures commandes pour une navigation sûre. Cela nécessite de prévoir et de simuler de multiples scénarios routiers, ce qui a conduit au développement de nombreuses hypothèses et algorithmes de prédiction pour estimer les états futurs des usagers.

Cette incertitude s'est accentuée avec l'introduction des Véhicules Personnels Légers Électriques (PLEVs), comme les trottinettes et vélos électriques. Ces dispositifs peuvent présenter des comportements multimodaux, se déplacer à basse vitesse comme des piétons puis accélérer soudainement, et leurs petites tailles compliquent leur détection. Deux défis majeurs se présentent : l'incertitude liée à l'état prédit (position, vitesse, taille) des agents environnants, et la difficulté de gérer les prédictions multi-modales sans adopter une approche trop conservatrice. Cette thèse de doctorat, menée dans le cadre du projet ANNAPOLIS (AutoNomous Navigation Among Personal mObiLity devIceS), s'attaque à ces scénarios complexes pour garantir une navigation sûre, confortable et pas excessivement conservatrice dans des environnements urbains dynamiques et contraints. L'objectif principal est de proposer et de valider une architecture globale de contrôle/commande (intitulée MiRA-MOC), permettant d'évaluer simultanément plusieurs risques possibles et agir en conséquent via des processus d'optimisation et de contrôle multi-niveaux, constituant la partie management du risque de MiRA-MOC.

Plus précisément, cette thèse aborde les défis de la conduite autonome liés aux incertitudes du mouvement dynamique et parfois erratique des agents environnants, y compris les PLEVs. Elle introduit une stratégie de décision et de contrôle fondée sur une méthode d'optimisation du mouvement multi-niveaux, capturant ces incertitudes via une Fusion of stochastic Predictive Inter-distance Profile (F-sPIDP). En utilisant le F-sPIDP comme métrique continue d'évaluation multi-risques, les incertitudes du mouvement sont projetées sur la distance inter-prévisionnelle entre le VA et son environnement. Pour répondre au premier défi, le F-sPIDP étend le Predictive Inter-Distance Profile (PIDP) à une version stochastique (sPIDP) intégrant l'incertitude dans l'état prédit des agents. Le second défi, la prédiction multi-modale, est traitée par la fusion de plusieurs sPIDPs. Une trajectoire optimale est ensuite sélectionnée parmi les manœuvres possibles grâce à la combinaison de l'échantillonnage de trajectoires sûres et du F-sPIDP. Enfin, les actions de contrôle minimisant le risque de collision et

respectant la dynamique du VA sont calculées par une optimisation du contrôle local. La stratégie a été validée par des simulations intensives et des essais sur un véhicule autonome réel.

PUBLICATIONS

INTERNATIONAL PUBLICATIONS

1. E. Alao, L. Adouane, and P. Martinet, "Reliable multi-level optimization for safe predictive control of autonomous vehicles to avoid uncertain multimodal plevs," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 19th-25th October 2025, Hangzhou, China.
2. E. Alao, L. Adouane, and P. Martinet, "Stochastic and safe multi-risk fusion for autonomous navigation in the presence of plevs," 36th IEEE Intelligent Vehicles Symposium (IV), 22nd-25th June 2025, in Cluj-Napoca, Romania.
3. E. Alao, L. Adouane, and P. Martinet, "Multi-Priority-Based Strategy for Risk Assessment and Management in the Presence of Multiple Personal Light Electric Vehicles", Springer Nature Computer Science, 2025.
4. E. Alao, L. Adouane, and P. Martinet, "Hybrid optimization method for safe autonomous navigation under uncertainty," 12th IFAC-Intelligent Autonomous Vehicles (IFAC-IAV), 7th-9th May 2025, in Phoenix, Arizona, USA.
5. E. Alao, L. Adouane, and P. Martinet, "Multi-risk assessment and management in the presence of personal light electric vehicles," 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO), 18th-20th November 2024, in Porto, Portugal. **Awarded: Best student paper.**
6. E. Alao, L. Adouane, and P. Martinet, "Reliable risk assessment and management using probabilistic fusion of predictive inter-distance profile for urban autonomous driving," in 22nd European Control Conference (ECC), 25th-28th June 2024, in Stockholm, Sweden.

OTHER CONGRESS AND WORKSHOPS

1. E. Alao, L. Adouane, and P. Martinet, "Safe Autonomous Navigation under Uncertainty: A Multi-Risk Control Optimization Framework," 2nd Safety of Intelligent and Autonomous Vehicles: Formal Methods vs. Machine Learning approaches for reliable navigation (SIIV-FM2L) workshop, at IROS2025, 20th October 2025, in Hangzhou, China.
2. E. Alao, L. Adouane, and P. Martinet, "A Multi-Risk Assessment and Management Approach for Autonomous Driving in the Presence of Uncertainties introduced by PLEVs," CPER Recherche et Innovation en Transports et Mobilité Eco-responsables et Autonomes (CPER RITMEA) workshop, 28th June 2024, in Compiègne, France.
3. E. Alao, L. Adouane, and P. Martinet, "Reliable Risk Assessment and Management Architecture for Autonomous navigation in Urban Centers," International Symposium on the Verification of Autonomous Mobile Systems (VAMS), 9th-10th March 2023, in Paris, France.

*For wisdom will enter your heart, and knowledge will be pleasant to your soul,
Discretion will protect you, and understanding will guard you.*

— Proverbs 2:10-11

ACKNOWLEDGMENTS

This thesis marks the culmination of a long and challenging journey, and I owe its completion to the support, guidance, and encouragement of many people.

I want to express my heartfelt gratitude to my thesis supervisors, Lounis Adouane and Philippe Martinet, for their invaluable guidance, encouragement, and advice throughout this journey. Their broad knowledge, patience, and constant support, especially during the most challenging times, have been a great source of strength and inspiration to steer this thesis in the right direction. Knowing that they were always there to back me up gave me the confidence to easily navigate the toughest times and move forward.

This thesis was carried out as a collaboration between Heudiasyc UMR-CNRS 7253/UTC and Acentauri/Inria, funded by the ANR Annapolis project. I am deeply thankful to all the members of both laboratories for their support, collaboration, and encouragement. Working alongside such talented and dedicated people has been an enriching experience.

I would also like to extend my sincere thanks to the jury members for taking the time to read and evaluate my manuscript. Their thoughtful feedback and genuine interest in my work are greatly appreciated. Special thanks go to Emilia Silvas and Maan El Badaoui El Najjar for being excellent rapporteurs for my thesis, and to Michel Basset, Philippe Bonnifait, Jean-Christophe Popieul, and Anne Spalanzani for being wonderful examiners.

A massive thank you goes to my family, my wife Gloria Alao, my son, my parents, and my siblings, for their endless love, understanding, and encouragement. Their care and belief in me, as they listened to my ramblings, celebrated the small victories, have kept me going through every step of this journey.

I would also like to thank my colleagues and friends in the office, Shan He, Kévin Bellingard, Hugo Pousseur, Armando Miranda Moya, Thibault Charmet, and Antoine Lima, for their support, collaboration, and the many enjoyable moments we shared. Thank you for the laughs, the help, the spontaneous debugging sessions, and the collective struggle against impossible deadlines. Sharing an office with you made this journey infinitely more fun.

Finally, and once again, I want to thank everyone at Heudiasyc and Acentauri, including students, engineers, and administrative staff, who supported me in both big and small ways. Whether you helped with an experiment, a form, a piece of advice, or a much-needed coffee break, you played a part in this adventure. Your help and kindness have meant a lot to me.

CONTENTS

1	General Introduction and Main Contributions	21
1.1	Overview	21
1.2	Motivation and Context	21
1.3	Positioning of the Work in the State of the Art	22
1.4	PLEVs and Problem Definition	23
1.5	Proposed Solutions and Contributions	23
1.6	Manuscript Outline	24
2	Autonomous Urban Driving: Architectures and Multi-modal Motion Prediction .	27
2.1	Introduction	27
2.2	Navigation Task in Urban Environments	27
2.3	Control Architectures for Autonomous Vehicles	30
2.3.1	Sense - Think - Act	30
2.3.2	Modular System Architectures	32
2.3.3	End-to-End System Architectures	33
2.4	Uncertainties in Agent Motion Prediction	35
2.4.1	Multi-modal Motion Prediction	36
2.4.2	Roadside Units (RSUs) for Augmented Perception	40
2.4.3	Benefits of RSU-assisted prediction	40
2.5	Conclusion	41
3	Planning and Decision-Making in Urban Environments	43
3.1	Introduction	43
3.2	Bridging perception outputs and low-level control	43
3.3	Global Trajectory Planning	44
3.3.1	Graph-based Global Trajectory Planning	45
3.3.2	Probabilistic and Sampling-based Global Trajectory Planner	47
3.3.3	Kinodynamic Search on Lattices Global Trajectory Planner	48
3.3.4	Optimization-based Global Trajectory Planner	50
3.3.5	Learning-based Global Trajectory Planner	50
3.4	Local trajectory planning and control	52
3.4.1	Trajectory Tracking Controllers (Classic and Geometric)	52
3.4.2	Frenet Frame-based Trajectory Planners	53
3.4.3	Velocity Space Reactive Controllers	53
3.4.4	Model-Based Optimal Controllers	54
3.4.5	Data-Driven and Learning-Based Controllers	55
3.5	Risk Assessment for Autonomous Driving Systems	55
3.5.1	Deterministic Risk Assessment Methods	55
3.5.2	Probabilistic Risk Assessment Methods	58
3.5.3	Learning-Based Risk Assessment Methods	59
3.5.4	Structured Framework for AV Risk Assessment Standards	61
3.6	Risk Management as a Dual of Risk Assessment	62
3.7	Conclusion	64
4	MiRA-MOC: Proposed Global Control Architecture	65
4.1	Introduction	65
4.2	Components of MiRA-MOC	66
4.3	Multi-modal Prediction Module	68
4.3.1	Problem Formulation	68
4.3.2	Role in the Architecture	69
4.4	Vehicle Trajectory Generation	69
4.4.1	Problem Formulation	70
4.4.2	Role in the Architecture	71
4.5	F-sPIDP Collision Risk Assessment	71
4.5.1	Problem Formulation	73

	4.5.2	Role in the Architecture	75
4.6		Optimal Trajectory Selection	75
	4.6.1	Problem Formulation	76
	4.6.2	Role in the Architecture	76
4.7		Local Control Optimization	76
	4.7.1	Problem Formulation	77
	4.7.2	Role in the Architecture	77
4.8		Conclusion	78
5		F-PIDP for Multi-Risk Assessment and Management	79
	5.1	Introduction	79
	5.2	Fusion of Predictive Inter-Distance Profile (F-PIDP)	80
	5.2.1	Computation of each PIDP	81
	5.2.2	Parameterization of each PIDP	81
	5.2.3	Probabilistic Center of Mass	82
	5.2.4	PIDP fusion using probabilistic weights	83
	5.2.5	PIDP Setpoint based on Safety Distance	84
	5.3	F-PIDP in Stochastic Model Predictive Control (SMPC)	85
	5.3.1	SMPC for Risk Management	85
	5.3.2	Road Boundary Constraint	86
	5.3.3	Control Input Constraint	87
	5.3.4	F-PIDP+SMPC Simulation Results and Discussion	87
	5.3.5	Scenario 1: Jaywalking PLEV Switching from Slow to Fast motion	87
	5.3.6	Scenario 2: PLEV overtaking a parked vehicle	91
	5.4	The Curse of Dimensionality in Predictive Control	92
	5.4.1	Priority-based Strategy for Risk Management	93
	5.4.2	Selection of Most Dangerous PLEV	94
	5.5	Simulation Results for Scenario with a Prioritized Agent	95
	5.5.1	Result using Standard MPC formulation	96
	5.5.2	Result using Priority-based F-PIDP+MPC	97
	5.6	Conclusion	99
6		F-sPIDP and Multi-Level Motion Optimization	101
	6.1	Introduction	101
	6.2	Stochastic PIDP (sPIDP) to deal with motion estimation uncertainty	102
	6.2.1	Uncertainty-based Motion Prediction of PLEV	103
	6.2.2	Deterministic Component of sPIDP	104
	6.2.3	Stochastic Component of sPIDP	104
	6.2.4	Fusion of Stochastic PIDP (F-sPIDP) for multi-modal Agents Multi-Risk Assessment	105
	6.2.5	F-sPIDP in Uncertainty-aware MPC (U-MPC)	106
	6.2.6	Definition of U-MPC Objective Function	107
	6.2.7	Uncertainty-Based Safety Constraint	108
	6.2.8	Simulation and Results using F-sPIDP + SMPC	109
	6.2.9	Chance Constraint Risk Factor Test	110
	6.2.10	Statistical Comparison with SMPC	111
	6.3	Multi-level Motion Optimization and Control	112
	6.4	Sampling-based Safe Optimal Trajectory Selection	114
	6.4.1	Lateral Trajectory Generation	114
	6.4.2	Longitudinal Trajectory Generation	115
	6.5	F-sPIDP for Optimal Safe Trajectory Selection	116
	6.5.1	Sampling Objective Cost Function	117
	6.5.2	Uncertainty-aware Collision Check	117
	6.6	Local Control Optimization	119
	6.6.1	Control Objective and Cost Function	120
	6.6.2	Control Constraints	120
	6.6.3	Overall Control Optimization Problem	121
	6.6.4	Multi-priority-based Collision Avoidance Strategy	121

6.7	Results and Discussion	122
6.7.1	Comparison between F-PIDP and F-sPIDP using Multi-level Motion Optimization	122
6.7.2	Scenario with Multiple Priority PLEVs	123
6.7.3	Comparative Analysis of the proposed MiRA-MOC	124
6.8	Conclusion	127
7	General Conclusion and Prospects	129
7.1	Summary and Conclusions	129
7.2	Prospects	130
A	Simulation Environment	133
A.1	Introduction	133
A.2	Simulation Test Environment	133
A.2.1	The 2D Simulation Environment	133
A.2.2	State-of-the-Art Simulation Environments	134
A.2.3	Review of Simulation Platforms	134
A.2.4	Selected Simulation Platform: CARLA	135
B	Experimental Platform	137
B.1	Introduction	137
B.1.1	Robotized Vehicle Platform	137
B.1.2	Test Software Architecture	137
B.1.3	Implementation Results	138
	Bibliography	141

LIST OF NOTATIONS

- i Agent index, $i \in \{1, \dots, N_{agents}\}$.
- j Mode / trajectory index, $j \in \{1, \dots, N_{traj}\}$.
- t_0 Current time / prediction start.
- $t, T, t_{horizon}$ Time variable and prediction horizon (final prediction time).
- Δt Sampling time / prediction time step.
- $\mathcal{P} = \{p_1, \dots, p_{N_{traj}}\}$ Set of predicted trajectories (modes) for a PLEV.
- $\mathbf{x}_i(t) = (x_i(t), y_i(t), \theta_i(t))$ State of agent i (Cartesian position and heading).
- $\Pr(j)$ Posterior probability (mode weight) of trajectory p_j (often denoted μ_j).
- $\Sigma_i^j(t)$ Covariance matrix for trajectory j of agent i (positional and heading uncertainty).
- $\mathbf{r}(t) = (x(t), y(t))^\top$ Ego-vehicle position at time t .
- $\mathbf{r}_{p_j}(t)$ Position of the j -th predicted PLEV trajectory at time t .
- $\tilde{\mathbf{x}}_j(t)$ Estimated / predicted state for mode j .
- $A_j(t), C(t)$ State transition and measurement matrices.
- $w_j \sim \mathcal{N}(0, \Sigma_j^w)$ Mode/process noise (Gaussian) for mode j .
- $\varepsilon \sim \mathcal{N}(0, \Sigma^\varepsilon)$ Measurement noise (Gaussian).
- $\mathbf{p}_{ref}(s)$ Reference path parameterized by curvilinear coordinate s .
- s Curvilinear distance along the reference path.
- $\mathbf{n}_{ref}(s)$ Normal vector to the reference path at s .
- $\mathbf{s}_{AV}(t)$ Frenet coordinates of the AV.
- $s_{AV}(t), d_{AV}(t), \theta_{AV}(t)$ Frenet components for the ego vehicle.
- a_n, b_m Polynomial coefficients for $s_{AV}(t)$ and $d_{AV}(t)$ parameterizations.
- $\mathbf{x}_{AV}^k(t)$ Cartesian form of the k -th ego candidate trajectory.
- $\mathcal{X}_{traj} = \{\mathbf{x}_{traj}^1, \dots, \mathbf{x}_{traj}^M\}$ Set of M candidate ego trajectories.
- $\mathcal{X}_{safe}$ Filtered safe set of candidate trajectories after F-sPIDP risk assessment.
- $\mathbf{x}_{traj}(t)$ Candidate trajectory of the ego vehicle.
- $\mathbf{p}_{traj}(t)$ Position component of $\mathbf{x}_{traj}(t)$ (i.e., (x, y)).
- $d_j(t), d_{ij}(t)$ Inter-distance (Euclidean) between ego and trajectory p_j (or between ego and agent i 's j -th mode) at time t .
- $PIDP_j(t)$ Predictive Inter-Distance Profile for trajectory j .
- $sPIDP(t)$ Stochastic PIDP.
- $\mathbb{E}, \text{Var}$ Mean and variance of the stochastic PIDP.
- $d_w(t)$ Predicted uncertainty term used in safety distance.
- $pCOM(t)$ Probabilistic center of mass.
- $F_PIDP(t), F_sPIDP(t)$ Fusion of (stochastic) PIDP.

$F_{-}(s)PIDP_{ref}(t)$ F-PIDP or F-sPIDP reference/setpoint.

d_{safe} safety distance.

R Radius of the ego-vehicle (enclosing circle).

r radius of a PLEV (enclosing circle).

δ_v Braking distance term.

v_{max} Maximum considered speed.

$ETTC$ Extended Time-To-Collision.

$\mathcal{D}_{safe}$ Safety distance function.

t_{SNR} Time when Safety-Not-Respected (first time the inter-distance falls below safety threshold).

SR / SNR Safety-Respected / Safety-Not-Respected region.

P_{dmin} Agent with current minimum inter-distance to ego (closest agent).

P_{tSNR} Agent with minimum predicted t_{SNR} (earliest safety violation).

P_T Selected priority target (agent chosen for focused avoidance by priority strategy).

N_p / N_{agent} Number of surrounding PLEVs (agents) considered.

N Number of prediction steps.

M Number of ego trajectory samples.

$\mathbf{U} = (\mathbf{u}(t_0), \dots, \mathbf{u}(t_0 + N_c))$ Control sequence over the control horizon.

$\mathbf{u}(t)$ Control inputs (e.g., acceleration and steering angle).

$a_{min}, a_{max}, \delta_{min}, \delta_{max}$ Admissible control bounds.

$\mathcal{U}$ Set of admissible control inputs.

$f(\cdot, \cdot)$ Discrete-time vehicle dynamics (state-update map).

$\mathbf{w}(t)$ Process noise / disturbance.

$\mathcal{J}(\cdot)$ Primary MPC /SMPC cost function.

$\mathcal{J}_p(\cdot)$ PIDP-related penalty: cost for deviating from $F_{-}PIDP_{ref}$.

$J_{smooth}, J_{comfort}, J_{efficiency}$ Components of trajectory-selection cost.

w_s, w_c, w_e Weights for smoothness, comfort, efficiency.

$\mathcal{F}$ Multi-objective cost used in U-MPC / optimization.

f_x, f_{x_N}, f_u, f_p Sub-costs composing $\mathcal{F}$.

γ_i Weights for the $\mathcal{F}$ components.

α Risk/confidence parameter.

$\sigma(t)$ Tightening term computed from variance.

Σ_T Simulated position / tracking error magnitude.

LIST OF FIGURES

1.1	Users of PLEVs such as gyros, scooters, and bicycles exhibit various uncertain behaviors in an Urban environment, making navigation difficult for the autonomous vehicle	22
1.2	A scenario showing the evolution of the uncertainty of multiple PLEVs with multi-modal motion uncertainties, the candidate trajectories of the AV, the selected global safe trajectory, and the local control trajectory	24
2.1	The SAE Levels of Driving Automation, ranging from Level 0 (no automation) to Level 5 (full automation), define the progressive capabilities and responsibilities of autonomous vehicle systems in various driving scenarios [47]	28
2.2	Stanley: winner of the 2005 DARPA Grand Challenge [59]	29
2.3	The Sense → Think → Act paradigm structures autonomous driving systems into three primary layers: Perception/Localization, Planning and Decision-Making, and Control. Each layer processes information and passes it to the next, with feedback loops for refinement and error correction	32
2.4	The modular architecture of an autonomous vehicle system, illustrating the sequential flow from perception to control. Each module processes data and passes it to the next stage, with potential feedback loops for refinement [183]	33
2.5	The end-to-end architecture of an autonomous vehicle system, where raw sensor inputs are directly mapped to control commands through a deep neural network [183]	35
2.6	The hybrid architecture of an autonomous vehicle system, combining modular components with learning-based modules for specific tasks such as planning or trajectory generation [183]	36
2.7	Illustrative physics-based prediction methods: (a) a constant dynamics model, (b) an attraction–repulsion model capturing simple intentions, and (c) an interacting multiple model filter (IMMF) with several dynamic modes [186], (d) limitation of physics-based models for long-term predictions [193]	38
2.8	Maneuver-based prediction methods: a Hidden Markov Model (HMM) classifying maneuvers (lane change, moving straight) to predict trajectories [126]	39
2.9	Learning-based prediction methods: (a) Social-LSTM with social pooling to model interactions between agents [3], (b) SoPhie: An Attentive GAN for predicting paths compliant to social and physical constraints [188], (c) Graph-structured recurrent model capturing interactions and multi-modal trajectories [191]	40
3.1	Graph-based global trajectory planning methods: (a) A* search on a weighted graph, (b) improved A* search with smaller search expansion [237]	45
3.2	Probabilistic and sampling-based global trajectory planning methods: Rapidly-exploring Random Trees (RRT) (left) and Improved RRT (right) building a tree via random sampling [220]	47
3.3	Probabilistic and sampling-based global trajectory planning methods: Probabilistic Roadmaps (PRM) with 60 nodes (left) and 150 nodes (right) [145]	48
3.4	State Lattice with motion primitives [169]	49
3.5	Vehicle trajectory in Frenet Frame [166]	53
3.6	TTC in co-linear navigation scenario (Left) and ETTC in 2D navigation scenario (Right) [244]	58
3.7	AV Safety Standards by the IEEE [110]	61
4.1	Global architecture for Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) in autonomous vehicles. The system integrates perception, multi-modal motion prediction, multi-risk assessment using Fusion of stochastic PIDP (F-sPIDP), sampling-based planner, and predictive controllers to ensure safe and efficient navigation in mixed traffic environments	67

4.2	Multi-modal trajectory prediction	69
4.3	Predictive Inter-Distance Profile (PIDP) for risk assessment between vehicles (Cars) [103]	72
4.4	Ego vehicle and a Jaywalking PLEV with multi-modal trajectories: this implies multiple PIDPs for each agent and often leads to conservative behaviors	74
4.5	Multimodal predictions with state estimation uncertainties	75
5.1	Ego vehicle and a Jaywalking PLEV with multi-modal trajectories: this implies multiple PIDPs for each agent and often leads to conservative behaviors	80
5.2	Fusion of PIDPs (F-PIDP) of an AV and a PLEV with three 3 possible multi-modal trajectories	81
5.3	Scenario 1: (a) Velocity plot showing a smooth linear velocity and steering (b) Probability plot with a switching from Traj 3 (blue) to Traj 2 (green) after $t = 3.9s$	87
5.4	Scenario 1 at $t = 0.9s$ (a) Jaywalking PLEV moving slowly (b) Inter-distance plot of ego-vehicle and PLEV: The F-PIDP _{ref} plot exhibits a convex curve, indicating current safety is satisfactory. However, the steep curve suggests a high risk of future safety distance violations due to PIDP ₂	88
5.5	Scenario 1 at $t = 4.5s$ (a) Jaywalking PLEV moving Fast (b) Inter-distance plot of ego-vehicle and PLEV: As the vehicle moves closer to the PLEV, the F-PIDP set-point takes a concave form even before Traj 2 enters the collision zone d_{col} .	89
5.6	The overall inter-distance plot of ego-vehicle and PLEV for scenario 1: shows the region where Safety is Not Respected (SNR) when the PLEV changes its trajectory from Traj 3 to 2	90
5.7	(a) Scenario 2: PLEV overtaking a parked vehicle at $t = 4.9s$ (b) Inter-distance plot of ego-vehicle and PLEV overtaking a stationary truck: the F-PIDP plot shows a concave curve that encourages escaping from the risky scenario	90
5.8	Scenario 2: (a) Velocity plot showing a smooth linear velocity and steering (b) Probability plot with increasing probability for Traj 2 (green)	91
5.9	Inter-distance plot showing the overall PIDP: collision is avoided and the vehicle is able to move into the Safety Respected region (SR)	91
5.10	Priority-based target selection (a) Example of scenario with multiple PLEVs (b) F-PIDP plot of AV and multiple PLEVs: Priority-Based Target (P_T):= PLEV3 ($P_{dmin} < d_{safe}$) else (P_T):= PLEV1 the minimum ($tSNR_1 < tSNR_2$)	94
5.11	A scenario using MPC showing an AV (in red) and multi-modal PLEVs (1 in Yellow & 2 in Blue): (a) At $t = 1 s$, the actual speed of both PLEVs is 2 m/s, (b) At $t = 7.5 s$, AV overtakes PLEV1 but stays beside PLEV2 to avoid a possible collision with a possible Left turn trajectory of PLEV2	97
5.12	PIDP plot between the AV and the PLEVs using MPC: (a) & (c) At $t = 1 s$, all PIDPs (red, blue, green) of PLEV1 & 2 are above safety distance d_{safe} , (b) & (d) At $t = 7.5 s$, the PIDPs of PLEV1 are above d_{safe} while a PIDP of PLEV2 may lead to collision, reproduced from [4]	98
5.13	At $t = 3 s$, MPC violates the boundary constraint due to the sudden change in speed of PLEV	98
5.14	A scenario using the F-PIDP+MPC reproduced from [4]: (a) At $t = 1 s$, AV overtakes the priority target PLEV1 (in red), (b) then at $t = 5 s$, AV overtakes the new priority target PLEV2 (in red)	99
5.15	PIDP plot between the AV and the PLEVs using F-PIDP+MPC: (a) At $t = 1 s$, F-PIDP setpoints suggest creating more distance between AV and PLEV1, (b), (c), & (d) All PIDPs are above d_{safe} as presented in [4]	100
6.1	Multimodal predictions with state estimation uncertainties	102
6.2	(a)&(b) Estimated PIDP shows safety distance is respected, while the Actual/True PIDP shows safety distance is not respected (c)&(d) Representing estimation uncertainties using stochastic PIDP (sPIDP)	103
6.3	Fusion of stochastic PIDPs (F-sPIDP) of an AV and a PLEV with three (3) possible multi-modal maneuvers and their uncertainties (variance)	105
6.4	Test Scenario of the AV and a single PLEV crossing the road: showing the different trajectories of the AV for Varying Risk Factor $\alpha \in [0.5, 1]$	109

6.5	Varying Risk Factor $\alpha \in [0.5, 1]$: (a) Velocity, (b) Inter-Distance, and (c) Safety Distance	110
6.6	Scenario with multiple agents: Left, shows the 2D scenario of the AV trying to avoid collision with the surrounding vehicles. Right, shows the plot of the estimated sPIDP for each agent and their corresponding variance ($V[sPIDP]$), all stochastic PIDP above d_{safe}	110
6.7	Final Position of AV at Final Time N for both the proposed U-MPC and SMPC methods with tracking error of 1m: SMPC violates the constraint by moving too close to the boundary, U-MPC maintains a safe distance without being too conservative	112
6.8	An overview of the multi-level motion optimization steps: (Left) Sampling-based Optimal Trajectory Selection, (Right) Local Trajectory Optimization	113
6.9	Global path in Frenet coordinates	113
6.10	The candidate trajectories and the corresponding F-sPIDP between an AV and the possible trajectories of an agent: each F-sPIDP curve represents the fusion of inter-distances between AV's candidate trajectory and all the maneuvers of the PLEV	119
6.11	The local trajectory of the AV while respecting the uncertainty constraint $V[sPIDP(t)]$	120
6.12	Comparison between F-PIDP and F-sPIDP: (a) the current state of the agents, the predicted motion of the AV, and the predicted multi-modal motion of the PLEV, (b) the PIDPs, F-PIDP, F-sPIDP, and estimated inter-distance uncertainty plot, showing possible collision with PLEV (all PIDPs $< d_{safe}$), (c) & (d) sampled trajectories for varying levels of uncertainty, where $F-PIDP_{ref}$ is invariant to the level of uncertainty	123
6.13	A scenario using the proposed multi-priority collision avoidance strategy: (a) At $t = 0$ s, AV overtakes the priority targets PLEV 1 (on the same lane as AV) and vehicle, V1 (on the adjacent lane), (b) then by $t = 4$ s, AV overtakes the new priority target PLEV 2 (in red)	124
6.14	A scenario using the proposed multi-priority collision avoidance strategy: (a) At $t = 1.5$ s, AV cannot overtake the priority targets PLEV 1 and V1 (in red), (b) predicted F-PIDP and PIDP of the PLEVs on the inter-distance plot indicates possible collision, (c) then at $t = 6.0$ s, the AV is able to complete the lane change maneuver, (d) with the inter-distance plot showing no predicted collision (all PIDP are above d_{safe})	125
6.15	Comparison of the trajectory performance: (a) Baseline results in oscillatory AV motion, excessively influenced by the PLEVs' uncertain behavior, (b) Proposed method shows little oscillation by generating collision-free trajectories that respect desired lane-changing and lane-keeping maneuvers while accounting for uncertainties in the PLEV motion	126
6.16	Minimum inter-distance between AV and PLEV over time for varying level of uncertainty $\Sigma_T = [0, 1]m$: (a) Baseline showing occasional drops below ($d_{safe} = 3$ m), (b) Proposed method maintaining clearance up to 1 m above desired d_{safe}	127
A.1	A snapshot of the 2D simulation environment	134
B.1	Software Architecture for Real test of the proposed MiRA-MOC architecture	138
B.2	Visualization of Real Autonomous Vehicle (available in Heudiasyc Lab) and Virtual Environment (cf. video shown through this link https://youtu.be/zcYIv6yFLZ0?list=PL0zhSFDHifx2ruWjKJ1ggbCKfnzertLLu)	139

GENERAL INTRODUCTION AND MAIN CONTRIBUTIONS

This chapter presents the motivation for this work by examining the rise of Personal Light Electric Vehicles (PLEVs) in cities and the challenges they pose for autonomous navigation. It defines the core problem of safe navigation, the main contributions, and outlines the chapters of this thesis.

1.1 OVERVIEW

This chapter introduces the motivation and context of this research, highlighting the growing impact of Personal Light Electric Vehicles (PLEVs) on urban mobility and the challenges they pose for autonomous vehicles. It outlines how unpredictable behavior of PLEV users, combined with evolving traffic regulations, creates complex decision-making problems for autonomous systems with limited onboard perception. The chapter then defines the core problem of ensuring safe and smooth navigation amid these uncertainties. Finally, it presents an overview of the proposed solutions and situates the work within the state of the art of autonomous navigation, focusing on advancing autonomous navigation for emerging personal mobility devices.

1.2 MOTIVATION AND CONTEXT

Autonomous driving systems promise a future of safer roads, reduced traffic congestion, lower emissions, and more efficient use of urban space, potentially freeing up areas currently dominated by cars for pedestrians, cyclists, and green spaces. These systems have the potential to drastically reduce accidents caused by human error, optimize traffic flow through intelligent coordination, and support the transition toward more sustainable cities.

However, the increasing invasion of urban centers by new multi-modal transport solutions is directly or indirectly the source of unpredictable behaviors in the traffic environment. These new multi-modal transport solutions are referred to as Personal Light Electric Vehicles (PLEVs), including gyropods, monowheels, hoverboards, and electric scooters. PLEVs are likened to pedestrians or bicycles; hence, they are expected to use dedicated lanes and are subjected to the same cycling rules. However, the "Mobility Law 2019" provides for the return of scooters to traffic lanes when dedicated bicycle lanes do not exist. In such a context, autonomous vehicles suffer from limited perception, obtained only from onboard sensors, which makes decision-making and control difficult. In such a situation, unforeseen and unexpected events stem from the presence of new electrical mobility systems or from behaviors of unstable pedestrians using (or not) new PLEV and respecting (or not) the traffic rules. The study of decision-making in

such uncertain conditions has gained considerable attention. However, many existing methods tend to be computationally expensive, necessitate numerous assumptions about the problem structure, and often require substantial prior knowledge. Hence, their applicability to intricate real-world issues is limited due to the impracticality or impossibility of meeting all these requirements. Therefore, a significant challenge in the field of autonomous vehicles (AVs) is ensuring safe and smooth navigation with respect to all entities surrounding the AV (cf. Figure 1.1). This thesis is part of the project **ANNAPOLIS: AutoNomous Navigation Among Personal mObiLity devIceS**, ANR 2022-2025.



Figure 1.1: Users of PLEVs such as gyros, scooters, and bicycles exhibit various uncertain behaviors in an Urban environment, making navigation difficult for the autonomous vehicle

1.3 POSITIONING OF THE WORK IN THE STATE OF THE ART

This work focuses on the development of autonomous navigation systems capable of safely and efficiently navigating in environments populated by various personal mobility devices, especially PLEVs. The aim is to address the challenges posed by the unpredictable behaviors of these devices and their users, as well as the evolving traffic regulations that govern their use. We cover several key areas in autonomous navigation, including problems in perception, decision-making, and control strategies for autonomous vehicles. Highlighting perception problems in accurately detecting and classifying personal mobility devices in the environment, as well as predicting their multiple possible (or multi-modal) future trajectories. Roadside Units (RSUs), such as cameras and lidars, are also used to extend/enhance the perception capabilities of the autonomous vehicle (cf. Figure 1.1). This work focuses on developing a decision-making and control strategy for autonomous vehicles, specifically addressing the challenges posed by the presence of personal mobility devices in urban environments. The goal is to develop robust algorithms that ensure safe and efficient navigation while accounting for the uncertainties and estimation errors introduced by the multi-modal behavior of these devices and their users, without being overly conservative.

1.4 PLEVS AND PROBLEM DEFINITION

Autonomous vehicles (AVs) are expected to offer significant benefits, including reduced congestion, lower emissions, increased lane capacity, and improved safety. Nevertheless, the level of safety an AV provides depends on its ability to predict the motion of other road users accurately. This is a difficult problem, also known as the motion modeling and prediction problem [78], [83], [186]. Moreover, this problem is further escalated by the widespread use of Personal Light Electric Vehicles (PLEVs) [5], such as electric scooters [172]. PLEVs are grouped under the L-category vehicles in the European Union (EU), which is equivalent to the Sub A category in the United States (US) [187]. Similar to AVs, the goal of PLEVs in the EU is to reduce pollution and enable fast and easy daily commutes over short distances [187].

However, PLEVs are found to pose numerous safety challenges to pedestrians, cars, and other road users. This is because they possess various small sizes and speeds that are relatively between those of pedestrians and cars [5]. This results in multiple hypotheses on their future motion (cf. Figure 1.1 & 1.2).

- **Visibility and Classification:** This problem stems directly from the perception of the environment. Due to occlusion, sensor noise, and the small size of PLEVs, it is often hard to distinguish between a pedestrian and a PLEV user. A PLEV may be misclassified as a pedestrian with a low-speed profile. Additionally, most perception algorithms rely on data-driven approaches, yet there is a lack of training datasets that clearly differentiate between pedestrians, PLEVs, and similar entities.
- **Speed Variability:** Unlike pedestrians, PLEVs exhibit higher speed profiles, making them more hazardous. This becomes very disastrous when combined with the classification issue. Meanwhile, constantly accounting for uncertainty in the speed of surrounding agents can lead to overly cautious behavior in autonomous systems.
- **User Behavior:** PLEV users often display erratic behavior, whether intentional or accidental. For example, a user may abruptly accelerate or decelerate, or lose balance and fall. These unpredictable actions make it unreliable to depend solely on a single motion prediction.

Therefore, it is crucial to assess collision risks and implement effective multi-risk management strategies. This ensures that an autonomous vehicle's motion remains safe and less conservative, while maintaining passenger comfort.

1.5 PROPOSED SOLUTIONS AND CONTRIBUTIONS

The proposed decision-making and control strategy employs a Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) method to address the above problems. More specifically, below are the key contributions of this work:

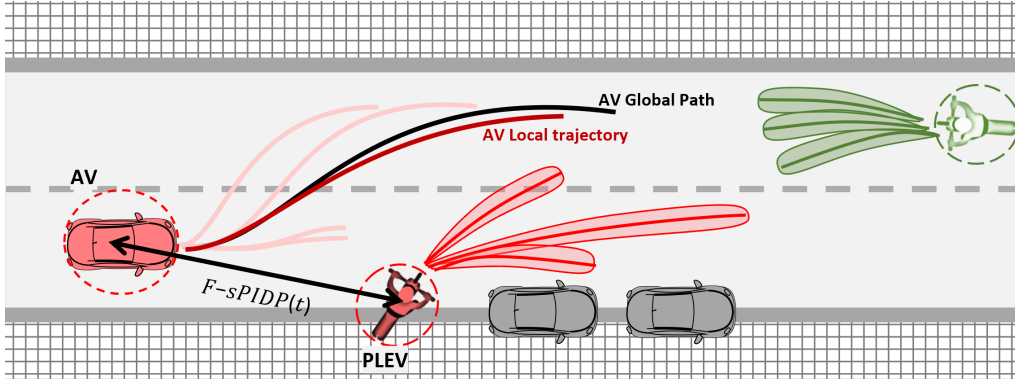


Figure 1.2: A scenario showing the evolution of the uncertainty of multiple PLEVs with multi-modal motion uncertainties, the candidate trajectories of the AV, the selected global safe trajectory, and the local control trajectory

- A **Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) architecture** is proposed to combine multi-risk assessment and sampling-based planning with local control refinement, enabling robust and smooth motion under uncertainty.
- The **Fusion of stochastic Predictive Inter-Distance Profile (F-sPIDP)** is integrated into the motion planning process, extending the concept of Predictive Inter-Distance Profile (PIDP) [5], [17], [102] to enable a structured assessment of the stochastic uncertainties and multi-modal behaviors of surrounding agents.
- A **sampling-based optimization planner** is developed to evaluate candidate trajectories using F-sPIDP-derived safety constraints. This method allows sampling multiple possible solutions, unlike the single solution most state-of-the-art algorithms provide.
- And a subsequent **local control optimization method** is presented to refine the selected trajectory by generating dynamically feasible control actions that remain consistent with the underlying uncertainty profile.
- The **overall multi-level framework (MiRA-MOC) is validated through simulations and experimentation**, demonstrating enhanced safety, robustness, and motion smoothness under varying levels of uncertainty.

1.6 MANUSCRIPT OUTLINE

This manuscript is outlined as follows:

- **Chapter 1:** presents the motivation of the work and an overview of the problem being addressed
- **Chapter 2:** highlights the state-of-the-art methods, introducing various control architectures, emphasizing multi-modal motion prediction for PLEVs.

- **Chapter 3:** presents planning, risk assessment & management, and control methodologies in the literature.
- **Chapter 4:** gives an overview of the proposed global architecture, Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC), for multi-risk assessment and management.
- **Chapter 5:** develops and discusses the Fusion of Predictive Inter-Distance Profiles (F-PIDP). This framework enables the continuous assessment of collision risk among multiple vehicles.
- **Chapter 6:** presents Fusion of stochastic PIDP (F-sPIDP), which extends F-PIDP, and the proposed strategy MiRA-MOC, detailing the methodology for managing collision risk.

Finally, **Chapter 7** provides a global conclusion of the proposed solution, summarizes the main contributions of this thesis, and discusses prospects.

The highlights of the ongoing experimentation of MiRA-MOC are presented in the **Appendix**.

AUTONOMOUS URBAN DRIVING: ARCHITECTURES AND MULTI-MODAL MOTION PREDICTION

This chapter reviews the advances of autonomous driving systems in urban environments. It outlines the challenges of complex traffic and the growing presence of Personal Light Electric Vehicles (PLEVs). System architectures—modular, end-to-end, and hybrid—are compared for safety and reliability. Key motion prediction methods and the impact of uncertainty on perception, planning, and control are discussed, emphasizing that managing uncertainty is essential for robust urban autonomy.

2.1 INTRODUCTION

Autonomous vehicles (AVs) promise transformative benefits in terms of safety, environmental impact, and transportation efficiency. First and foremost, AVs can dramatically enhance road safety by eliminating the human error that underlies most traffic accidents. For instance, [116] emphasizes that approximately 94% of crashes stem from human factors, and through automation, AVs have the potential to significantly reduce both collisions and driving variability, thereby reducing driver stress and increasing comfort [116]. In real-world deployments, Waymo vehicles recorded an 88% decrease in property-damage claims and a 92% reduction in bodily-injury claims compared to human-driven vehicles, underscoring the tangible safety gains of AVs [90].

Beyond safety, autonomous technology also offers notable environmental and efficiency improvements. AV systems optimized for eco-driving have been shown to reduce operational emissions and fuel consumption, with certain studies reporting emissions reductions of 21.2% in operation due to more efficient driving. Though life-cycle manufacturing effects must also be considered [164]. Moreover, literature reviews confirm that widespread AV adoption could lead to reductions in vehicle ownership, congestion, and greenhouse gas emissions, along with increases in accessibility, mobility, and fleet efficiency [175].

Therefore, the following section discusses a brief history of AVs and their evolution over the years.

2.2 NAVIGATION TASK IN URBAN ENVIRONMENTS

The vision of autonomous driving has evolved over several decades, beginning with early research in the 1980s. Landmark projects such as Carnegie Mellon University's Navlab and Mercedes-Benz's VaMP demonstrated the feasibility of computer-driven vehicles under controlled environments, laying the foundation for modern developments [52], [171]. The 2004 and 2005 DARPA Grand

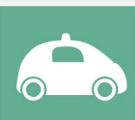
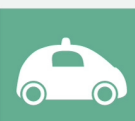
For on-road vehicles		 Human driver		 Automated system	
		Steering and acceleration/deceleration	Monitoring of driving environment	Fallback when automation fails	Automated system is in control
Human driver monitors the road	0 NO AUTOMATION				N/A
	1 DRIVER ASSISTANCE				SOME DRIVING MODES
	2 PARTIAL AUTOMATION				SOME DRIVING MODES
Automated driving system monitors the road	3 CONDITIONAL AUTOMATION				SOME DRIVING MODES
	4 HIGH AUTOMATION				SOME DRIVING MODES
	5 FULL AUTOMATION				

Figure 2.1: The SAE Levels of Driving Automation, ranging from Level 0 (no automation) to Level 5 (full automation), define the progressive capabilities and responsibilities of autonomous vehicle systems in various driving scenarios [47]

Challenges (cf. Figure 2.2) marked a turning point, catalyzing rapid progress in perception, planning, and control algorithms, and accelerating the involvement of industry leaders in automation research [211].

In parallel, Advanced Driver Assistance Systems (ADAS) emerged as a critical enabler of automated driving. Early ADAS functions, such as adaptive cruise control (introduced in the late 1990's by Mitsubishi and Mercedes-Benz), lane-keeping assistance, and automatic emergency braking, progressively enhanced vehicle safety and convenience [61]. These technologies not only improved driver support but also served as stepping stones toward higher levels of autonomy by integrating sensors, cameras, radar, LiDAR, and AI-based perception [203].

Today, the development of autonomous driving is structured around the SAE Levels of Autonomy (cf. Figure 2.1), which classify automation from Level 0 (no automation) to Level 5 (full automation in all conditions) [43]. Most commercially available systems fall into Levels 1 and 2, offering driver assistance features where human supervision remains mandatory. More advanced pilots, such as Tesla's Full Self-Driving Beta and Mercedes-Benz's Drive Pilot, operate at Level 2+ and



Figure 2.2: Stanley: winner of the 2005 DARPA Grand Challenge [59]

Level 3, where conditional automation allows the driver to disengage under limited circumstances [11]. Experimental prototypes and robotaxi services from companies like Waymo, Tesla, and Cruise continue to push toward Levels 4 and 5, where vehicles are capable of fully autonomous operation within geofenced or, eventually, unrestricted domains [30].

However, the rapid proliferation of micromobility devices, especially PLEVs such as e-scooters and e-bikes, poses challenges that may hinder these benefits:

First, PLEV users often exhibit unpredictable behavior and take on elevated risk, which complicates the perception and prediction systems of AVs. E-scooters, for instance, are frequently perceived as “fun” devices, leading to less cautious behavior in traffic [153]. A systematic review given in [217] corroborates this, pointing out the prevalence of risky behaviors among predominantly young, urban, highly educated users, and the consequent safety concerns. Authors in [31] similarly note that such behaviors contribute to negative public perception and vary widely across urban environments, complicating traffic modeling and AV response.

Second, the mixing of PLEVs and conventional vehicles in ambiguous infrastructure zones, such as sidewalks, bike lanes, and roadsides, creates spatial and regulatory ambiguity. [53] investigated vehicle and e-scooter interactions in a mini-roundabout scenario and highlighted how inconsistent positioning and speed behavior make it harder for autonomous vehicles to navigate safely and efficiently.

Above all, these unpredictable PLEV behaviors can undermine public trust and the perceived reliability of AV systems. If widespread accidents or unsafe behavior from PLEVs gives rise to negative media and public sentiment, AV adoption could suffer from collateral skepticism, a phenomenon that parallels concerns around other emerging mobility technologies [31].

Urban driving introduces a level of unpredictability that challenges conventional AV planning and control pipelines, which can be summarized as follows:

1. **Dynamic and Unstructured Environments:** constantly changing traffic patterns, jaywalking pedestrians, cyclists weaving through traffic, and PLEVs

(e-scooters, e-bikes) sharing road space create a highly dynamic environment that is difficult to model and predict.

2. Complex Infrastructure: narrow streets, irregular layouts, traffic lights, roundabouts, and roadworks complicate route planning and rule adherence.
3. High-Density Multi-Agent Interactions: AVs must safely navigate around pedestrians, cyclists, scooters, and aggressive drivers, all with heterogeneous dynamics and intentions.

2.3 CONTROL ARCHITECTURES FOR AUTONOMOUS VEHICLES

Control architectures define how the different functional components of an autonomous system: perception, planning/decision-making, and control, are structured and interact. Two dominant paradigms have emerged: modular system architectures and end-to-end learning approaches. While modularity provides interpretability and systematic validation, end-to-end methods promise efficiency and adaptability. Hybrid approaches attempt to combine the strengths of both.

2.3.1 *Sense - Think - Act*

The dominant design philosophy in AV development follows the Sense → Think → Act paradigm: Perception/Localization → Planning and Decision-Making → Control, a framework inherited from robotics [211]. This decomposition provides clear interfaces between functional layers, enabling both performance optimization and safety certification (cf. Figure 2.3).

a) SENSE: Perception and Localization

The perception subsystem integrates data from multiple sensors (lidar, radar, cameras, GNSS, IMU) to generate a representation of the environment. Core tasks include:

- Object Detection and Tracking: identifying and tracking vehicles, pedestrians, cyclists, and PLEVs in real-time [33], [76].
- Semantic Segmentation: classifying each pixel in camera images or points in lidar scans to understand road layouts, sidewalks, traffic signs, and other relevant features [44].
- Localization: estimating the vehicle's precise position and orientation within a (high-definition) map using sensor fusion techniques [131].

Sensor fusion methods [222], [235] — e.g., Kalman Filters [19], particle filters [93], or deep learning-based fusion [27], combine noisy measurements into consistent state estimates. The reliability of this module critically affects downstream tasks, since errors in perception/localization directly propagate into planning.

b) *THINK: Planning and Decision-Making*

Given the perception output, the planning subsystem determines the vehicle's future motion while considering safety, efficiency, and rule compliance. Tasks include:

- **Global Route Planning:** generating a high-level route from the current location to the destination using map data and traffic information [89].
- **Local Motion Planning:** creating dynamically feasible trajectories that avoid obstacles and respect kinematic constraints [226], [246].
- **Decision-Making under uncertainty:** managing complex scenarios such as intersections, merges, and yielding to pedestrians, often under uncertainty [24].

Planning increasingly integrates (probabilistic) risk assessment, using risk indicators such as Time-to-Collision (TTC), or probabilistic collision envelopes [130] to make safe decisions.

c) *ACT: Control*

The control subsystem executes the planned trajectory by generating low-level actuator commands (steering, throttle, and brake) within the limits of the vehicle's dynamics. Classical control methods include:

- **Proportional-Integral-Derivative (PID) Control:** simple feedback controllers for longitudinal and lateral control [177].
- **Model Predictive Control (MPC):** optimization-based controllers that handle constraints and multi-variable interactions [62], [151].
- **Learning-Based Control:** such as reinforcement learning approaches that adaptively learn control policies from data [115].

A critical challenge is ensuring robustness to disturbances (e.g., wind, tire slip, model error) while also accounting for uncertainties passed down from perception and planning. Control algorithms must ensure stability, responsiveness, and passenger comfort while adhering to safety constraints. The performance of the control layer is tightly coupled to the accuracy of the planning output; discrepancies can lead to unsafe situations.

Together, these three layers: Perception / Localization → Planning / Decision-Making → Control, constitute the backbone of most AV systems. However, **uncertainties at each layer tend to accumulate, motivating architectures that explicitly account for uncertainty propagation** [183], [232]. This thesis focuses on enhancing the planning/decision-making and control layer to better manage uncertainties arising from perception errors and unpredictable agent behaviors, particularly in complex urban environments with diverse road users.

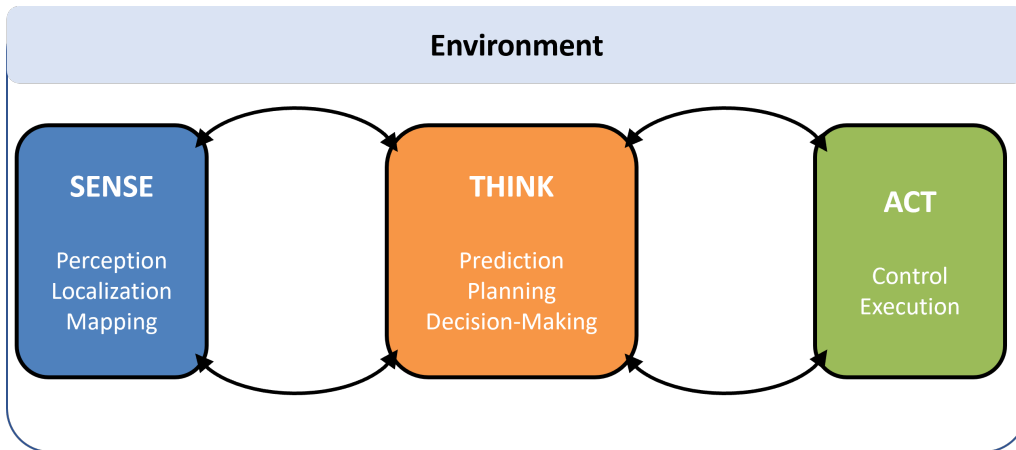


Figure 2.3: The Sense → Think → Act paradigm structures autonomous driving systems into three primary layers: Perception/Localization, Planning and Decision-Making, and Control. Each layer processes information and passes it to the next, with feedback loops for refinement and error correction

2.3.2 Modular System Architectures

The modular approach has long been the dominant paradigm for autonomous driving systems, particularly in academic research and industrial prototypes such as those from Waymo, Cruise, and Baidu Apollo [179], [216]. In this architecture, the AV software stack is decomposed into distinct layers, such as **perception**, **prediction**, **planning**, and **control**, with well-defined functional modules that pass structured information downstream (cf. Figure 2.4). This hierarchical design promotes interpretability, debugging, and validation, which is essential for safety-critical systems. However, it also introduces potential issues of error propagation, where inaccuracies in one stage cascade into subsequent modules.

a) Advantages of Modular System Architectures

- **Modularity in Development:** each module can be independently improved, allowing different research teams to specialize in their domain.
- **Interpretability and Debugging:** engineers can isolate errors in one subsystem without affecting others. For instance, if trajectory deviations occur, it is possible to verify whether the source lies in perception (object misclassification), prediction (incorrect intent estimation), or planning (trajectory optimization errors).
- **Safety Validation and Certification:** modular systems facilitate formal verification and validation processes, as each component can be tested against specific performance metrics and safety requirements. Hence, regulatory frameworks (e.g., ISO 26262, SOTIF) prefer modularity to perform safety validation on well-bounded components [123].

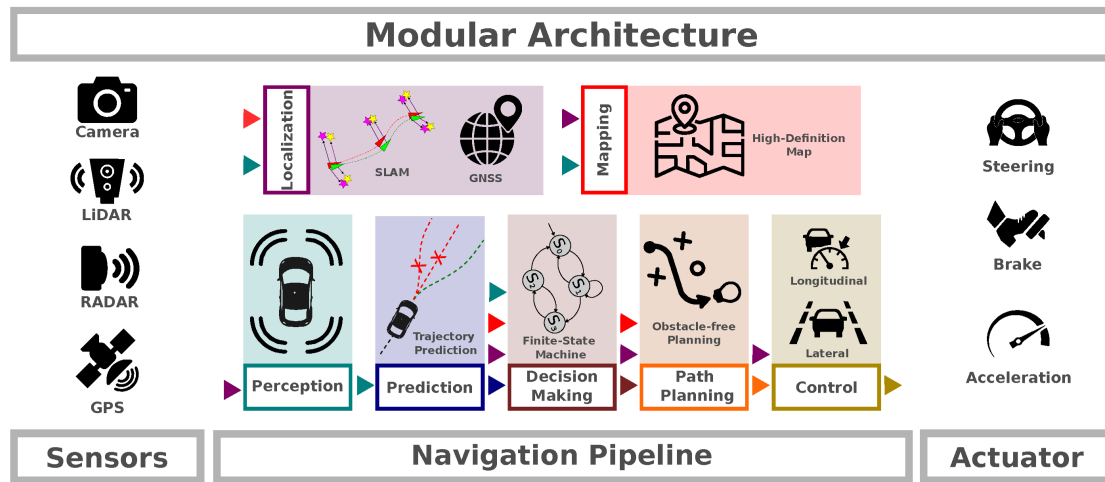


Figure 2.4: The modular architecture of an autonomous vehicle system, illustrating the sequential flow from perception to control. Each module processes data and passes it to the next stage, with potential feedback loops for refinement [183]

b) Limitations of Modular System Architectures

The main drawback of modular pipelines is error and uncertainty propagation. For example:

- **Perception Errors:** misclassifications or missed detections in the perception module can lead to unsafe planning decisions. For instance, if a PLEV is not detected, the planner may generate a trajectory that ignores critical constraints and collides with the agent.
- **Localization Drift:** small errors in vehicle localization can accumulate, leading to significant deviations from the intended path, especially in urban regions where GPS signals are weak.
- **Prediction Uncertainty:** the behavior of other agents (vehicles, pedestrians, PLEVs) is inherently uncertain and multi-modal. If the prediction module only provides a single trajectory hypothesis, discarding valuable uncertainty information, the planner may not account for all possible future scenarios, leading to unsafe maneuvers.

This phenomenon is often referred to as the cascading error problem. Recent research advocates explicitly modeling uncertainty in perception and propagating probabilistic representations through the pipeline [71], [106] as proposed in this work. A generic modular pipeline is illustrated in Figure 2.4.

2.3.3 End-to-End System Architectures

In contrast to modularity, end-to-end System Architectures seek to directly map sensor inputs (e.g., raw camera images) to driving actions (steering, throttle, braking) using a single or multiple neural networks [240]. This approach was

popularized by NVIDIA’s PilotNet [20], [21], which demonstrated that convolutional neural networks can learn lane-following behavior from human driving demonstrations (cf. Figure 2.5).

a) Advantages of End-to-End System Architectures

- **Reduced Complexity in Design:** by eliminating hand-crafted modules, end-to-end learning bypasses the need for explicit detection, prediction, and planning.
- **Simplicity in Data Flow:** the entire perception-to-control pipeline is learned jointly, potentially allowing the model to optimize performance holistically rather than in isolated stages.
- **Potential for Rich Representations:** neural networks can exploit subtle cues in the environment (e.g., shadows, motion cues) that might be overlooked in modular systems.
- **Data-driven Adaptation:** models can, in principle, adapt to new environments without redesigning modules.

b) Limitations of End-to-End System Architectures

- **Lack of Interpretability:** the black-box nature of deep networks makes it difficult to explain specific driving decisions or predict system behavior, complicating debugging and safety validation.
- **Data Requirements:** training requires vast and diverse datasets to cover edge cases. Rare but safety-critical events (e.g., emergency braking for a PLEV cut-in scenario) are underrepresented [42].
- **Generalization:** end-to-end models often fail to generalize to unseen environments, weather conditions, or traffic rules.
- **Safety Certification:** regulatory bodies currently lack methods to certify black-box decision systems.

c) Hybrid System Architectures

Emerging work explores semi-end-to-end systems, where neural networks are trained for specific submodules (e.g., prediction or trajectory generation), but still operate within a modular pipeline (cf. Figure 2.6). Examples include:

- **Imitation Learning for Waypoint Prediction:** networks that predict waypoints or trajectories given high-level commands, which are then tracked by a traditional controller [42].
- **Learning-Based Prediction Modules:** deep models that generate multi-modal future trajectories for surrounding agents directly from sensor data before feeding into a classical planner [191].

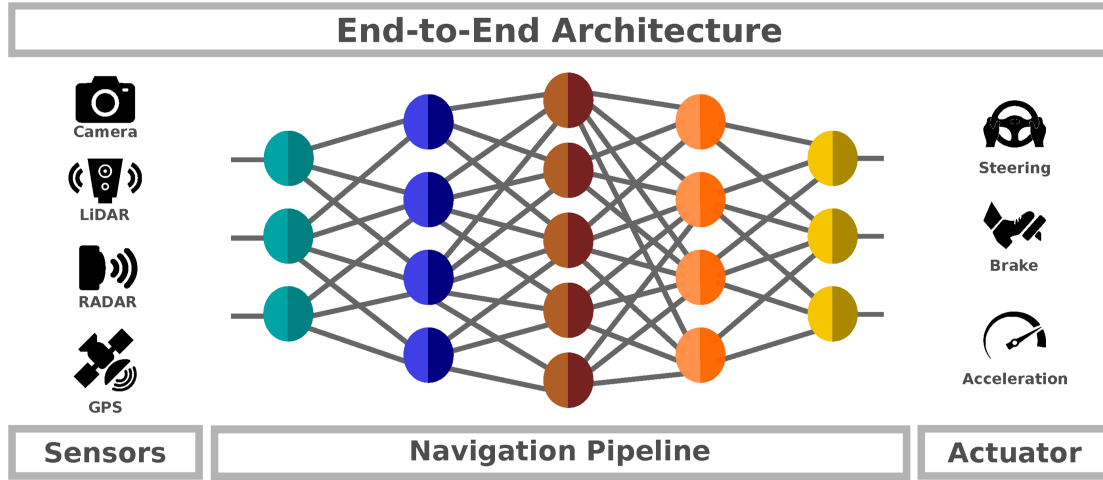


Figure 2.5: The end-to-end architecture of an autonomous vehicle system, where raw sensor inputs are directly mapped to control commands through a deep neural network [183]

- **Reinforcement Learning for Decision-Making:** using RL to learn high-level decision policies (e.g., when to change lanes or yield) with safety verification layers that override unsafe actions [202].

This work adopts a hybrid approach, leveraging a mixture of learning-based perception and planning modules, while maintaining modularity for interpretability and safety validation. By explicitly modeling uncertainties in agent behavior and propagating them through the planning and control layers, we aim to enhance the robustness of autonomous vehicles in complex urban environments with diverse road users.

2.4 UNCERTAINTIES IN AGENT MOTION PREDICTION

A critical bottleneck in autonomous navigation lies in predicting the future behavior of surrounding agents. Unlike static obstacles, dynamic agents such as vehicles, pedestrians, cyclists, and PLEVs exhibit intent-driven, multi-modal, and stochastic motion patterns. As such, deterministic single-trajectory prediction is insufficient for safety-critical decision-making.

According to Lefèvre et al. [130] and Rudenko et al. [186], uncertainty in motion prediction arises from three major sources:

- **Perception Uncertainty:** sensor noise, occlusions, and limited field-of-view lead to inaccuracies in estimating the current state (position, velocity, heading) of other agents.
- **Behavioral Uncertainty:** human drivers and VRUs (vulnerable road users) make decisions influenced by hidden internal states (e.g., destination, aggressiveness) that are not directly observable.

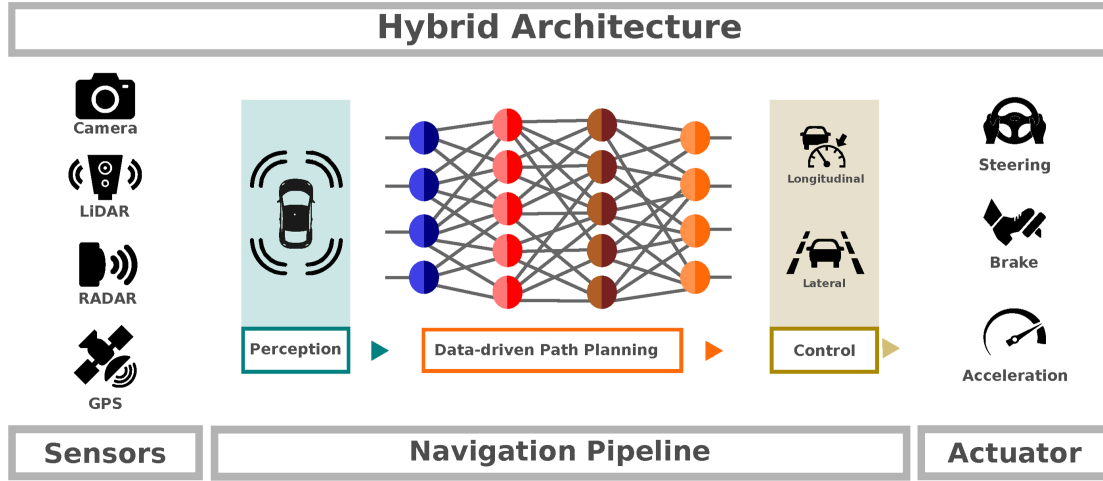


Figure 2.6: The hybrid architecture of an autonomous vehicle system, combining modular components with learning-based modules for specific tasks such as planning or trajectory generation [183]

- **Environmental Uncertainty:** road geometry, traffic rules, and weather conditions influence feasible maneuvers, but may be partially known or change over time.

The following reviews state-of-the-art methods for predicting agent motion, with an emphasis on multi-modal predictions and uncertainty quantification.

2.4.1 Multi-modal Motion Prediction

Early approaches to motion prediction relied on physics-based models (e.g., constant velocity/acceleration), maneuver-based classifiers (e.g., lane change, turn), or deep learning methods that output a single future trajectory. However, these methods struggle to capture the complex interactions and inherent multi-modality of human behavior. The **multi-modality** of human behavior refers to the fact that at any given moment, multiple plausible future actions exist. For instance, at an intersection, an agent may turn left, turn right, or go straight, each with different probabilities. The following subsections summarize key categories of motion prediction methods.

Table 2.1 summarizes the key models of multi-modal motion prediction methods, highlighting their approaches to modeling multi-modal uncertainty and their respective strengths and limitations.

a) Physics-based models

Early approaches relied on kinematic and dynamic models such as constant velocity (CV) [12], [193], constant acceleration (CA) [98], [135], constant turn (CT) [130], [135], [143], or bicycle models [195] (cf. Figure 2.7 (a)). While computationally efficient, these models fail in highly interactive scenarios, since they cannot account for intention changes. The social force model (SFM) [40], [146] is a popular method used to incorporate agent intentions into the physics-based models. The

Table 2.1: Summary of multi-modal motion prediction algorithm classes

Class	Representative methods	How multi-modality is modelled	Strengths / Limitations
Physics-based	Constant Velocity (CV), Constant Acceleration (CA), Constant Turn (CT), bicycle models, Social Force models, Interacting Multiple Model Filter (IMMF)	Multiple dynamic modes (IMMF) or stochastic noise; intentions via attractive/repulsive forces	Fast and interpretable; good short-term, poor at long-term interactions and complex intent modelling.
Maneuver-based	Maneuver classifiers, Hidden Markov Models (HMM), Conditional Random Fields (CRF), probabilistic finite-state machines	Discrete maneuver hypotheses (e.g., lane-change, turn, stop) each conditioned to produce trajectory sets	Incorporates traffic rules and priors; requires labeled maneuvers and limited by predefined maneuver set.
Learning-based	RNN / LSTM, Transformer, Graph Neural Networks (GNN), CVAE / VAE, GAN, Trajectron / Trajectron++, Normalizing flows	multi-modality via latent variables, mixture-density outputs, generative sampling or search to produce multiple plausible futures	Captures complex interactions and multi-modality; requires large dataset, less interpretable, and can be very computationally intensive.

model superimposes attractive forces to a goal with repulsive forces from other agents and obstacles to model agent intentions [152] (cf. Figure 2.7 (b)). These methods are still widely used as a baseline or for short-term predictions. However, they do not capture multi-modality or uncertainty beyond simple Gaussian noise models. To handle this, Interacting Multiple Model Filter (IMMF) [39], [48], [129], [136], [192] frameworks combine multiple dynamic models, by maintaining a probability distribution over different motion hypotheses (CV, CA, CT) (cf. Figure 2.7 (c)). Nevertheless, as seen in [193], these models still struggle with long-term predictions in complex scenarios (cf. Figure 2.7 (d)).

b) *Maneuver-based models*

Maneuver-based models classify agent behavior into discrete categories (e.g., lane change, turn, braking, crossing) and predict mid to long-term future trajectories conditioned on the predicted maneuver [95]. Hidden Markov Models (HMMs)

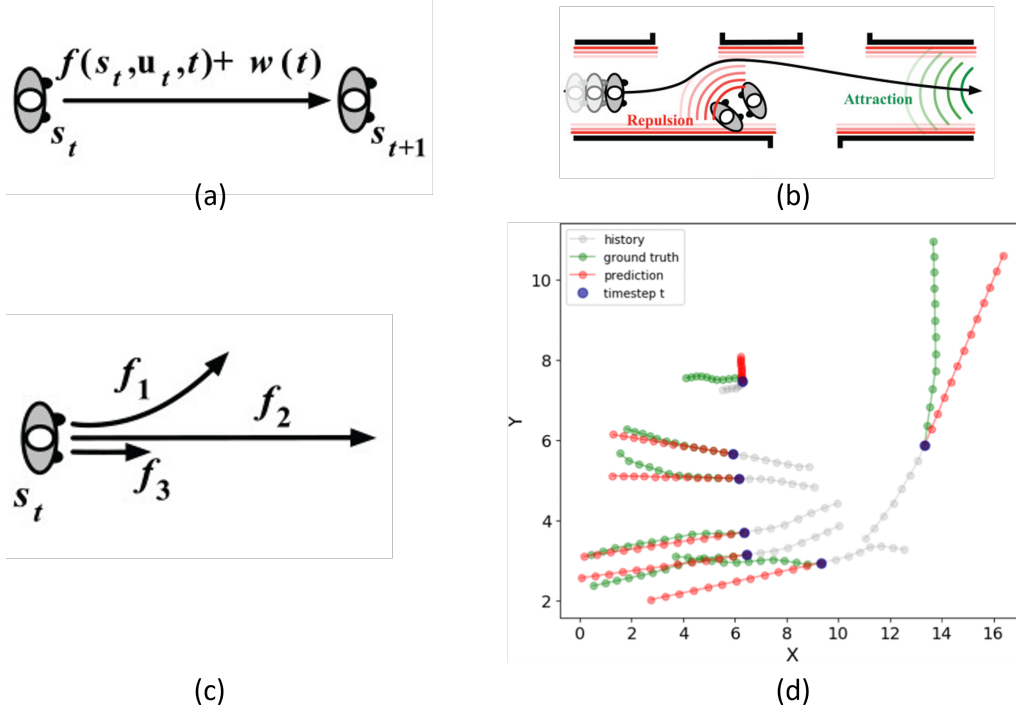


Figure 2.7: Illustrative physics-based prediction methods: (a) a constant dynamics model, (b) an attraction–repulsion model capturing simple intentions, and (c) an interacting multiple model filter (IMMF) with several dynamic modes [186], (d) limitation of physics-based models for long-term predictions [193]

and Conditional Random Fields (CRFs) have been widely applied [130], they use probabilistic graphical model that can effectively capture sequential data by considering contextual information. The authors in [95], [126] predefined a set of trajectories corresponding to various maneuvers. A Hidden Markov Model is subsequently employed to determine the most probable trajectory of an agent for risk assessment by analyzing its sequence of observed measurements (cf. Figure 2.8(a)). In [233], a CRF model is used for multiple target detection and tracking using a CRF model of appearances and motion patterns. A recent CRF model in [87] explicitly builds a CRF over future intention variables along-side spatiotemporal trajectory variables, allowing the model to jointly encode interactions and temporal consistency to predict multi-modal motions. Similarly, the work in [100] employs a probabilistic finite-state machine to model complex driving maneuvers as sequences of primitive elements defined by fuzzy-logic rules. These rules are obtained from a dataset containing signals such as velocity, acceleration, and steering angle. A Bayes filter is then used to recognize maneuvers by estimating the probability of each primitive element given the maneuver model (cf. Figure 2.8(b)).

Maneuver-based methods serve as an intermediate between simple physics models and complex learning-based approaches. They can incorporate domain knowledge and traffic rules to constrain predictions. However, uncertainty remains high when maneuvers are ambiguous. Furthermore, these methods can capture some multi-modality but are limited by the predefined maneuver set and often require extensive labeled data.

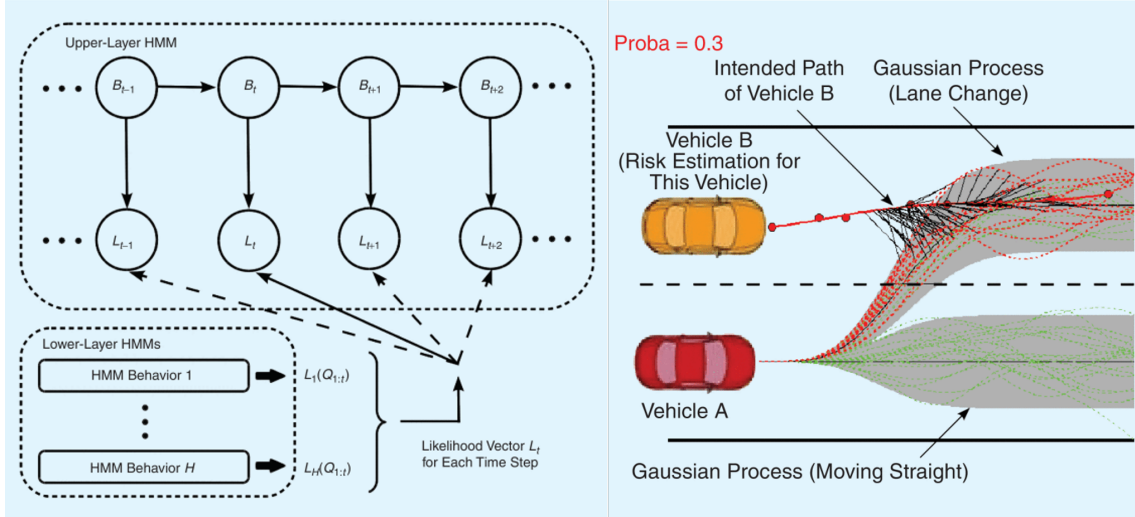


Figure 2.8: Maneuver-based prediction methods: a Hidden Markov Model (HMM) classifying maneuvers (lane change, moving straight) to predict trajectories [126]

c) Learning-based models

Recent advances leverage deep learning to model the rich spatiotemporal dependencies in agent behavior. Recurrent Neural Networks (RNNs) [189] or Long Short-Term Memory (LSTM) [3] networks, and more recently, Transformer architectures [84], [188] have been employed to capture temporal dynamics. These models can learn complex patterns from large datasets, enabling more accurate predictions. Several models have been proposed to explicitly model multi-modality and uncertainty:

- **RNN/LSTM-based models:** capture temporal dependencies in trajectories [51]. Social-LSTM [3] incorporates social pooling to model interactions and the influence of neighboring agents (cf. Figure 2.9(a)).
- **Graph Neural Networks (GNNs):** encode interactions between multiple agents as graph structures for stochastic trajectory prediction [144].
- **Generative models:** Variational Autoencoders (VAE) and Generative Adversarial Networks (GANs) [188] generate a distribution over diverse multi-modal trajectories [128]. Social-GAN [84] uses a GAN framework to generate socially acceptable trajectories.
- **Graph Recurrent Networks:** combine the strengths of GNNs, VAEs, and RNNs to model spatiotemporal dependencies in motion data, such as trajectories of pedestrians, vehicles, or PLEVs in a scene. The output is probability distributions over multi-modal trajectories [106], [191].

These models can generate multiple trajectory hypotheses, each with an associated likelihood or distribution, enabling planners and decision-making systems to reason about uncertainty and risk.

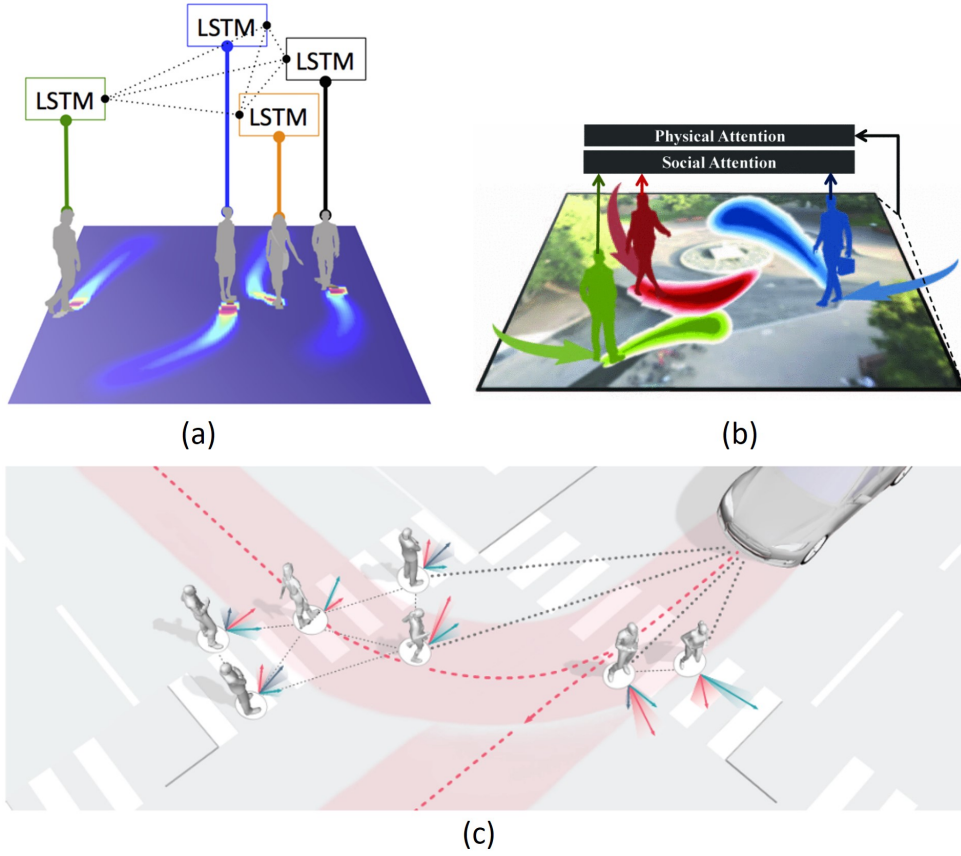


Figure 2.9: Learning-based prediction methods: (a) Social-LSTM with social pooling to model interactions between agents [3], (b) SoPhie: An Attentive GAN for predicting paths compliant to social and physical constraints [188], (c) Graph-structured recurrent model capturing interactions and multi-modal trajectories [191]

2.4.2 Roadside Units (RSUs) for Augmented Perception

One of the primary sources of uncertainty in prediction is the limited perception range and occlusion. For example, an approaching PLEV hidden behind a parked car may not be visible until it is too late for a safe reaction. The integration of Roadside Units (RSUs) into autonomous vehicle systems represents a significant advancement in enhancing perception capabilities, particularly in such complex urban environments (cf. Figure 1.1). RSUs are fixed infrastructure components equipped with various sensors, such as cameras, LiDAR, and radar, that can monitor traffic conditions, detect obstacles, and communicate with vehicles in real-time. When connected via Vehicle-to-Everything (V2X) communication, RSUs can extend the perception field of AVs [85], [157], [158], thereby overcoming some of the limitations of onboard sensors, such as occlusions and limited field-of-view.

2.4.3 Benefits of RSU-assisted prediction

The deployment of RSUs offers several key benefits for motion prediction and overall autonomous driving performance:

- Occlusion handling: RSUs positioned at intersections can detect pedestrians hidden from the AV's line of sight.
- Extended range: infrastructure sensors often have higher vantage points, offering longer detection ranges.
- Cooperative perception: by aggregating observations from multiple vehicles and RSUs, global scene understanding improves [34].
- Reduced prediction uncertainty: with better perception inputs, the variance in predicted trajectories decreases.

2.5 CONCLUSION

This chapter has provided an overview of advances in autonomous driving, state-of-the-art control architectures, and techniques for predicting the motion of agents in urban environments. We discussed the challenges posed by the complexity and unpredictability of urban settings, highlighting the importance of accurate motion prediction for surrounding agents. Various approaches to multi-modal motion prediction were explored, including physics-based models, maneuver-based models, and Learning-based models, with a focus on their strengths and limitations. We also examined the role of roadside units (RSUs) in enhancing perception and prediction capabilities, particularly in occluded scenarios. Overall, this chapter has laid the groundwork for understanding the critical components of autonomous driving systems and set the stage for further exploration of specific algorithms for planning and decision-making in subsequent chapters.

PLANNING AND DECISION-MAKING IN URBAN ENVIRONMENTS

This chapter reviews planning and decision-making methods for autonomous navigation. It covers global and local planning approaches, from graph and optimization-based methods to model-predictive and learning-based control. Various risk assessment and management techniques are discussed alongside relevant safety standards. The chapter highlights how integrated risk assessment and management enable safer trajectory planning and motivate the multi-risk framework introduced next.

3.1 INTRODUCTION

The key research challenge is not just predicting future motion, but also effectively incorporating uncertainty into the decision-making process. For instance, if two trajectories are equally likely (pedestrian crosses vs. waits), a robust planner must account for both. Ignoring uncertainty can result in either unsafe (overly aggressive) or inefficient (overly conservative) behavior [201].

State-of-the-art frameworks now integrate probabilistic forecasts into risk-aware planning, where trajectory optimization explicitly accounts for the distribution of possible agent motions.

3.2 BRIDGING PERCEPTION OUTPUTS AND LOW-LEVEL CONTROL

Planning and decision-making form the central “thinking” layer of the autonomous driving pipeline, bridging perception outputs and low-level control execution. The role of the planning subsystem is to generate safe, feasible, and socially compliant trajectories under uncertainty. In urban settings, this is particularly challenging due to dynamic interactions with pedestrians, cyclists, and other vehicles, as well as the need to follow traffic rules and implicit social conventions.

Broadly, planning and decision-making involve:

- **Global Trajectory Planning:** Planning the high-level path across the road network from the current location to the destination, considering both static and dynamic map data and traffic conditions.
- **Local Trajectory Planning and Control:** Generating dynamically feasible motions in real time that avoid obstacles and respect kinematic constraints.
- **Risk Assessment and Management:** Ensuring safety under uncertainty through risk-aware decision-making, often by evaluating multiple trajectory hypotheses and their associated risks.

3.3 GLOBAL TRAJECTORY PLANNING

Global trajectory planning involves determining an optimal path from the vehicle's current location to a destination, considering the road network, traffic conditions, and any relevant constraints. This high-level planning may require path re-planning to handle dynamic obstacles efficiently and safely in complex urban environments.

Table 3.1: Summary of global trajectory planning classes

Class	Representative methods	Strengths / Typical use-cases / Limitations
Graph-based	A*, D* Lite, ARA*, LPA* [89], [119], [120], [139]	Deterministic, complete (under assumptions), efficient with good heuristics; excellent for structured maps and frequent replanning. Limitation: scalability on very large graphs and sensitivity to heuristic quality.
Probabilistic / Sampling-based	RRT, RRT*, Informed-RRT*, PRM, PRM* [74], [111], [114], [127]	Handles high-dimensional and unstructured spaces; probabilistic completeness, RRT*/PRM* offer asymptotic optimality. Limitation: may require smoothing/optimization for comfort and dynamic feasibility; convergence depends on sample density.
Kinodynamic / Lattice-based	State lattices, Hybrid A* (motion primitives, Dubins/Reeds-Shepp), gPELC [1], [54], [55], [168], [169]	Encodes kinematic/dynamic constraints via motion primitives; produces trackable, smooth paths for non-holonomic vehicles. Limitation: discretization resolution trade-offs and higher per-node cost.
Optimization-based	MILP / MIQP, Sequential Convex Optimization (CHOMP, TrajOpt) [180], [181], [196], [247]	Produces smooth, constraint-respecting trajectories and can encode complex costs/logic; useful as refinement stage. Limitation: computational cost, local minima, scaling with problem complexity.
Learning-based	Imitation learning (Dagger), Reinforcement learning, VINs, PRM-RL, learned heuristics/samplers [65], [104], [185], [209]	Learns priors, heuristics or policies from data to speed planning and capture semantics; can bias samplers or heuristics for efficiency. Limitation: generalization, safety guarantees, and need for large/representative data; often used in hybrid pipelines.

Table 3.1 summarizes the key classes of global trajectory planning methods, highlighting their strengths, typical use-cases, and limitations.

3.3.1 Graph-based Global Trajectory Planning

The core idea is to represent the road network as directed or weighted graph, where nodes correspond to intersections or waypoints, and edges represent road segments with associated costs. Algorithms such as Dijkstra, A*, and their incremental variants (LPA*, D* Lite) search this graph for a path that minimizes a cost function (e.g., distance, energy, travel time, traffic conditions). These methods are deterministic, complete, and can guarantee optimality under certain conditions. Their efficiency depends strongly on heuristics and update mechanisms, making them well-suited for structured maps and environments with frequent cost updates.

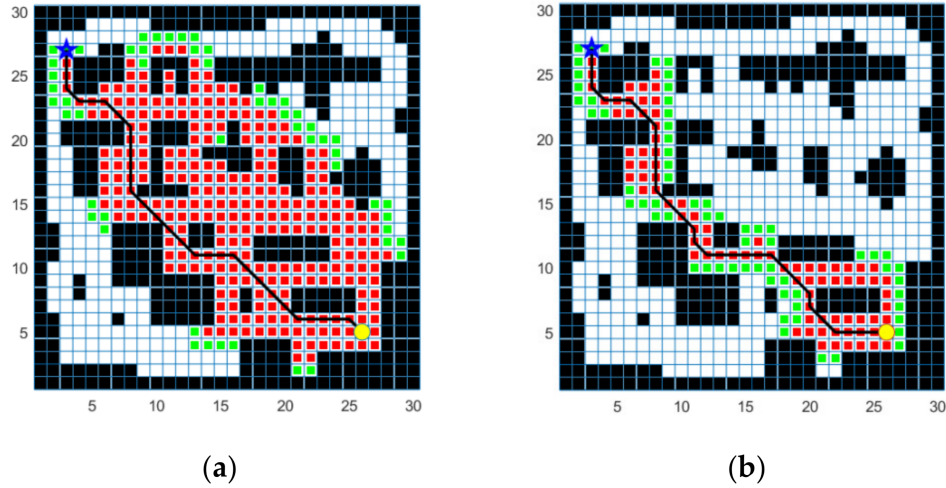


Figure 3.1: Graph-based global trajectory planning methods: (a) A* search on a weighted graph, (b) improved A* search with smaller search expansion [237]

- A* algorithm:** is an informed search algorithm that extends Dijkstra's [64], [108] by incorporating heuristics $h(n)$ to guide the search towards the goal more efficiently [89]. It maintains a priority queue of nodes to explore, ordered by the sum of the cost from the start node and a heuristic estimate of the cost to the goal. This allows A* to focus on promising paths, significantly reducing the number of expansions compared to uninformed methods. The choice of heuristic is critical; it must be admissible (never overestimate) to guarantee optimality. Common heuristics include Euclidean or Manhattan distance, depending on the environment. A* is widely used in autonomous vehicle global planning due to its balance of optimality and efficiency. It can handle dynamic updates to edge costs, making it suitable for real-time route adjustments in response to traffic conditions or road closures. Variants like D* Lite and Anytime Repairing A* further improve efficiency by reusing previous search results when the environment changes [119]. However, A* can still struggle with very large graphs or highly dynamic

scenarios, where the overhead of maintaining the open list and re-expanding nodes can become significant. In such cases, hierarchical or multi-resolution approaches can help manage complexity.

- **D* Lite algorithm:** is an incremental search algorithm designed for dynamic environments where edge costs (e.g., new obstacles) may change over time [119]. It builds upon the principles of A* but optimizes for scenarios where the graph is not static, making it ideal for global planning in unknown or evolving environments. D* Lite maintains a priority queue of nodes to explore, similar to A*, but it efficiently updates the search tree when edge costs change, allowing for rapid replanning without starting from scratch. This makes it particularly well-suited for autonomous vehicles operating in urban environments, where traffic conditions, road closures, and obstacles can frequently alter the optimal path. D* Lite guarantees optimality as long as the heuristic remains admissible. Its efficiency comes from reusing previous search results, significantly reducing the number of node expansions required for replanning. However, the algorithm's performance can still be affected by the frequency and extent of changes in the environment, as well as the complexity of the underlying graph.
- **Anytime Repairing A* (ARA*) algorithm:** is an anytime version of the A* algorithm that provides a trade-off between solution quality and computation time [139]. It starts with a suboptimal solution and iteratively refines it as time allows, making it suitable for real-time applications where quick responses are necessary. ARA* uses a weighted heuristic to find an initial solution quickly, then gradually reduces the weight to improve optimality. This approach allows the algorithm to provide a feasible path almost immediately, which can be crucial for autonomous vehicles in dynamic urban environments where decisions must be made rapidly. ARA* maintains the same optimality guarantees as A* when the weight is reduced to one, ensuring that the final solution is optimal. However, the initial suboptimal solutions may not always be safe or efficient, so careful tuning of the weight parameter is necessary to balance speed and quality.
- **Lifelong Planning A* (LPA*) algorithm:** LPA* generalizes A* to efficiently solve a sequence of similar shortest-path problems on a static topology with changing costs, reusing search trees, and updating only affected regions. Conceptually, it is the substrate upon which D* Lite is built, offering clear proofs and mechanisms for incremental consistency [120]. In global planning, LPA* is ideal for scenarios with repeated queries (e.g., patrol routes) or streaming map updates (sensor fusion). It balances recomputation and reuse via two value functions per node ($g - value$, $rhs - value$) and a key function influenced by the heuristic. LPA* is chosen when the graph connectivity is stable but costs change, enabling persistent planning without discarding prior effort. It integrates with admissible heuristics and supports consistent, optimal solutions after each update, often with far fewer expansions than fresh A*. However, like other incremental algorithms, its performance can

be affected by the frequency and extent of changes in the environment, as well as the complexity of the underlying graph.

3.3.2 Probabilistic and Sampling-based Global Trajectory Planner

Sampling-based methods, such as Rapidly-exploring Random Trees (RRT) [127] and Probabilistic Roadmaps (PRM)[114], offer an alternative approach that is particularly useful in high-dimensional or unstructured environments where full graph discretization is impractical. Instead of discretizing the entire state space, probabilistic methods randomly sample the configuration space to build a graph or tree of feasible paths, which can then be searched, for an optimal route. These algorithms are probabilistically complete, meaning the chance of finding a path (if one exists) approaches one as samples increase. While they do not guarantee optimality, they can efficiently explore complex spaces and handle dynamic obstacles by re-sampling or re-growing the tree. Variants like RRT* provide asymptotic optimality, making them attractive for real-time applications.

- **Rapidly-exploring Random Trees (RRT) and variants (RRT*, Informed-RRT*):** RRT [127], [182] builds a tree by randomly sampling the state space and extending toward samples with a steering function, quickly finding feasible paths in high-dimensional spaces, ideal for global planning with complex geometry and dynamics. It is probabilistically complete but not optimal. RRT* adds rewiring to achieve asymptotic optimality: as samples grow, the solution cost converges almost surely to the optimum, with similar order-complexity to RRT [111]. This makes RRT* a principled global planner when solution quality matters, but exact discretization is hard. Informed-RRT* [74] accelerates convergence by sampling directly within the subset of states that can improve the current best path (a prolate hyperspheroid under path-length costs), retaining RRT*'s guarantees while focusing global search. Extensions like Batch Informed Trees (BIT*) [73] further blend graph and sampling ideas for efficiency (cf. Figure 3.2).

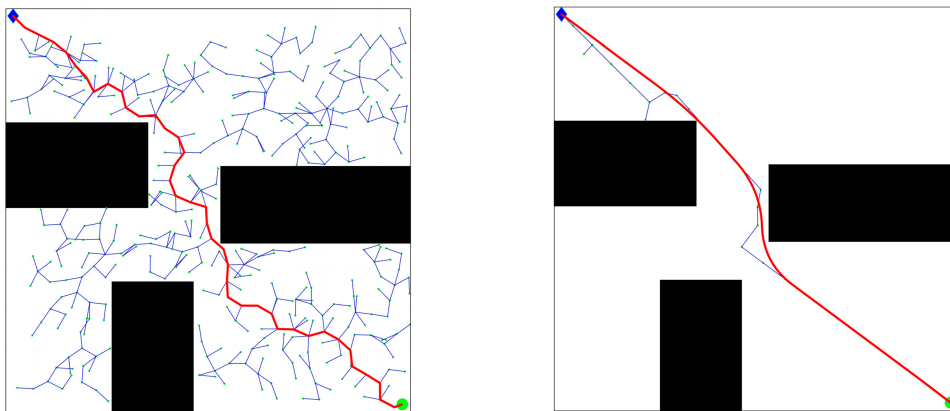


Figure 3.2: Probabilistic and sampling-based global trajectory planning methods: Rapidly-exploring Random Trees (RRT) (left) and Improved RRT (right) building a tree via random sampling [220]

- **Probabilistic Roadmaps (PRM):** PRM [114] is a two-phase global planner: a learning phase builds a roadmap by sampling free configurations and connecting neighbors; a query phase runs graph search (e.g., Dijkstra/A*) on the roadmap (cf. Figure 3.3). It scales nicely to high-dimensional static environments with multiple queries. Variations (k-PRM, lazy PRM, PRM*) [111] tune connection strategies and optimality/efficiency trade-offs; PRM* adds asymptotic optimality akin to RRT*. In practice, roadmap density and connection radii are chosen to balance connectivity versus collision-check cost. PRMs are attractive for long-range navigation where many start/goal pairs are asked on the same world (warehouses, factory floors). They integrate naturally with learned local planners or dynamics-aware edges, supporting kinodynamic feasibility in the global graph.

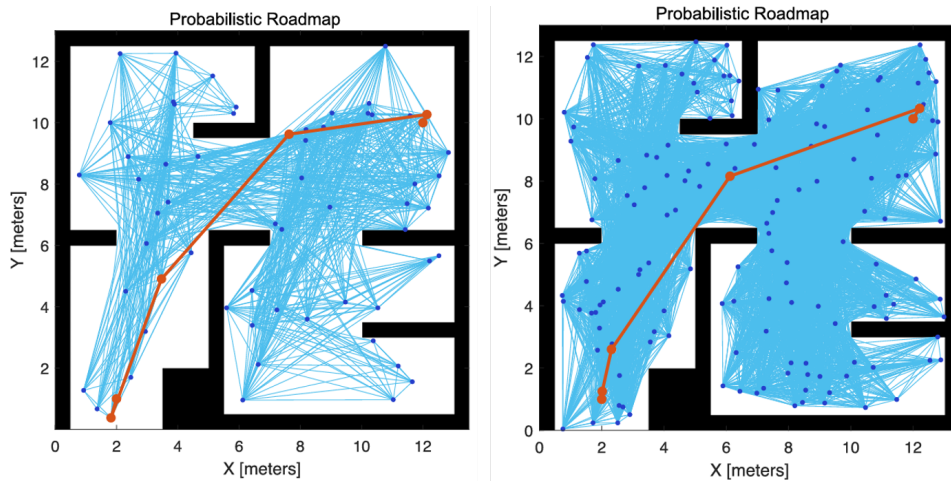


Figure 3.3: Probabilistic and sampling-based global trajectory planning methods: Probabilistic Roadmaps (PRM) with 60 nodes (left) and 150 nodes (right) [145]

3.3.3 Kinodynamic Search on Lattices Global Trajectory Planner

Kinodynamic planning integrates both kinematic constraints (e.g., turning radius, acceleration limits) and dynamic constraints (e.g., velocity, force limits) into the search process. Lattice-based methods discretize the state space into a grid or lattice structure, where each node represents a feasible vehicle state, and edges correspond to motion primitives that respect the vehicle's dynamics. The search algorithm then explores this lattice to find a path that satisfies both the geometric and dynamic constraints of the vehicle. State lattices use pre-computed motion primitives, while Hybrid A* extends A* into continuous state spaces with vehicle models (like Dubins or Reeds-Shepp curves). This approach is particularly effective for vehicles with non-holonomic constraints, such as cars, and can produce smooth, drivable paths. However, it can be computationally intensive due to the high dimensionality of the state space and the need for fine discretization to capture complex maneuvers.

- **State Lattice Planning:** State lattices [168] discretize the continuous, differentially constrained state space into a grid of states connected by precomputed

motion primitives that respect kinematics or dynamics. Global planning then reduces to graph search over feasible motions, yielding dynamically consistent paths [169]. By encoding curvature and speed limits into primitives, the global plan is inherently trackable and smooth, avoiding the “grid jaggedness” of naive A*. Lattice resolution and primitive sets govern optimality and runtime; multi-resolution lattices can speed large-area searches [79], [170]. Lattices excel in off-road autonomous driving and planetary rovers, where feasibility matters as much as optimality. They also enable full-space cost evaluation (terrain, slip) and fast collision checks, supporting real-time global planning in rough terrain.

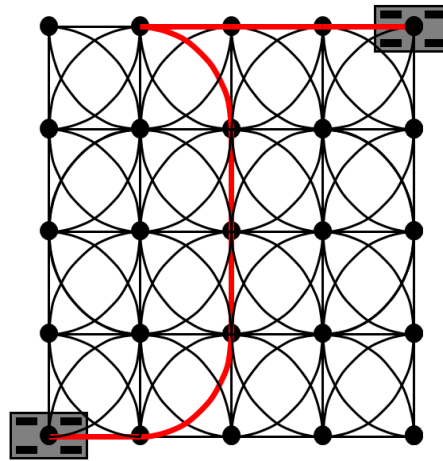


Figure 3.4: State Lattice with motion primitives [169]

- **Hybrid A*:** [54] extends A* to continuous states $(x, y, \theta, \text{sometimes } v)$ by expanding with kinematically feasible forward simulations and using continuous heuristics, producing smooth, drivable global paths even in tight spaces. It often follows with a local optimization phase to polish the solution. The method combines the robustness of grid search with vehicle models (e.g., Dubins/Reeds-Shepp steering), handling forward/reverse and non-holonomic constraints. It is widely used in autonomous-driving stacks for parking lots and structured roads [55]. Trade-offs include higher per-node expansion cost and sensitivity to discretization of controls. Heuristics (analytic path lengths) and motion-primitive sets ameliorate search effort, while post-processing (gradient-based smoothing) ensures curvature/acceleration limits are met.
- **Global Parallel Elliptic Limit-Cycle (gPELC):** and its multi-criteria optimization (PELC*) [1] represents admissible maneuvers as parameterized elliptic limit cycles (ELC) [2], [103] in the vehicle state or configuration space. It allows a controller to steer the vehicle along smooth, dynamically feasible, nonholonomic-compliant paths by selecting and continuously deforming an appropriate ellipse for reactive and cognitive obstacle avoidance and goal following. PELC* casts the ellipse-parameter choice as a multi-criteria optimization problem, considering safety/clearance, progress toward the goal, smoothness/comfort (such as curvature and jerk), and control effort.

It selects balanced or weighted solutions via sampling or small constrained optimization to trade off aggressiveness and conservatism.

3.3.4 *Optimization-based Global Trajectory Planner*

Optimization-based approaches frame trajectory planning as solving a classical constrained optimization problem: minimize a cost (e.g., path length, smoothness, energy) subject to dynamics and collision-avoidance constraints. Mixed-integer linear or quadratic programming (MILP/MIQP) formulations can encode logical and dynamic constraints, while convex relaxations and penalty methods (e.g., CHOMP, TrajOpt) iteratively refine continuous trajectories. These methods are powerful for generating smooth, globally consistent trajectories; however, their scalability is limited by the problem's complexity. Often, they are used to refine or optimize an initial path generated by a graph or sampling-based planner.

- **Mixed-Integer Linear/Quadratic Programming (MILP/MIQP):** [181] programs encode trajectory planning with obstacle avoidance and logic (e.g., region sequencing, collision disjunctions) as a single global optimization. They can yield globally optimal plans under chosen discretization and are amenable to receding-horizon use [180]. Strengths include principled handling of multi-vehicle deconfliction and mission logic [58]; weaknesses are scaling with integers and the need for convex obstacle encodings and time discretization. MIQP variants capture quadratic costs (e.g., control effort) with a similar structure.
- **Convex relaxations & penalty methods (trajectory optimization):** Sequential Convex Optimization (SCO) [196] solves nonconvex motion planning by iteratively convexifying dynamics/collision constraints and penalizing violations, producing smooth, collision-free global trajectories from coarse initial guesses. TrajOpt popularized hinge-loss collision penalties within SCO. Such methods (including CHOMP [247]) optimize a continuous curve with smoothness and obstacle costs, often converging quickly and scaling to high-DOF robots. They are local, where quality depends on initialization, but pair well with global graph/sampling seeds for completeness and optimality in practice. Convex corridor and trust-region strategies improve robustness [197], while penalty schedules and signed-distance fields stabilize convergence. These techniques are a mainstay for refining global paths into dynamically feasible, comfortable trajectories for execution.

3.3.5 *Learning-based Global Trajectory Planner*

Learning-based planners leverage data or experience to accelerate or improve trajectory planning. Imitation learning (e.g., DAgger [88]) trains policies from demonstrations, while reinforcement learning (RL) can discover global policies or dynamics-aware controllers. Hybrid approaches like Probabilistic RoadMaps with RL (PRM-RL) combine classical planning with learned policies, and differentiable architectures like Value Iteration Networks embed planning directly into

neural networks. These methods capture semantic cues and adapt to complex environments, but often require classical planners as safety layers to guarantee feasibility and robustness.

- **Imitation learning for global planning:** IL [185] trains policies to produce global routes or planning priors from expert demonstrations; DAgger addresses covariate shift by aggregating data under the learner’s induced state distribution. In mapping/route tasks, this yields fast global proposals that reflect expert cost shaping. These policies can output waypoints, cost-to-go maps, or heuristics for A*/RRT, reducing search. Benefits include data-driven generalization to semantics (e.g., “roads”) that are hard to hand-craft; risks include distribution shift in novel environments without recovery mechanisms. Hybrid systems combine IL with classical planners using a learned heuristic to guide global search, while a classical planner verifies feasibility or optimality. Active or improved DAgger variants reduce labeling burden while retaining robustness in long-horizon planning.
- **Reinforcement learning for global planning:** RL [49], [147] trains agents to maximize cumulative rewards through trial-and-error interactions with the environment. In global planning, RL can learn policies that map states (e.g., vehicle position, map features) to actions (e.g., waypoints, route segments) that optimize long-term objectives like travel time, safety, or comfort. PRM-RL [65] combines an RL controller (that captures robot dynamics and constraints) with a global PRM that connects the states that the controller can reliably traverse, enabling long-range navigation with dynamics-aware edges. This leverages RL’s expressiveness while retaining PRM’s global reachability. The approach improves end-to-end success on long paths and noisy sensing by ensuring roadmap edges are controllable under the learned policy, not merely geometrically free. In practice, it scales from hundreds of meters to kilometers in large maps.
- **Differentiable planners and Learned Samples or Heuristics:** Value Iteration Networks (VINs) [209] embed a differentiable planning module inside a neural net, enabling end-to-end training of planners or global heuristics that generalize across maps. For global planning, VINs can directly output policy/value maps guiding navigation. Learning to sample [104] focuses on global sampling-based planners where good solutions likely lie. CVAE-based samplers bias RRT/PRM toward promising regions while keeping probabilistic guarantees, providing order-of-magnitude speedups on complex problems. In integrated autonomous driving systems, learned heuristics or sampling strategies are used to guide classical planners, which subsequently verify feasibility and, where possible, optimality. This hybrid approach leverages the strengths of data-driven priors for efficient exploration while maintaining the reliability and robustness of traditional search and optimization methods, enabling scalable and effective global trajectory planning [111].

3.4 LOCAL TRAJECTORY PLANNING AND CONTROL

Local trajectory planning and control refer to the planning and control process of an autonomous system, bridging the gap between global navigation goals and the real-time dynamic constraints of the environment. While global trajectory planning provides a high-level route from origin to destination, local trajectory planning and control are responsible for computing dynamically feasible and safe trajectories in the short horizon, typically ranging from a few meters up to several seconds of driving. This stage operates under tighter real-time constraints and must account for the kinematic and dynamic limitations of the vehicle, while reacting to moving obstacles, road curvature, and uncertainties in perception and prediction.

The main goals of local trajectory planning are:

- **Feasibility:** The generated trajectory must respect the vehicle's dynamic and kinematic constraints.
- **Safety:** It must avoid collisions with static and dynamic obstacles.
- **Comfort:** Trajectories should minimize abrupt steering, acceleration, or jerk.
- **Optimality:** Local planners often optimize a cost function balancing safety, efficiency, comfort, and rule compliance.

This process can be divided into two interconnected problems:

- **Trajectory planning:** computing a dynamically feasible trajectory (position, velocity, and orientation profiles over time) that respects vehicle kinematics, dynamics, and environmental constraints.
- **Control:** generating low-level actuator commands (steering angles, throttle/brake torques, or motor commands) to accurately track the planned trajectory, compensating for disturbances, noise, or modeling errors.

Many local planning and control algorithms are hybrid extensions of global planning techniques combined with dynamic control algorithms. Below, we review the main approaches.

3.4.1 *Trajectory Tracking Controllers (Classic and Geometric)*

Classical tracking controllers provide interpretable, lightweight solutions for real-time trajectory following. They assume the existence of a global trajectory to be tracked.

- **Proportional-Integral-Derivative (PID) controller:** are widely used for longitudinal control (speed and distance regulation) due to their simplicity and effectiveness [178]. They adjust throttle and braking commands based on the error between desired and actual speed or distance to a lead vehicle [17].

- **Pure Pursuit:** a geometric method for lateral control [72] that computes steering commands to follow a reference path by targeting a look-ahead point on the trajectory [9], [46]. It performs well at low to medium speeds, though its accuracy degrades under aggressive maneuvers.
- **Stanley Controller:** developed during the DARPA Grand Challenge [211], corrects both lateral deviation and heading error, and is widely recognized as a robust baseline for lateral control on structured roads [56], [82].

These controllers are often combined with higher-level planning or modern model-based control to improve robustness in nonlinear and dynamic scenarios.

3.4.2 Frenet Frame-based Trajectory Planners

Road-aligned planning in the Frenet coordinate system has been especially successful for curved roads and dynamic traffic scenarios [166]. The Frenet Optimal Trajectory (FOT) approach formulates the planning problem in terms of longitudinal and lateral motion relative to a reference centerline, enabling efficient polynomial-based trajectory generation [214], [226]. Later improvements introduced corridor-constrained optimization, where trajectories are optimized inside safe corridors defined by obstacles, improving collision avoidance in complex urban environments [97]. Because Frenet-based planning decouples longitudinal and lateral motion, it provides a computationally efficient approach for urban driving and highway automation [113]. It can be applied to compute both mid-term global trajectories and short-term trajectories.

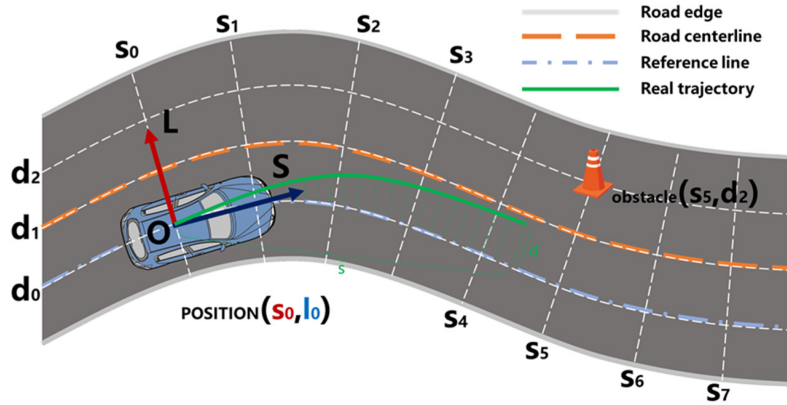


Figure 3.5: Vehicle trajectory in Frenet Frame [166]

3.4.3 Velocity Space Reactive Controllers

Velocity-space reactive controllers sample admissible velocity commands within the vehicle's dynamic limits, forward-simulate short-horizon trajectories, and score them by safety, comfort, and progress to the goal. Classic instances include the Dynamic Window Approach (DWA) [69], [70], which evaluates velocity pairs under kinodynamic constraints and collision checks, and the Timed Elastic Band

(TEB) planner [230], which optimizes a time-parameterized sequence of poses with elastic-band regularization to enforce smoothness and dynamic feasibility. Related velocity-space paradigms that model collision avoidance directly in velocity space are the Velocity Obstacles (VO) framework [67], [227] and its reciprocal extensions (RVO / ORCA) for multi-agent reciprocal collision avoidance [218]. These families of methods are widely used for short-term obstacle avoidance in low-dynamic and cluttered scenes.

3.4.4 *Model-Based Optimal Controllers*

Model-based optimal control methods explicitly consider vehicle dynamics and constraints in trajectory optimization and tracking.

- Iterative Linear Quadratic Regulator (iLQR) and iterative LQG (iLQG) [210], [212] approximate nonlinear dynamics locally and iteratively refine control sequences for efficient optimal control. They have been applied to parking and dynamic obstacle avoidance with uncertainty [38], [148].
- Differential Dynamic Programming (DDP) [150] extends iLQR to handle more general cost functions and constraints, making it suitable for complex urban driving scenarios [99], [210]
- Model Predictive Control (MPC) [37], [159], [173] has become the standard for local trajectory tracking, offering real-time receding-horizon optimization that handles nonlinear dynamics, input constraints, and comfort objectives [62]. Nonlinear MPC (NMPC) [174] formulations capture full vehicle dynamics and constraints, enabling simple or parameterized obstacle avoidance in dynamic environments [66]. NMPC has been successfully deployed in urban driving tasks, balancing safety, comfort, and efficiency.
- Extensions of MPC, Stochastic MPC (SMPC) [29], [154] and Robust MPC [219], explicitly accounts for uncertainty in prediction models. SMPC models the uncertainties as chance constraints in the optimization process. It was employed in [35] [213] [206] to address the interaction between a vehicle and a single pedestrian with low dynamics. In [5], the authors presented an approach to tackle the problem introduced by the multi-modal motion of PLEVs with high dynamics using a Fusion of PIDPs (F-PIDP) and SMPC. Scenario-based MPC [14] extends SMPC by considering multiple constraints to account for multi-modal motion of obstacles. Motion uncertainties can also be modeled as a Value at Risk (VaR) constraint [86] [63]. However, VaR methods are more conservative than chance constraint methods, leading to infeasible solutions that often violate the constraints or hinder performance.
- Model Predictive Path Integral (MPPI) [228] is a sampling-based optimal control method that uses stochastic sampling to approximate the optimal control distribution. It has been applied to aggressive driving and obstacle avoidance tasks, leveraging its ability to handle complex dynamics and constraints in real-time [75], [229].

3.4.5 *Data-Driven and Learning-Based Controllers*

Learning-based controllers leverage data to approximate dynamics or directly map observations to actions. Imitation learning approaches, starting with Pomerleau’s ALVINN system, have evolved into modern deep learning-based frameworks for conditional imitation of expert driving [42], [171]. Reinforcement learning (RL) [223] methods learn policies by optimizing reward signals through interaction, with deep RL [115], [118] providing scalability to complex driving tasks [190]. End-to-end controllers, such as NVIDIA’s PilotNet [21], directly map camera inputs to steering commands. While powerful, these systems face challenges related to interpretability, robustness, and safety certification (cf. Section 3.5.4). A promising research direction combines model-based control with learning-based adaptation, leveraging the strengths of both approaches for improved robustness and generalization.

3.5 RISK ASSESSMENT FOR AUTONOMOUS DRIVING SYSTEMS

Urban driving involves significant uncertainty due to unpredictable behaviors of pedestrians, cyclists, and other vehicles. The decision-making process for collision handling in an autonomous vehicle primarily requires a risk assessment and risk management component. Each of these components plays a crucial role in ensuring the safety and reliability of autonomous driving systems. This includes evaluating the likelihood of collisions, near-misses, and other safety-critical events based on the current environment, vehicle state, and predicted behaviors of surrounding agents. Techniques for risk assessment and management often involve probabilistic modeling, machine learning, and scenario analysis to capture the inherent uncertainties in urban driving.

Risk assessment involves identifying, quantifying, and predicting potential hazards that autonomous vehicles may encounter during operation. This includes fault-based risks (e.g., hardware failures, software bugs) and environment-based risks (e.g., pedestrian behavior, adverse weather, road conditions). The goal is to evaluate the likelihood and potential impact of these risks on the vehicle’s operation. They don’t control the AV directly, but provide safety indicators to inform decision-making and control laws. The following section groups risk assessment methods into three main categories: (1) Deterministic, (2) Probabilistic, and (3) Learning-based methods.

3.5.1 *Deterministic Risk Assessment Methods*

Deterministic methods use simplified prediction models with predefined rules or thresholds to evaluate risk. They emphasize formal rules, logical reasoning, or physical/mathematical models to define safety boundaries, transforming safety requirements into verifiable conditions such as temporal logic or geometric relationships [194]. These approaches are efficient in structured environments where clear rules can be applied, providing high interpretability. They often rely on geometric relationships or criticality indicators, such as distance to obstacles,

overlap between different shapes (ellipses, circles, polygons), collision time, and other safety margins. They use collision-based metrics derived from current vehicle states, such as position, velocity, and acceleration, to assess risk. Examples include Time-to-Collision (TTC) [125], Post-Encroachment Time (PET) [7], and Predictive Inter-Distance Profile (PIDP) [103]. TTC is among the most popular and oldest metrics due to its simplicity, low computational cost, and many variants for different scenarios. Variants of TTC include Extended Time to Collision (ETTC) [156], Time Integrating Time-to-Collision (TIT) [23], and Time Exposed Time-to-Collision (TET) [36].

These metrics provide a straightforward way to quantify the immediacy of potential collisions based on current trajectories. Deterministic methods are computationally efficient for short-term predictions and can be implemented in real-time systems, making them suitable for scenarios where quick decision-making is essential. However, they may struggle to capture the full complexity of dynamic urban environments, especially when dealing with unpredictable or uncertain behaviors of other road users. Below are some examples of deterministic risk assessment methods:

- **Time-to-Collision (TTC):** TTC estimates the time remaining before a collision occurs, assuming constant speeds and paths for both vehicles. It is calculated based on the relative longitudinal speed and distance between co-linear vehicles [125]. A lower TTC indicates a higher risk of collision, prompting the autonomous vehicle to take preventive actions in a lead-follower situation. Equation for TTC is given by:

$$\text{TTC} = \frac{x_1 - x_2 - L_1}{v_1 - v_2} \quad (3.1)$$

where x_1 and x_2 are the positions of the two vehicles, L_1 is the length of the first vehicle, and v_1 and v_2 are their respective velocities. The assumption of constant linear motion limits the application of TTC [94], [126].

- **Extended Time to Collision (ETTC):** ETTC [94], [109] extends the basic TTC concept to 2D planar scenarios, providing a more accurate risk assessment in dynamic scenarios [156], [224], [244]. It accounts for changes in speed over time, making it suitable for urban environments where vehicles frequently accelerate or decelerate. ETTC is calculated using the following equation:

$$\text{ETTC} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (3.2)$$

where $a = (v_{x1} - v_{x2})^2 + (v_{y1} - v_{y2})^2$, $b = 2((x_1 - x_2)(v_{x1} - v_{x2}) + (y_1 - y_2)(v_{y1} - v_{y2}))$, and $c = (x_1 - x_2)^2 + (y_1 - y_2)^2 - (L_1 + L_2)^2$. Where (x_1, y_1) and (x_2, y_2) are the positions of the two vehicles, (v_{x1}, v_{y1}) and (v_{x2}, v_{y2}) are their respective velocities in the x and y directions, and L_1 and L_2 are their lengths. Figure 3.6 shows the use case of TTC and ETTC in a 2D scenario.

- **Time Exposed Time-to-Collision (TET):** TET measures the duration for which the TTC remains below a certain threshold [36], [156], indicating

prolonged exposure to potential collision risk. It is beneficial for assessing situations where vehicles are in close proximity for extended periods, such as in traffic jams or congested urban areas. TET is calculated as:

$$\text{TET} = \int_{t_0}^{t_f} \mathbb{1}_{\text{TTC}(t) < \text{TTC}_{\text{threshold}}} dt \quad (3.3)$$

where $\mathbb{1}$ is the indicator function that equals 1 when the condition is met and 0 otherwise, and t_0 and t_f are the start and end times of the observation period.

- **Post-Encroachment Time (PET):** measures the time difference between when one vehicle leaves a potential collision point and when another vehicle arrives at that point [7], [205], [234]. It is particularly useful for assessing intersection safety and scenarios where vehicles cross paths. A lower PET indicates a higher risk of collision, prompting the autonomous vehicle to take preventive actions. PET is calculated as:

$$\text{PET} = t_{\text{arrival}} - t_{\text{departure}} \quad (3.4)$$

where t_{arrival} is the time when the second vehicle arrives at the potential collision point, and $t_{\text{departure}}$ is the time when the first vehicle leaves that point.

- **Predictive Inter-Distance Profile (PIDP):** evaluates the distance between two or more vehicles over a prediction horizon, considering their current velocities and accelerations with any path for each of these vehicles [16], [103]. It provides a dynamic assessment of risk by predicting future inter-distance and calculating the minimum distance between vehicles during the prediction period. A smaller minimum distance indicates a higher risk of collision [5]. PIDP is calculated as:

$$\text{PIDP} = \min_{t \in [0, T]} \|p_1(t) - p_2(t)\| \quad (3.5)$$

where $p_1(t)$ and $p_2(t)$ are the predicted positions of the two vehicles at time t , and T is the prediction horizon.

- **Deceleration to Safety Time (DST):** estimates the required deceleration for a vehicle to avoid a collision within a specified time frame [101], [194]. It is calculated based on the current speed, distance to the obstacle, and the time available to react. A higher required deceleration indicates a higher risk of collision, prompting the autonomous vehicle to take preventive actions. DST is calculated as:

$$\text{DST} = \frac{3(v_1 - v_2)^2}{2(x_1 - x_2 - L_1)} \quad (3.6)$$

where v_1 and v_2 are the velocities of the two vehicles, x_1 and x_2 are their positions, and L_1 is the length of the first vehicle.

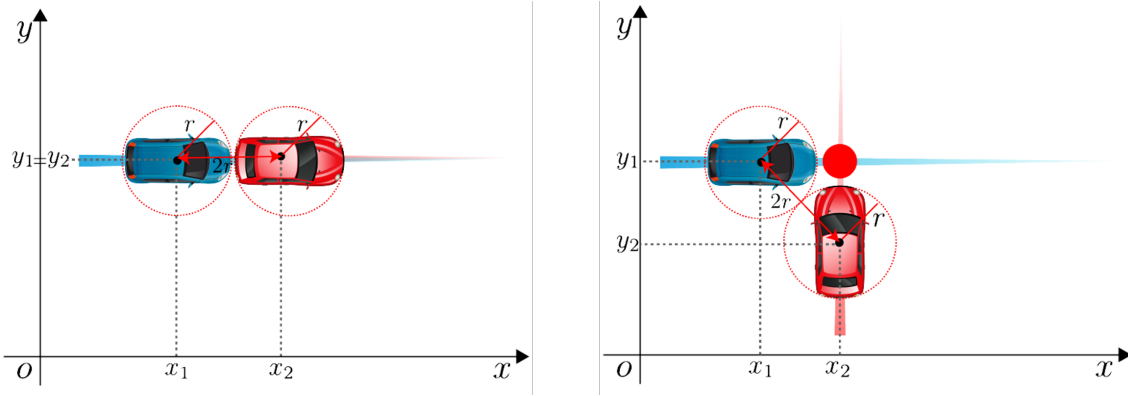


Figure 3.6: TTC in co-linear navigation scenario (Left) and ETTC in 2D navigation scenario (Right) [244]

3.5.2 Probabilistic Risk Assessment Methods

Probabilistic risk assessment methods model the uncertainty in the environment and agent behaviors using statistical techniques. They estimate the likelihood of various outcomes based on predicted trajectories and environmental factors. Hidden Markov Models (HMM) [155], Bayesian networks [132], Gaussian processes [142], and Monte Carlo simulations [28] are commonly used techniques. These methods provide a more comprehensive risk assessment by considering the inherent uncertainties in urban driving scenarios. They can capture complex interactions between multiple agents and account for variations in behavior, leading to more robust decision-making. However, probabilistic methods often require more computational resources and may involve complex modeling assumptions that can be challenging to validate in real-world scenarios. Below are some examples of probabilistic risk assessment methods:

- **Hidden Markov Models (HMM):** are statistical models that represent systems with hidden states. In the context of autonomous driving, HMMs can be used to model the behavior of surrounding agents, such as pedestrians or other vehicles [155]. By inferring the hidden states based on observed actions, HMMs can predict future behaviors and assess the associated risks [68]. For instance, a HMM can model the likelihood of a pedestrian crossing the street based on their current position, speed, and environmental context.
- **Bayesian Networks:** are graphical models that represent probabilistic relationships among variables [132]. In the context of autonomous driving, they can be used to model the dependencies between various factors influencing risk, such as vehicle states, environmental conditions, and agent behaviors. By updating beliefs based on new observations, Bayesian networks can provide real-time risk assessments that adapt to changing conditions [199]. For example, a Bayesian network can model the likelihood of a collision based on the positions and velocities of surrounding vehicles, as well as road conditions and weather [112].

- **Gaussian Processes (GP):** are non-parametric models that can capture complex, continuous functions and their uncertainties [92], [198]. In risk assessment, GPs can be used to predict the trajectories of surrounding agents and estimate the uncertainty associated with these predictions. By modeling the spatial and temporal correlations in agent behaviors, GPs can provide probabilistic risk predictions that account for the variability in agent actions [142], [162]. For instance, a GP can predict the future positions of obstacles based on their current motion patterns and environmental context, allowing the autonomous vehicle to assess the risk of potential collisions [91], [236].
- **Monte Carlo Simulations:** simulations involve generating a large number of random samples to estimate the probability distribution of outcomes [15], [241]. In autonomous driving, Monte Carlo methods can be used to simulate various scenarios and evaluate the risk associated with different actions. By sampling from the distributions of agent behaviors and environmental conditions, Monte Carlo simulations can provide a comprehensive risk assessment that captures the full range of possible outcomes. For example, an autonomous vehicle can simulate multiple trajectories for itself and surrounding vehicles to estimate the probability of collision under different maneuvers [28]. The drawback is the computation cost required to consider all possible scenarios.
- **Grid Based Risk Assessment:** this method divides the environment into a grid and assigns a risk value to each cell based on factors such as obstacle density, agent behavior, and environmental conditions [60]. Using Bayesian occupancy filtering (BOF) [45], risk values can be updated in real-time as new information becomes available, allowing the autonomous vehicle to adapt its risk assessment dynamically.

3.5.3 *Learning-Based Risk Assessment Methods*

Learning-based risk assessment methods leverage machine learning techniques to model complex interactions and predict risks based on historical data. These methods can learn from large datasets of driving scenarios, capturing patterns and correlations that may not be easily modeled using traditional techniques. Deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), can be trained to predict risk levels based on sensor data, agent behaviors, and environmental context. Long Short-Term Memory (LSTM) networks, a type of RNN, are particularly well-suited for modeling temporal dependencies in driving data. LSTM processes sequences for trajectory and risk prediction, often in encoder-decoder setups for networked risks or combined with other techniques for lane-change intentions. Reinforcement Learning (RL) can also be employed to learn risk-aware policies that balance safety and efficiency by maximizing rewards, such as collision avoidance, through environment interaction. Variants like Risk-Sensitive RL, such as RACER [207],

use Conditional Value at Risk (CVaR) [18] as an objective and for uncertainty estimation in safe decision-making.

Learning-based methods can provide high accuracy and adaptability, as they can continuously improve with more data. However, they may also suffer from issues related to interpretability, overfitting, and the need for extensive training data. Below are some examples of learning-based risk assessment methods:

- **Convolutional Neural Networks (CNNs):** are deep learning models that excel at processing spatial data, such as images or point clouds. In the context of autonomous driving, CNNs can be used to analyze sensor data (e.g., camera images, LiDAR point clouds) to identify potential hazards, assess risk levels, and directly control the vehicle [21], [22]. By learning hierarchical features from raw sensor inputs, CNNs can detect objects, classify road conditions, and estimate the likelihood of collisions. For example, a CNN can analyze camera images to identify pedestrians, vehicles, and road signs, providing a risk assessment based on their positions and movements [32].
- **Recurrent Neural Networks (RNNs):** are designed to process sequential data and capture temporal dependencies. In risk assessment for autonomous driving, RNNs can be used to model the behavior of surrounding agents over time, allowing the vehicle to predict future actions and assess associated risks [13]. Long Short-Term Memory (LSTM) networks [8], a type of RNN, are particularly effective at handling long-term dependencies in driving data. For instance, an LSTM network can analyze the trajectories of nearby vehicles to predict their future positions and evaluate the risk of potential collisions [117], [231].
- **Reinforcement Learning (RL):** involves training agents to make decisions by maximizing cumulative rewards through interactions with the environment. In the context of autonomous driving, RL [107], [115] can be used to learn risk-aware policies that balance safety and efficiency. By incorporating collision risk metrics into the reward function, RL agents can learn to avoid high-risk situations while optimizing for travel time and comfort [238]. In [208], an RL agent learn to navigate through traffic while minimizing the likelihood of collisions by adjusting its speed and trajectory based on the behavior of surrounding vehicles [96], [207].
- **Graph Neural Networks (GNNs):** are designed to process graph-structured data and capture relationships between entities [176], [239]. In autonomous driving, GNNs can be used to model interactions between multiple agents and assess the overall risk in complex traffic scenarios. By representing vehicles, pedestrians, and road infrastructure as nodes in a graph, GNNs can learn to predict the likelihood of collisions and other safety-critical events based on the spatial and temporal relationships between agents [204]. For example, a GNN can analyze the interactions between vehicles at an intersection to assess the risk of potential collisions and inform the autonomous vehicle's decision-making process [140].

3.5.4 Structured Framework for AV Risk Assessment Standards

The development and deployment of ADAS and automated driving systems (ADS) have necessitated robust risk assessment frameworks to ensure safety in road vehicles. Risk assessment in automotive systems addresses two primary concerns: functional safety (ensuring systems do not fail dangerously) and safety of the intended functionality (ensuring systems perform safely even when operating as designed). These concerns are compounded in urban environments, where dynamic interactions with other road users introduce additional complexity. For any AV to operate on public roads, safety is the utmost factor of consideration for deployment, since the consequence of deviations from functional safety actions may result in a nonlinear fatality risk level. Hence, to achieve high levels of safety, a well-considered approach of risk assessment and management is essential as part of the design, verification, validation, and deployment of AVs. Several risk assessment standards are being proposed to address the challenges posed by higher levels of autonomous driving (SAE Levels 2+). These standards can be broadly classified into IEEE standards and non-IEEE standards, as shown in Figure 3.7.

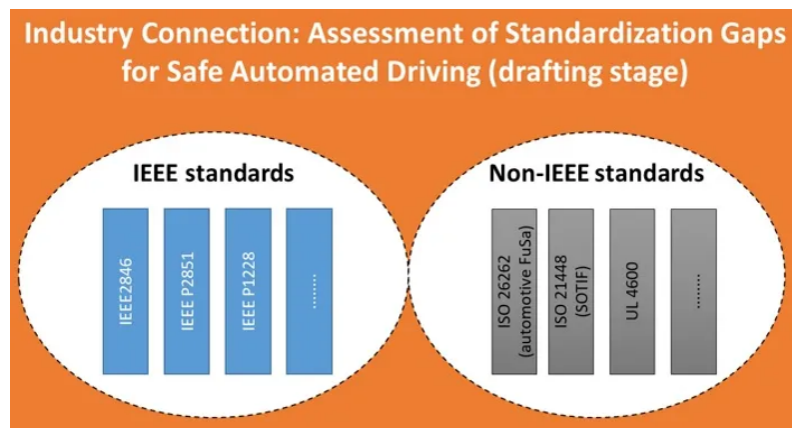


Figure 3.7: AV Safety Standards by the IEEE [110]

Some of the most relevant standards include:

- ISO 26262: Road vehicles - Functional safety, is an international standard adapted from IEC 61508 for E/E systems in road vehicles [165]. This standard provides a framework for ensuring the functional safety of electrical and electronic systems in road vehicles. It includes guidelines for Hazard Analysis, Risk Assessment (HARA) [225], and the development of safety measures to mitigate identified risks.
- ISO/PAS 21448-2022: Road vehicles - Safety of the intended functionality (SOTIF) [80]. This standard focuses on the safety of systems that may not have been designed to fail but could still lead to hazardous situations due to limitations in their intended functionality, particularly in sensor-dependent systems like ADAS (SAE Levels 1-2) and AVs. It provides guidance on identifying and mitigating risks associated with limitations in sensors, algorithms, or Operational Design Domains (ODDs) [163].

- UL 4600: Standard for Safety for the Evaluation of Autonomous Products [121]. This standard provides a comprehensive framework for evaluating the safety of autonomous products, including AVs. It covers aspects such as system design, risk assessment, and validation processes to ensure that autonomous systems operate safely in real-world environments [122].
- IEEE P2846: This standard for AV Safety aims to establish safety requirements and guidelines for the design, development, and deployment of AVs [149]. It addresses various aspects of AV safety, including risk assessment, system architecture, and operational safety measures. It defines minimum assumptions for scenarios like cut-ins, lane changes, or pedestrian interactions, using discrete mathematics and logic.
- RSS: Responsibility-Sensitive Safety, proposed by Mobileye and Intel in 2017 [221]. This is a formal model for AV safety that defines safe distances and behaviors based on the actions of other road users [141]. It provides a mathematical framework for ensuring that AVs can operate safely in dynamic environments by adhering to defined safety principles.

The layered relationship between these standards creates a robust framework for automotive risk assessment and management, ensuring that all aspects of safety are considered throughout the development and deployment process. Our focus is therefore to develop a generic solution that can be adapted to these standards, particularly the RSS, to ensure the AV achieves a clear and measurable safety level, such as the safety distance.

3.6 RISK MANAGEMENT AS A DUAL OF RISK ASSESSMENT

Risk management involves making decisions and taking actions to mitigate identified risks. This includes selecting safe trajectories, adjusting speed, and executing maneuvers that minimize the likelihood of collisions or other adverse events. Risk management strategies often incorporate risk assessments into the planning and control processes, allowing the autonomous vehicle to adapt its behavior based on the current risk level. Therefore, in the domain of decision-making under uncertainty, risk assessment and risk management form a symbiotic duality. Risk assessment provides the necessary information about potential hazards and their likelihood. In contrast, risk management employs both reactive and proactive strategies, based on this information, to make informed decisions that prioritize safety.

Traditional approaches to risk management often involve rule-based systems or predefined safety margins [194]. These are mostly Finite State Machine (FSM) decision-making strategies [216] that are effective in structured environments as employed by the Tartan Racing team "Boss" in the DARPA Grand Challenge 2007 and other participants. State machines were also used by the team that worked on the autonomous drive on the Bertha Benz Memorial Route [245]. However, they lack the flexibility to handle complex and dynamic urban scenarios. A pivotal aspect of risk management in AVs is trajectory planning, which serves

as the operational core for navigating risks by generating feasible, safe paths from current positions to desired goals. Trajectory planning methods can directly incorporate risk assessment outputs to optimize paths, balancing objectives like safety, comfort, efficiency, and compliance with traffic rules. For instance, MPC-based planners predict future states over a finite horizon, solving optimization problems that minimize risk metrics such as time-to-collision or potential field energies, while respecting vehicle dynamics and constraints. These methods excel in structured environments like highways, where they can proactively adjust trajectories to avoid high-risk zones identified through predictive risk analysis. Sampling-based approaches, such as RRT and its variants like RRT* (cf. Section 3.3.2), often integrate risk by biasing samples away from hazardous areas or using cost functions that penalize proximity to obstacles [41].

Probabilistic risk management strategies leverage uncertainty estimates from risk assessment to make decisions that account for the stochastic nature of urban environments. Techniques such as Bayesian Networks and Partially Observable Markov Decision Processes (POMDPs) [26] enable AVs to evaluate the expected outcomes of different actions under uncertainty, selecting maneuvers that minimize expected risk. Two well-known extensions of Bayesian networks for decision-making under uncertainty are Decision Networks (DNs) or Influence Diagrams (IDs) and Dynamic Bayesian Networks (DBNs). BNs model the relationship between random variables using directed acyclic graphs (DAGs). DNs and DBNs extend BNs by adding decision and utility nodes to model sequential decision-making problems. In [194], combined Deceleration to Safety Time (DST) and DNs to formalize lane change decision-making, while the authors in [200] applied DBN to explicitly model the intentions of surrounding vehicles and the temporal inter-dependencies between their future behaviors.

On the other hand, POMDP is an extension of Markov Decision Processes (MDPs) [25] that accounts for partial observability of the environment. The authors in [215] propose an online POMDP-based framework for urban driving that incorporates probabilistic risk assessments to guide lane change decision-making while accounting for inevitable sensor noises faced in urban environments.

Learning-based risk management approaches utilize machine learning models to learn risk-aware policies from data. Reinforcement learning (RL) [161] methods can be trained to optimize decision-making in complex environments by maximizing rewards that incorporate safety considerations. For example, risk-sensitive RL algorithms can learn to avoid high-risk maneuvers while still achieving navigation goals, adapting to new scenarios through continuous learning [207].

Hence, techniques for risk management may involve optimization, decision theory, and control theory to ensure that the vehicle operates safely while achieving its navigation goals. Overall, trajectory planning methods not only help to mitigate identified risks but also contribute to iterative risk assessment and management, fostering safer AV deployment in diverse scenarios.

3.7 CONCLUSION

In this chapter, we have explored the various aspects of planning and decision-making in the context of autonomous vehicles (AVs). We delved into global and local trajectory planning methods, covering graph-based algorithms, sampling-based techniques, optimization-based approaches, and learning-based strategies. The integration of risk assessment and management into the planning process was also discussed, emphasizing the need for safety in uncertain urban environments. We discussed the importance of establishing a robust safety framework that incorporates multiple standards and methodologies to ensure the safe operation of AVs in complex urban environments. By leveraging advanced modeling techniques and probabilistic approaches, we can better understand and mitigate the risks associated with AVs, ultimately paving the way for their widespread adoption. The next chapter will present the proposed global architecture for multi-risk assessment and management in autonomous urban driving, highlighting the methodologies and algorithms employed to enhance safety.

MIRA-MOC: PROPOSED GLOBAL CONTROL ARCHITECTURE FOR MULTI-RISK ASSESSMENT AND MANAGEMENT

This chapter presents the proposed architecture for Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC). The system integrates perception, multi-modal prediction, sampling-based trajectory generation, collision-risk assessment (Fusion of stochastic Predictive Inter-Distance Profiles, F-sPIDP), and uncertainty-aware local control. The functionalities and connectivity of each block in the architecture are presented in this chapter as a foundation for the multi-risk-aware methods developed in subsequent chapters.

4.1 INTRODUCTION

Ensuring safety in the presence of vulnerable agents such as pedestrians, cyclists, and Personal Light Electric Vehicles (PLEVs) requires not only accurate perception and reliable motion planning but also continuous risk assessment and management to evaluate the likelihood and severity of potential conflicts. While predictive control algorithms provide a powerful framework for generating safe trajectories under constraints, they are inherently limited by their reliance on uncertain predictions of other agents. As highlighted in Chapter 2 & 3, this dependence often leads to overly conservative behavior, particularly in the presence of PLEVs whose motion can be erratic, multi-modal, and challenging to predict.

To overcome these challenges, risk assessment and management frameworks introduce an intermediate decision-making layer between prediction and control. Instead of relying solely on worst-case assumptions or probabilistic constraints embedded within the control law, risk-aware approaches quantify safety in terms of interpretable risk metrics, such as TTC, Predictive Inter-Distance Profiles (PIDP), or Time Exposed TTC (TET) (cf. Section 3.5). These metrics allow the system to reason explicitly about collision likelihoods, severity levels, and acceptable safety margins, thereby enabling balanced trade-offs between safety, efficiency, and social compliance.

Model Predictive Control (MPC) and its variants, such as Stochastic MPC (SMPC), are widely used control strategies for autonomous vehicles due to their ability to handle constraints and optimize performance over a prediction horizon (cf. Section 5.3). However, MPC has several limitations when it comes to multi-risk assessment and management in dynamic and uncertain environments, particularly in the presence of multiple moving agents. Some of the key limitations include:

- **Computational Complexity:** MPC requires solving an optimization problem at each time step, which can be computationally intensive, especially in

complex environments with multiple agents. This can lead to delays in decision-making, which is critical for safety.

- **Limited Prediction Horizon:** The prediction horizon of MPC is typically limited due to computational constraints. This can result in suboptimal decisions that do not account for long-term safety considerations, especially in scenarios with fast-moving agents.
- **Handling Uncertainty:** MPC often assumes perfect knowledge of the environment and the behavior of other agents. In reality, there is significant uncertainty in the motion of surrounding agents, which can lead to unsafe decisions if not properly accounted for. Though Stochastic MPC tries to bridge this gap, it can be very conservative.
- **Conservatism vs. Aggressiveness:** To ensure safety, MPC may adopt a conservative approach, which can lead to overly cautious behavior and reduced efficiency. Conversely, an aggressive approach may compromise safety in favor of performance.
- **Scalability Issues:** As the number of surrounding agents increases, the complexity of the optimization problem grows exponentially. This can make it challenging to scale MPC to scenarios with many dynamic agents while maintaining real-time performance.
- **Non-convex Constraints:** The presence of non-convex constraints, such as collision avoidance with multiple agents and road curvature, can complicate the optimization problem and lead to suboptimal or infeasible solutions.

To address these limitations, it is essential to develop strategies that enhance the safety and efficiency of predictive control algorithms in dynamic and uncertain environments. This may involve integrating probabilistic models, hierarchical control structures, and multi-level optimization approaches, as proposed in this work.

Predictive risk assessment does not replace predictive control; rather, it complements it. By filtering, shaping, or constraining the set of feasible trajectories based on multi-criteria risk evaluations, AVs can exhibit safer, less conservative behavior. In particular, integrating predictive risk metrics with stochastic controllers offers a path to manage uncertainty in mixed traffic while avoiding unnecessary hesitation or abrupt maneuvers.

4.2 COMPONENTS OF MIRA-MOC

The proposed global architecture for predictive Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) in autonomous vehicles (AVs) is illustrated in Figure 4.1. It is designed as a modular system that integrates perception, prediction, risk assessment, motion optimization, and robust control. Like most modular architectures, it requires inputs from the perception and localization modules to obtain information about the state of surrounding

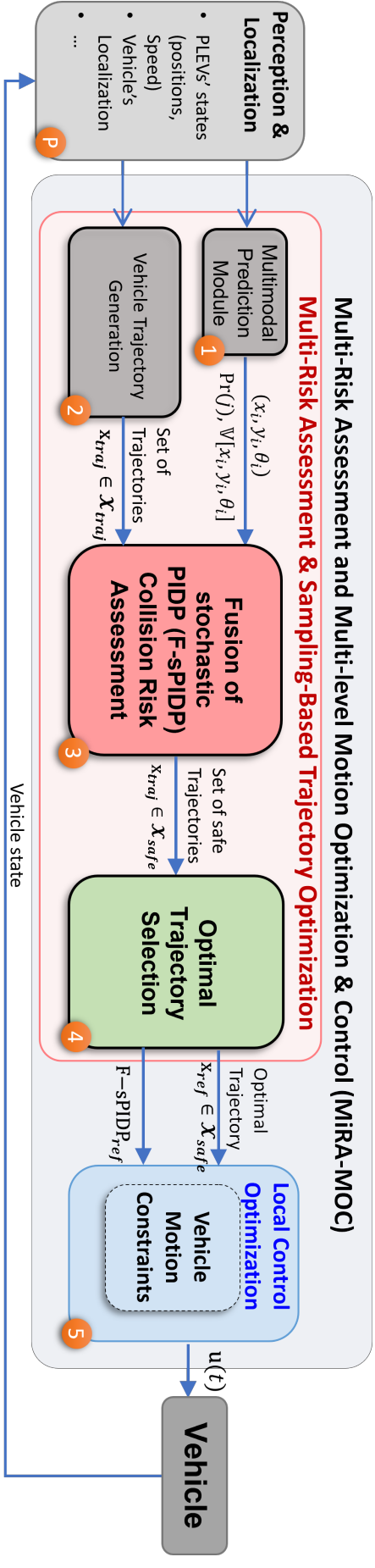


Figure 4.1: Global architecture for Multi-Risk Assessment and Multi-level Motion Optimization and Control (MIRA-MOC) in autonomous vehicles. The system integrates perception, multi-modal motion prediction, multi-risk assessment using Fusion of stochastic PIDP (F-sPIDP), sampling-based planner, and predictive controllers to ensure safe and efficient navigation in mixed traffic environments

agents and the environment (cf. block “P” in Figure 4.1). These inputs serve as the foundation for downstream modules that are essential to ensure precise assessment of the situation for both safety and efficiency in autonomous driving.

The architecture comprises five key modules, organized sequentially as:

1. Multi-modal Prediction Module (Block “1”, cf. Section 4.3).
2. Vehicle Trajectory Generation (Block “2”, cf. Section 4.4)
3. F-sPIDP Collision Risk Assessment (Block “3”, cf. Section 4.5)
4. Optimal Trajectory Selection (Block “4”, cf. Section 4.6)
5. Local Control Optimization (Block “5”, cf. Section 4.7)

The following Sections provide an overview of each component.

4.3 MULTI-MODAL PREDICTION MODULE

Predicting the future motion of surrounding agents is an essential component of autonomous driving. In mixed-traffic urban environments, especially with pedestrians, and PLEVs, motion patterns are highly uncertain and multi-modal in nature. Agents may change lanes, yield, cross unexpectedly, or accelerate/decelerate abruptly. Relying on a single deterministic trajectory forecast can therefore lead to unsafe or overly optimistic behavior of the ego-vehicle.

Recent advances in trajectory prediction have emphasized multi-modal and probabilistic approaches, which provide a distribution of possible futures rather than a single trajectory. This probabilistic representation allows other planning modules to evaluate safety across diverse hypotheses, instead of conditioning decisions on a single, potentially incorrect prediction [81], [105].

4.3.1 Problem Formulation

Let $i \in \{1, \dots, N_{agents}\}$ denote a surrounding agent. At the current time t_0 , the goal is to predict its future motion over a horizon T .

- The agent’s state is parameterized as: $\mathbf{x}_i(t) = [x_i(t), y_i(t), \theta_i(t)]$ where (x_i, y_i) are Cartesian coordinates of the agents position and θ_i is the heading angle.
- Instead of a single prediction, the module produces N_{traj} modes:

$$\{\mathbf{x}_i^j(t)\}_{j=1}^{N_{traj}}, \quad t \in [t_0, t_0 + T] \quad (4.1)$$

each mode $j \in N_{traj}$ represents a possible future trajectory.

- Each trajectory is associated with a probability $Pr(j)$, such that

$$\sum_{j=1}^{N_{traj}} Pr(j) = 1 \quad (4.2)$$

- Uncertainty in each trajectory is estimated by a covariance matrix $\Sigma_i^j(t)$, representing the positional and heading uncertainty at each time step.

$$\Sigma_i^j(t) = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 \end{bmatrix} \quad (4.3)$$

where $\sigma_x, \sigma_y, \sigma_\theta$ are the variance in each of the state variables and $\sigma_{xy}, \sigma_{x\theta}, \sigma_{y\theta}$ are their covariances. The output of the Multi-modal Prediction Module is thus a set of probabilistic trajectories for each agent:

$$\mathcal{P}_i = \{\mathbf{x}_i^j(t), Pr(j), \Sigma_i^j(t)\}_{j=1}^{N_{traj}}, \quad t \in [t_0, t_0 + T] \quad (4.4)$$

This formulation captures both multi-modal uncertainty (different possible intentions, e.g., unexpected crossing, lane-keeping, lane-change, partially observed trajectories, sudden high-speed movements, or possible occluded motion) (cf. Figure 4.2) and aleatoric uncertainty (estimation noise or unpredictability within each mode) [81], [84].

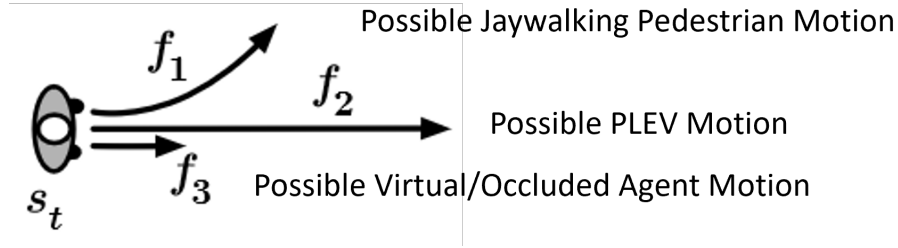


Figure 4.2: Multi-modal trajectory prediction

4.3.2 Role in the Architecture

The multi-modal prediction module provides probabilistic predictions of all surrounding agents, which serve as inputs to the F-sPIDP risk assessment (block “3” in Section 4.5). By propagating multiple hypotheses, the AV avoids premature pruning of possible agent behaviors.

Moreover, since the generated candidate trajectories of the ego vehicle (block “2” in Section 4.4) are evaluated against these multi-modal predictions, the accuracy and tuning of this module may influence the system’s balance between safety and non-conservatism.

4.4 VEHICLE TRAJECTORY GENERATION

Once the multi-modal predictions of surrounding agents are available, the AV must generate its own set of possible motions. These candidate trajectories serve as the ego vehicle’s motion hypotheses, from which safe and optimal motions can be later selected.

The trajectory generation process is not arbitrary: it must respect roadway constraints, follow traffic rules, and ensure comfort and feasibility. The proposed approach is to represent trajectories in the Frenet coordinate system (also known as the road-aligned frame), and decompose motion into longitudinal progress along a reference path (s) and lateral deviation from it (d). This representation simplifies lane-keeping and lane-changing maneuvers while aligning vehicle motion with the road geometry [201], [226].

By systematically generating a diverse set of candidate trajectories, the AV ensures that its downstream risk assessment module has sufficient options to evaluate, avoiding overcommitment to a single trajectory too early as compared to classical methods like MPC.

4.4.1 Problem Formulation

Let the reference path be parameterized as $\mathbf{p}_{ref}(s) = [x_{ref}(s), y_{ref}(s)]$, where s is the curvilinear distance along the path. The Frenet coordinates of the AV at time t are given by:

$$\mathbf{s}_{AV}(t) = [s_{AV}(t), d_{AV}(t), \theta_{AV}(t)] \quad (4.5)$$

where $s_{AV}(t)$ is the longitudinal position along the path, $d_{AV}(t)$ is the lateral offset from the path, and $\theta_{AV}(t)$ is the heading angle relative to the path tangent.

The trajectory generation problem is commonly posed as solving polynomial parameterizations for $s(t)$ and $d(t)$ over a finite horizon T :

$$s_{AV}(t) = \sum_{n=0}^N a_n t^n, \quad d_{AV}(t) = \sum_{m=0}^M b_m t^m \quad (4.6)$$

where a_n and b_m are polynomial coefficients to be determined, and N, M are the polynomial orders. The coefficients a_n, b_m are solved using constraints such as boundary conditions that are defined by the vehicle's initial state (position, velocity, acceleration) and terminal goals (desired lane, target velocity, lane-change offset) as shown in Section 6.4.

Candidate trajectories are generated by sampling a set of longitudinal and lateral profiles over the prediction horizon T . Each trajectory $\mathbf{s}_{AV}^k(t)$ for $k \in \{1, \dots, M\}$ is defined as:

$$\mathbf{s}_{AV}^k(t) = [s_{AV}^k(t), d_{AV}^k(t), \theta_{AV}^k(t)], \quad t \in [t_0, t_0 + T] \quad (4.7)$$

where:

- $s_{AV}^k(t)$ is generated by sampling different longitudinal velocity and acceleration profiles, ensuring compliance with speed limits and vehicle's dynamics.
- $d_{AV}^k(t)$ is generated by sampling lateral offsets corresponding to lane-keeping, lane-changing, or overtaking maneuvers, while respecting lane boundaries and road curvature.

- $\theta_{AV}^k(t)$ is derived from the path tangent and lateral offset, ensuring smooth heading transitions.

Each candidate trajectory is then transformed to their equivalent Cartesian coordinates for collision checking, risk assessment and management:

$$\mathbf{x}_{AV}^k(t) = [x_{AV}^k(t), y_{AV}^k(t), \theta_{AV}^k(t)], \quad t \in [t_0, t_0 + T] \quad (4.8)$$

where:

$$\begin{bmatrix} x_{AV}^k(t) \\ y_{AV}^k(t) \end{bmatrix} = \mathbf{p}_{ref}(s_{AV}^k(t)) + d_{AV}^k(t) \cdot \mathbf{n}_{ref}(s_{AV}^k(t)) \quad (4.9)$$

and $\mathbf{n}_{ref}(s)$ is the normal vector to the reference path at s .

The resulting candidate set is denoted as:

$$\mathcal{X}_{traj} = \{\mathbf{x}_{traj}^1, \mathbf{x}_{traj}^2, \dots, \mathbf{x}_{traj}^M\} \quad (4.10)$$

where M is the total number of trajectory samples. This set provides a diverse range of feasible motions for the AV, which can then be evaluated for safety and optimality in the subsequent F-SPIDP (block “3” in Section 6.2.4) risk assessment and trajectory selection modules.

4.4.2 Role in the Architecture

The candidate trajectory generation module (block “2” in Section 4.4) provides a set of feasible ego vehicle motions, which are then evaluated for collision risk using the Fusion of stochastic Predictive Inter-Distance Profiles (F-SPIDP) method (block “3” in Section 6.2.4). Hence it serves as a bridge between high-level navigation goals and low-level control execution:

- Inputs: AV state (pose, velocity), road geometry, traffic rules, and desired maneuvers.
- Outputs: a diverse pool of feasible, lane-disciplined trajectories.

The quality of this module is critical. If the candidate set is too narrow, the AV may miss safe or efficient options. If too broad, computational overhead increases for risk assessment and optimization.

Therefore, the following sections will highlight how the proposed F-SPIDP risk assessment framework and the subsequent local controller overcome these challenges, leveraging the generated candidate trajectories and the multi-modal predictions of surrounding agents.

4.5 F-SPIDP COLLISION RISK ASSESSMENT

Predictive Inter-Distance Profile (PIDP) is a metric used to assess the risk of collision between two or more traffic participants [17], [103]. The PIDP is essentially a projection of the inter-distance between the entities, assuming their future

trajectories are known and maintained during a specific time horizon, $t_{horizon}$. It is a risk assessment metric that can perform both time-scale and distance-scale measurements for continuous monitoring of the dangerous situations between two or more traffic participants over the given time horizon. PIDP is independent of the shape of the trajectories of the interacting agents [16], which makes it appropriate for situations involving PLEVs. In the context of PIDP, this means ensuring that the computed inter-distance between the ego vehicle and the PLEV is consistently maintained above a predefined safety distance, represented as d_{safe} (cf. Figure 4.3). Collision occurs when the PIDP is less than the desired safety distance between the ego-vehicle and PLEV (i.e., $PIDP(t) \leq d_{safe}$).

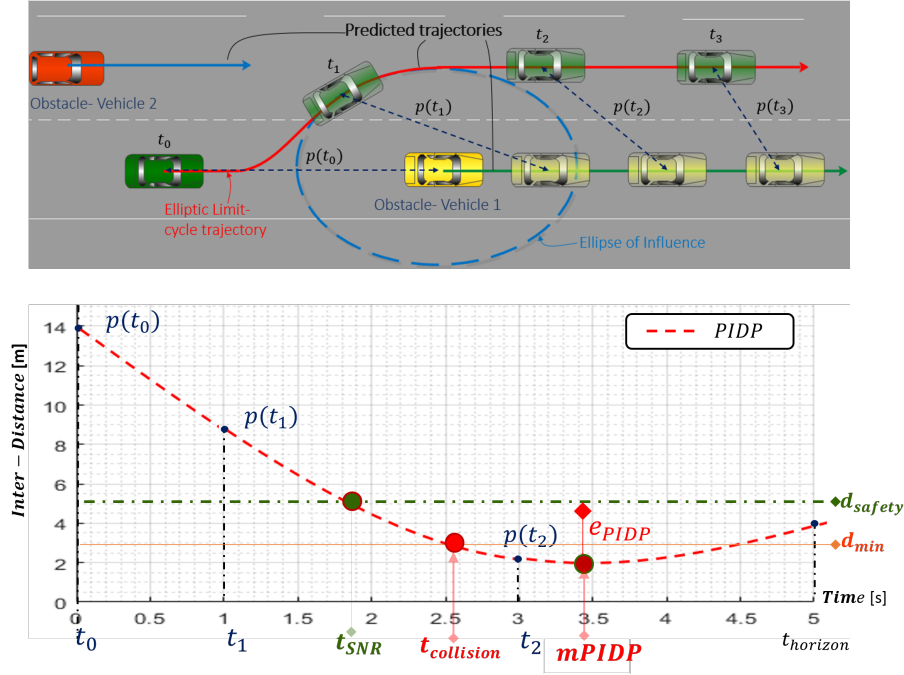


Figure 4.3: Predictive Inter-Distance Profile (PIDP) for risk assessment between vehicles (Cars) [103]

PIDP is flexible and has several features that make it suitable for risk assessment in mixed traffic scenarios (cf. Figure 4.3):

- $t_{collision}$: equivalent to TTC, it represents the time until a potential collision occurs if both entities maintain their current path and speeds.
- $minPIDP$: the minimum inter-distance between the entities over the time horizon, indicating how close they will get to each other.
- e_{PIDP} : the error in the PIDP, which is the difference between the desired safety distance and the minimum PIDP. It quantifies the extent to which the ego-vehicle deviates from maintaining a safe distance.
- t_{SNR} : the time when the PIDP first drops below the safety distance, indicating the onset of a potentially dangerous situation.

However, conventional risk assessment methods such as TTC and PIDP (cf. Section 3.5.1) often fail in complex urban scenarios where surrounding agents

(e.g., PLEVs, cyclists, pedestrians) exhibit highly uncertain and multi-modal behaviors. These methods tend to be either:

- Overly conservative, considering worst-case trajectories when uncertainty is high, or
- Overly optimistic, underestimating risks by discarding many feasible trajectories, ignoring variability in agent intentions.

Since the motion of the PLEV is assumed to be multi-modal with probabilities, the typical PIDP is therefore not sufficient for risk assessment in this context, hence the need for a fusion of multiple PIDPs arises. The Fusion of PIDP (F-PIDP) (cf. Section 5.2) integrates multiple PIDPs corresponding to different motion modes, effectively capturing the diversity of possible future trajectories. Moreover, as each mode's trajectory includes estimation uncertainties in the predicted states due to perception noise and modeling errors, a stochastic PIDP (sPIDP) (cf. Section 6.2) is developed to represent these uncertainties explicitly. To jointly address both multi-modal motion predictions and stochastic state uncertainties, a Fusion of stochastic PIDP (F-sPIDP) (cf. Section 6.2.4) is finally proposed, providing a unified and robust probabilistic framework for reliable PLEV risk assessment.

F-sPIDP (cf. Section 6.2.4) is a collision risk assessment module (block “3” in Figure 4.1) that serves as a critical intermediary between the candidate trajectory generation (cf. Section 4.4), the final trajectory selection (cf. Section 4.6), and control optimization (cf. Section 4.7). F-sPIDP method extends classical PIDP formulations by incorporating uncertainty distributions from the multi-modal prediction module (cf. Section 4.3). Instead of computing deterministic safety margins, F-sPIDP evaluates the fusion of probabilistic risk of collision with respect to each ego trajectory candidate, including its uncertainty. This provides a principled way to filter uncertain motions while avoiding excessive conservatism.

Its primary function is to evaluate the safety of each candidate trajectory generated in block “2” against the multi-modal predictions of surrounding agents from block “1”. By quantifying the collision risk associated with each trajectory, the F-sPIDP module enables the AV to make informed decisions that balance safety and efficiency (cf. Section 6.2.4 for details).

4.5.1 Problem Formulation

Let $\mathbf{x}_{traj}(t)$ denote a candidate trajectory of the ego vehicle, and $\mathbf{x}_i^j(t)$ the j -th predicted trajectory of surrounding agent i (cf. Section 4.3). The inter-distance profile between the ego vehicle and agent i along the candidate trajectory is defined as:

$$d_{ij}(t) = \|\mathbf{p}_{traj}(t) - \mathbf{p}_i^j(t)\| \quad (4.11)$$

where $\mathbf{p}(t) = (x(t), y(t))$ denotes position in Cartesian coordinates of the trajectories and $\|\cdot\|$ denotes the Euclidean distance (cf. Figure 4.3).

a) Fusion of PIDP

In the classical PIDP formulation, a candidate trajectory is considered safe if:

$$\min_{t \in [t_0, t_0+T]} d_{ij}(t) > d_{safe} \quad (4.12)$$

where d_{safe} is a predefined safety distance (e.g., minimum braking distance). If this condition is violated for any agent i and trajectory j , the trajectory is discarded.

The problem is that for each agent i , there are multiple predicted modes j with associated probabilities $Pr(j)$. Simply checking the minimum distance across all modes can lead to overly conservative behavior, because multi-modal trajectories imply multiple PIDPs (cf. Figure 4.4), when even low-probability modes or trajectories of the PLEVs are considered.

Therefore, a better approach is needed to fuse the risk across all modes and agents based on their likelihood.

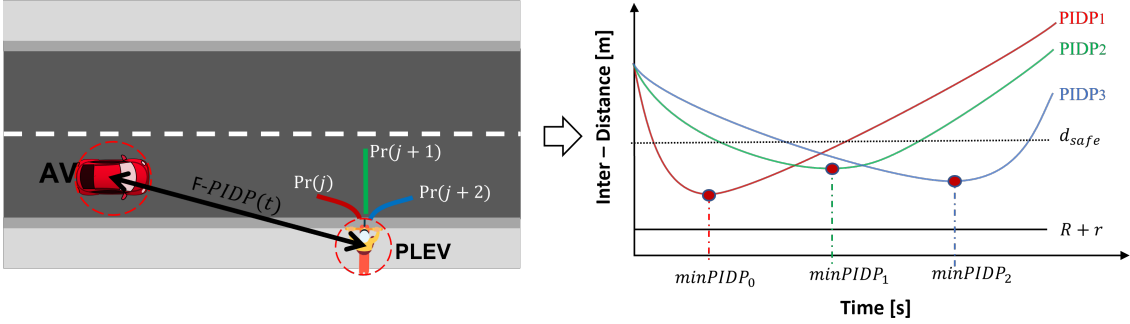


Figure 4.4: Ego vehicle and a Jaywalking PLEV with multi-modal trajectories: this implies multiple PIDPs for each agent and often leads to conservative behaviors

b) Stochastic PIDP (sPIDP) and its Fusion (F-sPIDP)

Stochastic PIDP (sPIDP) extends PIDP to account for the estimation uncertainties of predicted agent states in each trajectory (cf. Figure 4.5). The Fusion of sPIDP (F-sPIDP) then generalizes the approach to multi-modal trajectories: For each agent i , prediction modes $j = [1, \dots, N_{traj}]$ are weighted by $Pr(j)$.

Each $d_{ij}(t)$ is treated as a random variable due to trajectory uncertainty $\Sigma_i^j(t)$. Thus, the expected output of this module is the filtered set:

$$\mathcal{X}_{safe} = \left\{ \mathbf{x}_{traj} \in \mathcal{X}_{traj} \mid \min_{t \in [t_0, t_0+T]} d_{ij}(t) > d_{safe} \right\} \quad (4.13)$$

The F-sPIDP framework differs from start-of-the-art methods like TTC or PIDP by explicitly fusing multi-modal prediction distributions with predictive distance profiles, thereby integrating both intention uncertainty, trajectory variability, and state estimation uncertainties. The details of stochastic PIDP (sPIDP) and its fusion (F-sPIDP) are given in Section 6.2.4.

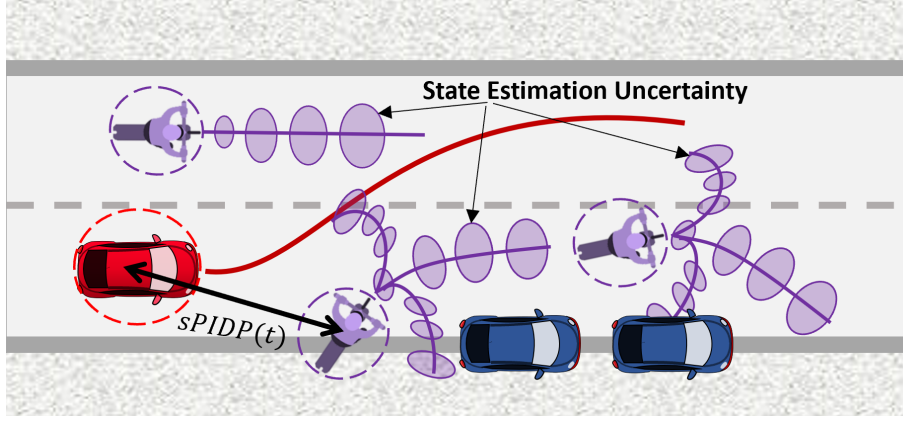


Figure 4.5: Multimodal predictions with state estimation uncertainties

4.5.2 Role in the Architecture

The F-sPIDP collision risk assessment module (block “3”) serves as a critical intermediary between the candidate trajectory generation (block “2”) and the optimal trajectory selection module (block “4”). Its primary function is to evaluate the safety of each candidate trajectory against the multi-modal predictions of surrounding agents, thereby filtering out unsafe options before final selection, the definition of the most suitable set-points, and the control.

- Inputs: candidate ego trajectories $\mathcal{X}_{traj}$ (from Section 4.4) and multi-modal predictions $\mathbf{x}_i^j(t)$ with probabilities (from Section 4.3).
- Outputs: safe set $\mathcal{X}_{safe}$, containing only trajectories that satisfy probabilistic risk bounds.

This step ensures that subsequent trajectory selection operates only on risk-feasible options, thereby reducing the likelihood of unsafe plans being executed.

4.6 OPTIMAL TRAJECTORY SELECTION

The optimal trajectory selection module (block “4” in Figure 4.1) is responsible for choosing the best candidate trajectory from the safe set $\mathcal{X}_{safe}$ produced by the F-sPIDP collision risk assessment (block “3”). This selection process aims to choose one optimal reference trajectory (a global path) that balances multiple objectives, including safety, efficiency, and comfort, while aligning with the long-term navigation objectives.

This module is essential because, while multiple safe options may exist, not all are equally desirable. For example:

- One trajectory may be safe but inefficient (e.g., too slow).
- Another may be efficient but uncomfortable (e.g., involving sharp accelerations).

- Yet another may be safe and smooth, but deviate from the navigation goal (e.g., wrong lane for the upcoming turn).

Thus, multi-objective optimization is required to select the “best” trajectory according to a weighted set of criteria.

4.6.1 Problem Formulation

Let $\mathcal{X}_{safe} = [\mathbf{x}_{traj}^1, \mathbf{x}_{traj}^2, \dots, \mathbf{x}_{traj}^K]$ denote the set of candidate trajectories. Each trajectory is evaluated using a cost function $J(\mathbf{x}_{traj})$, which aggregates several objectives [6] (cf. Section 6.4):

$$J(\mathbf{x}_{traj}) = w_s J_{smooth} + w_c J_{comfort} + w_e J_{efficiency} \quad (4.14)$$

where:

J_{smooth} : penalizes large curvature, jerk, or lateral oscillations.

$J_{comfort}$: penalizes high acceleration or sudden steering changes.

$J_{efficiency}$: encourages progress along the route and higher speeds (within safe limits).

w_s, w_c, w_e : weighting coefficients tuned to balance objectives.

The optimal trajectory is selected as:

$$\mathbf{x}_{ref} = \arg \min_{\mathbf{x}_{traj} \in \mathcal{X}_{safe}} J(\mathbf{x}_{traj}) \quad (4.15)$$

This process ensures that the chosen reference trajectory is not only safe but also well-aligned with operational and comfort criteria.

4.6.2 Role in the Architecture

- Inputs: Risk-feasible candidate trajectories $\mathcal{X}_{safe}$ from F-sPIDP.
- Output: One reference trajectory $\mathbf{x}_{ref}$ as vehicle’s set-points.

This stage acts as the decision-making bridge between global safety evaluation and local control execution. Narrowing down to a single reference trajectory provides clarity and consistency for the control system.

4.7 LOCAL CONTROL OPTIMIZATION

The local control optimization module (block “5” in Figure 4.1) is responsible for computing the actual control inputs (e.g., steering angle, velocity, or acceleration) required to track in real time, the selected reference trajectory $\mathbf{x}_{ref}$ from the optimal trajectory selection module (block “4”). This module ensures that the AV follows the desired global path while respecting vehicle dynamics, actuator limits, and safety constraints.

A key challenge is that both the reference trajectory and the surrounding environment are subject to uncertainties. Thus, the controller must be robust against disturbances, modeling errors, and trajectory deviations. Standard controllers, such as PID, Stanley, or MPC methods, are insufficient in complex urban settings with PLEVs and other multi-modal agents. Instead, uncertainty-aware optimization-based control (e.g., Stochastic Model Predictive Control (SMPC), robust control) is required.

4.7.1 Problem Formulation

Let the AV's state at time t be represented as $\mathbf{x}_{AV}(t)$:

The vehicle dynamics can be modeled as:

$$\dot{\mathbf{x}}_{AV}(t) = f(\mathbf{x}_{AV}(t), \mathbf{u}(t)) + \mathbf{w}(t) \quad (4.16)$$

where $\mathbf{u}(t) = [a(t), \delta(t)]$ are the control inputs (acceleration and steering angle), and $\mathbf{w}(t)$ represents process noise or disturbances.

The control objective is to minimize the deviation from the reference trajectory $\mathbf{x}_{ref}(t)$ over a prediction horizon T while respecting constraints:

$$\min_{\mathbf{u}(t)} \int_{t_0}^{t_0+T} \|\mathbf{x}_{AV}(t) - \mathbf{x}_{ref}(t)\|^2 + \lambda \|\mathbf{u}(t)\|^2 dt \quad (4.17)$$

subject to:

- Vehicle dynamics: $\dot{\mathbf{x}}_{AV}(t) = f(\mathbf{x}_{AV}(t), \mathbf{u}(t)) + \mathbf{w}(t)$
- Control limits: $a_{min} \leq a(t) \leq a_{max}$, $\delta_{min} \leq \delta(t) \leq \delta_{max}$, where a_{min} , a_{max} , δ_{min} and δ_{max} are the minimum and maximum allowable acceleration and steering angles.
- Safety constraints: ensuring a minimum distance from obstacles or other agents, derived from F-sPIDP outputs.

This optimization can be solved using techniques such as SMPC, which explicitly accounts for uncertainties in the vehicle dynamics and environment.

4.7.2 Role in the Architecture

- Inputs: selected reference trajectory $\mathbf{x}_{ref}$ from block "4" and current vehicle state $\mathbf{x}_{AV}(t_0)$.
- Outputs: Control inputs $\mathbf{u}(t)$ to be applied to the vehicle actuators.
- Execution: the computed control inputs are sent to the vehicle's low-level controllers (e.g., throttle, brake, steering). This module closes the loop by ensuring that the high-level decision (reference trajectory) is executed reliably at the vehicle level.

This module serves as the final execution step, translating high-level planning decisions into actionable commands that ensure the AV follows the desired path safely and efficiently.

4.8 CONCLUSION

This chapter presented a comprehensive architecture for Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) in AVs. The system integrates perception, multi-modal motion prediction, risk assessment using Fusion of stochastic PIDP (F-sPIDP), sampling-based candidate trajectory generation, optimal trajectory selection, and local control optimization. Each module plays a critical role in ensuring that the AV can navigate safely and efficiently in complex urban environments with vulnerable road users such as pedestrians and Personal Light Electric Vehicles (PLEVs). By leveraging multi-modal predictions, probabilistic risk assessment, and uncertainty-aware control, the proposed architecture, MiRA-MOC, addresses key challenges in autonomous driving, including handling uncertain agent behaviors, balancing safety and efficiency, and executing robust control actions. The F-sPIDP module, in particular, provides a principled way to evaluate collision risks across multiple hypotheses, enabling the AV to make informed decisions without excessive conservatism.

The following chapters will dive into the derivation of the F-PIDP and F-sPIDP methods, which are the core of the proposed architecture, presenting their mathematical formulation and their integration with Stochastic Model Predictive Control (SMPC) schemes to achieve real-time risk-aware navigation in mixed traffic scenarios.

This chapter introduces a framework for multi-risk assessment and management of autonomous vehicles interacting with uncertain micromobility agents (PLEVs). Building on the Predictive Inter-Distance Profile (PIDP), a Fusion of PIDP (F-PIDP) is developed to integrate multi-modal predictions and uncertainty into smooth, risk-weighted safety profiles. It is then integrated into a Stochastic MPC (SMPC) for risk management. A priority-based target selection strategy addresses multi-risk situations. Simulations demonstrate improved safety, efficiency balance and reduced conservatism compared with baseline predictive control methods.

5.1 INTRODUCTION

Risk assessment in autonomous driving relies on estimating the likelihood and severity of future collisions with surrounding agents. In environments such as highways, classical surrogate safety metrics, including Time-to-Collision (TTC), Time-Exposed TTC (TET), and Post-Encroachment Time (PET), provide useful safety indicators [205] (cf. Section 3.5.1), but they only capture instantaneous or deterministic aspects of risk. However, in urban settings populated with Personal Light Electric Vehicles (PLEVs), e-scooters, bicycles, and other micromobility agents, the situation becomes considerably more complex. PLEVs are lightweight, exhibit highly diverse motion patterns, and are strongly influenced by human intent, making their trajectories inherently multi-modal and uncertain.

To address this challenge, Predictive Inter-Distance Profile (PIDP) [103] has been introduced as a functional safety metric (cf. Section 4.5). Unlike TTC, which outputs a scalar collision time, PIDP describes the evolution of inter-vehicle distance over the entire prediction horizon. PIDP generalizes TTC by accounting for spatial and temporal variations. PIDP is generic, independent of the path geometry, and flexible with several features for risk assessment.

The difficulty arises when multi-modal trajectory predictions are considered. Each mode produces its own PIDP curve. Treating them independently leads to conservative behavior (cf. Figure 5.1): the AV would assume the worst-case trajectory at all times, which causes unnecessary braking or hesitation. To mitigate this conservatism, the Fusion of PIDPs (F-PIDP) is proposed as a fusion method that integrates multi-modal PIDPs into a single representative curve using the likelihood of each trajectory.

Finally, we embed F-PIDP into Stochastic Model Predictive Control (SMPC). These control schemes enforce safety constraints derived from fused PIDPs, balancing collision avoidance with efficiency.

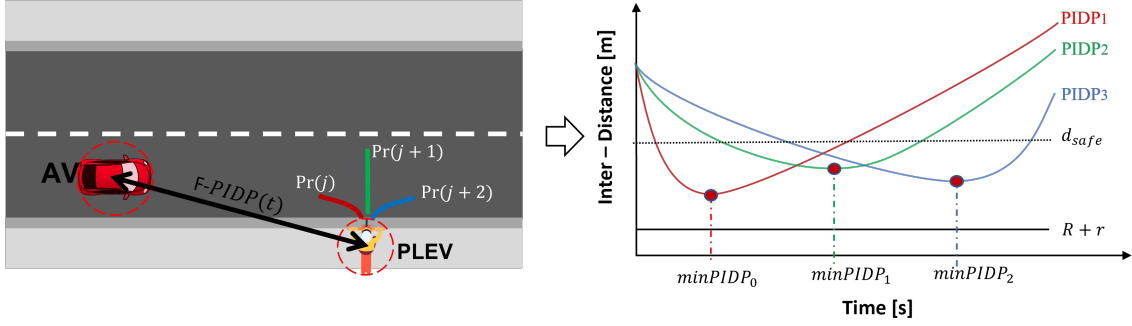


Figure 5.1: Ego vehicle and a Jaywalking PLEV with multi-modal trajectories: this implies multiple PIDPs for each agent and often leads to conservative behaviors

5.2 FUSION OF PREDICTIVE INTER-DISTANCE PROFILE (F-PIDP)

Multi-modal model of pedestrian motion implies that there are multiple PIDPs for each pedestrian, either due to the uncertainty in the intended trajectories or the true velocity of the PLEV (cf. Figure 5.1). This increases the complexity of the navigation problem since each predicted trajectory is equivalent to a virtual PLEV with a probability of existence. The authors in [16], [103], similar to many research studies, solely focus on a single trajectory, which is equivalent to selecting only the most probable trajectory, that is, the trajectory with the highest probability. However, considering a single trajectory often fails because PLEV can rapidly switch its behaviors, such as accelerating its movement. Another control strategy is to only consider the most dangerous trajectory, that is, the trajectory with the smallest inter-distance. Such a strategy would lead to highly conservative behavior. We propose to solve this multi-risk problem by performing a weighted fusion of the PIDPs while using a quadratic polynomial curve with probabilistic weights. Accordingly, we obtain a Fusion of PIDPs (F-PIDP). The following subsections give a detailed explanation of the steps for computing the F-PIDP.

To compute the F-PIDP between an AV and a PLEV or other road users (cf. Figure 5.1 and 5.2):

1. Compute the PIDP for all j^{th} traj (cf. Section 5.2.1)
2. Compute the Start-point (t_0), minPIDP (t_{min}), End-point (t_N) of all the PIDPs (cf. Section 5.2.2)
3. Compute probabilistic Center of Mass (pCOM) (cf. Section 5.2.3)
4. Compute the parameters q_0, q_1, q_2 for F-PIDP (cf. Section 5.2.4)
5. Compute F-PIDP_{ref} setpoints (cf. Section 5.2.5)

The following sections explain how the steps above are implemented.

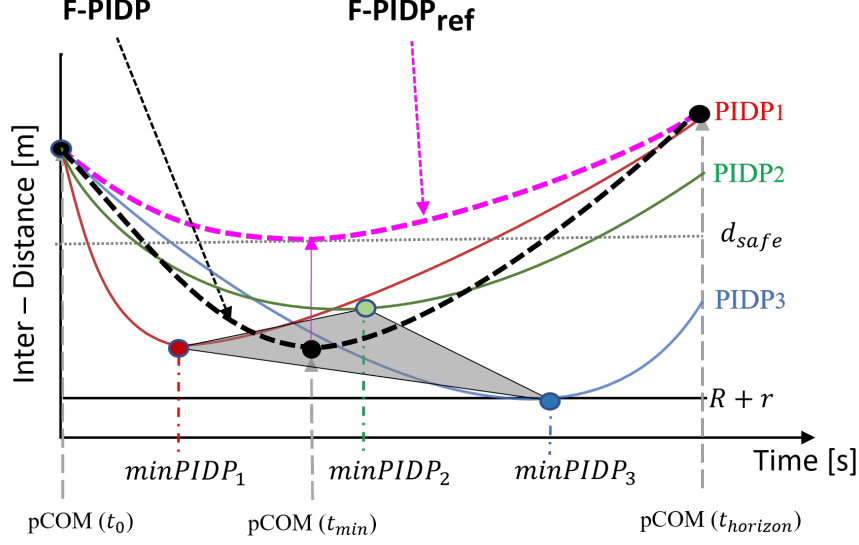


Figure 5.2: Fusion of PIDPs (F-PIDP) of an AV and a PLEV with three 3 possible multi-modal trajectories

5.2.1 Computation of each PIDP

The first step is to compute the PIDP of each probable trajectory. Given a set of possible future trajectories of a PLEV, $\mathcal{P} = \{p_1, p_2, \dots, p_m\}$. The aim is to compute the magnitude of the distance vector between the future position of the ego-vehicle and the position of the PLEV at each time step t , if $r(t) = [x(t), y(t)]^T$ and $r_{p_j}(t) = [x_{p_j}(t), y_{p_j}(t)]^T$ are the position vectors of the vehicle and the j^{th} PLEV trajectory respectively. Mathematically, the PIDP for a single trajectory can be expressed as:

$$PIDP_j(t) = \{d_j(t) | p_j \in \mathcal{P}, 0 \leq t \leq t_{horizon}\} \quad (5.1)$$

$$d_j(t) = ||r(t) - r_{p_j}(t)|| \quad (5.2)$$

where $PIDP_j$ denotes the inter-distance between the ego-vehicle and PLEV's j^{th} trajectory, obtained by calculating their Euclidean norm $|| \cdot ||_2$.

5.2.2 Parameterization of each PIDP

In general, the inter-distance profile is a nonlinear function due to the changing motion of the ego-vehicle and PLEV. Hence, important characteristics of each PIDP are extracted to perform a fusion of the PIDPs and maintain a smooth curve. The extracted parameters from each $PIDP_j$ are (cf. Figure 5.2):

- $PIDP_j(t_0)$ represents the start point of the j^{th} PIDP, which is the same for all probable trajectories
- $minPIDP_j(t_{min})$ the minimum of each PIDP at time t_{min}
- $PIDP_j(t_{horizon})$ the PIDP at the end of each predicted trajectory

5.2.3 Probabilistic Center of Mass

The goal is to compute the probability-based center of mass (pCOM), which is a fused estimate of the inter-distance across N_{traj} trajectories. The approach is expected to generalize to various trajectory estimators including the physics-based and learning-based motion prediction algorithms (cf. Section 2.4.1). To achieve this, we recall that an IMM filter (cf. Section 2.4.1 (a)) is a recursive Bayesian estimator designed to handle systems with multiple possible modes (or models). The IMM filter maintains a bank of filters, each corresponding to a different mode, and combines their outputs according to mode probabilities. In this context, the modes are interpreted as corresponding to different trajectories, with each trajectory j providing an estimated inter-distance $d_j(t)$ at time t .

- Mode-Matched Filtering: for each trajectory $j = [1, 2, \dots, N_{traj}]$, a mode-matched algorithm (e.g., a Kalman filter / GAN tuned to that trajectory's dynamics, cf. Section 2.4.1) produces an estimate of the (trajectory or) inter-distance $d_j(t)$ at time t . This is the output of the individual filter for mode j .
- Mode Probabilities: an IMM filter computes the posterior probability $\Pr(j)$ (often denoted $\mu_j(t)$) that trajectory j is the "active" or correct mode at time t . These probabilities are updated recursively using Bayes' rule, based on the likelihood of the measurements under each mode:

$$\Pr(j | z^t) = \frac{\pi_j(t) \cdot \Pr(j | z^{t-1})}{\sum_{i=1}^{N_{traj}} \pi_i(t) \cdot \Pr(i | z^{t-1})} \quad (5.3)$$

where z^t is the measurement history up to time t , and $\pi_j(t)$ is the likelihood of the measurement at time t under mode j . Note that $\sum_{j=1}^{N_{traj}} \Pr(j) = 1$.

- Interaction (Mixing): prior to updating at time t , a mixture of the state estimates and covariances from the previous time step across the multi-modal predictions is obtained using mixing probabilities derived from the mode transition matrix π_{ij} (probability of switching from mode i to j) and prior mode probabilities. The mixed estimate for mode j is:

$$\hat{d}_{0j}(t-1) = \sum_{i=1}^{N_{traj}} \mu_{i|j}(t-1) \hat{d}_i(t-1) \quad (5.4)$$

$$\text{where } \mu_{i|j}(t-1) = \frac{\pi_{ij} \Pr(i|z^{t-1})}{\sum_{i=1}^{N_{traj}} \pi_{ij} \Pr(i|z^{t-1})}.$$

- **Combined Output (Fusion):** the step above accounts for mode switching. The key equation for the overall fused estimation is the probabilistic mixture of the mode-conditioned estimates, weighted by their probabilities. Treating the inter-distance as the state variable, the fused estimate at time t is the expected value under the posterior mode distribution:

$$\hat{d}(t) = \mathbb{E}[d(t) | z^t] = \sum_{j=1}^{N_{traj}} \Pr(j | z^t) \cdot d_j(t) \quad (5.5)$$

This is the probability-weighted average, analogous to a center of mass in a system where masses are replaced by probabilities and positions by distances. The fused estimate $\hat{d}(t)$ is defined as the probability-based center of mass:

$$pCOM(t) = \sum_{j=1}^{N_{traj}} \Pr(j) \cdot d_j(t) \quad (5.6)$$

The pCOM provides a soft fusion, avoiding hard decisions on a single trajectory and incorporating uncertainty via the probabilities.

5.2.4 PIDP fusion using probabilistic weights

The fusion step in the F-PIDP process combines multiple PIDPs into one representative prediction based on their level of uncertainty. In an ideal situation where the motion or velocity profile of the AV and PLEV follows a smooth path, the PIDP is expected to be a polynomial curve derived from the Euclidean distance (cf. Figure 5.1). Therefore, to simplify the curve over a short term horizon, the parameters of all the PIDPs are approximated together to form a single quadratic polynomial curve. From the equations of a quadratic polynomial:

$$d(t) = a_0 + a_1 t + a_2 t^2 \quad (5.7)$$

$$\begin{bmatrix} d(t_0) \\ \vdots \\ d(t_N) \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 \\ \vdots & \vdots & \vdots \\ 1 & t_N & t_N^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \quad (5.8)$$

where $d(t)$ denotes the PIDP at time t and $\{a_0, a_1, a_2\} \in \mathbb{R}$ are parameters of the polynomial curve to be calculated. The parameters of this curve are determined by calculating the probabilistic center of mass (pCOM) of all PIDPs. From the pCOM equation in (5.6), F-PIDP is defined as:

$$F_PIDP(t) = \sum_{j=1}^{N_{traj}} Pr(j) \times d_j(t) \quad (5.9)$$

where, $N_{traj} \in \mathbb{N}$ denotes the number of predicted trajectories, $Pr(j)$ and d_j denote the probability and inter-distance of the j^{th} predicted trajectory of the PLEV at the desired time t (i.e., start, minimum, and endpoint) of the PIDPs. This center of mass is a weighted average that considers the likelihood of each trajectory without a specific assumption on the source of the predicted trajectory (model-based or data-based prediction). The desired control points to find the parameters of the polynomial are the fusion of PIDP (F-PIDP):

- $F_PIDP(t_0)$ fusion of the initial points $PIDP_j(t_0)$, which is the same for all PIDPs
- $F_PIDP(t_{min})$ the fusion of all the minimum PIDPs $minPIDP_j(t_{min})$ based on pCoM
- $F_PIDP(t_{horizon})$ the fusion of all the final points $PIDP_j(t_{horizon})$

Since the fusion of the feature points $d(t) = F_PIDP(t)$ in (5.8) are known, the parameters of the fusion $\{a_0, a_1, a_2\}$ can be found using the matrix (or pseudo) inverse. The resulting parameters are then substituted in (5.7) to get a single unified F-PIDP curve that encapsulates the combined information as shown in Figure 5.2.

5.2.5 PIDP Setpoint based on Safety Distance

After the fusion process, the lower bound of the curve may fall below the desired safety distance (cf. Figure 5.2), thereby indicating a potential risk of collision. To ensure collision avoidance within the proposed multi-risk assessment framework, the required F_PIDP_{ref} setpoint for the ego-vehicle is determined by translating, when necessary, the minimum point of the F-PIDP curve according to the following procedure:

- The initial and final points of the F-PIDP curve remain unchanged.
- The minimum point of the F-PIDP, denoted as m_{F-PIDP} is translated to a value equal to or greater than the maximum safety distance, such that:

$$F_PIDP(t_{min}) \triangleq \max\{F_PIDP(t_{min}), d_{safe_{max}}\} \quad (5.10)$$

$$d_{safe_{max}} \triangleq R + r + v_{max} \times ETTC(1sec) \quad (5.11)$$

where t_{min} is the time when F-PIDP has a minimum, $d_{safe_{max}}$ is the maximum safety distance derived from the maximum velocity of the ego vehicle v_{max} and the extended time to collision (ETTC) of 1 seconds. R and r correspond to the radius of the circle that surrounds the ego vehicle and the PLEV, respectively.

This adjustment is motivated by the requirement that the future motion of the ego-vehicle minimizes the deviation of the F-PIDP curve from the safety distance. Alternatively, one could consider translating the tSNR or the TTC metrics. However, this approach can induce more aggressive control behavior due to the inherent uncertainty in the motion of the PLEV. Consequently, translating the minimum of the F-PIDP curve offers a more straightforward implementation while preserving smooth vehicle motion, as demonstrated in Section 5.3.

The new curve obtained after the translation (cf. Figure 5.2) can serve as a reference to a SMPC algorithm (cf. Section 5.3) and guarantees that the predicted trajectory maintains a safe separation from potential obstacles. By establishing a set-point equal to or greater than the safety distance, collision risks are minimized. This step is therefore crucial for safe and proactive navigation.

5.3 F-PIDP IN STOCHASTIC MODEL PREDICTIVE CONTROL (SMPC)

This section presents how F-PIDP is integrated in an SMPC planner (cf. Section 3.4.4). Classical SMPC [154], [160] enforces safety constraints for every predicted mode, which yields excessive conservatism and many collision constraints in (5.15). Instead, we enforce the explicit safety constraint only for the single most probable predicted trajectory (the mode with highest probability). The $F-PIDP_{ref}$ is used as a smooth, time-varying safety setpoint: deviations from this setpoint are penalized in the SMPC cost to produce anticipatory, smooth evasive actions (5.17). This design reduces conservatism while respecting the safety constraint.

5.3.1 SMPC for Risk Management

To validate the F-PIDP risk assessment method, SMPC is proposed to perform risk management by computing the optimal control actions of the ego-vehicle while accounting for various constraints and uncertainties in the motion of the traffic agents. SMPC is formulated as a chance-constraint problem, where the objective is to minimize a cost function that balances reference pose reaching and control effort. Unlike MPC, SMPC accepts a tunable parameter ensuring that the probability of constraint violation remains below a specified threshold. The optimization problem is defined as follows:

$$\min_{\mathbf{U}} \mathcal{J}(\mathbf{x}_0, \mathbf{x}_{ref}, \mathbf{U}) + \mathcal{J}_p(\mathbf{x}_0, \mathbf{x}_{ref}, \mathbf{U}) \quad (5.12)$$

subjected to:

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad \forall t \in [t_0, t_0 + T], \quad (5.13)$$

$$g(\mathbf{x}(t+1), \mathbf{u}(t)) \leq \mathbf{0} \quad \forall t \in [t_0, t_0 + T], \quad (5.14)$$

$$Pr[\mathbf{x} \in \mathcal{S}] \leq 1 - \alpha \quad \forall t \in [t_0, t_0 + T] \quad (5.15)$$

where the aim of the SMPC is to optimize the objective costs $\mathcal{J}$ and $\mathcal{J}_p$. The dynamics of the ego vehicle (5.13) is used to predict the future state of the vehicle $\mathbf{x}(t+1)$, over the prediction horizon $t \in [t_0, t_0 + T]$ given the current condition $\mathbf{x}_0$ and the future control inputs $\mathbf{u}(t) \in \mathbf{U}$. Constraint (5.14) are inequality constraints such as limits on the minimum and maximum control inputs. The chance constraint in (5.15) represents the probabilistic conditions to be satisfied over the future motion of the vehicle given the safe set $\mathcal{S}$. The SMPC algorithm will guarantee that predefined levels of probability $0 < \alpha < 1$ are met for uncertainties in the outcome. Therefore, the design parameter α is referred to as the safety level. $\mathcal{J}$ is the objective cost to reach the desired goal and velocity smoothly. Here, we define $\mathcal{J}_p$ as the cost to reach the desired F-PIDP set-points, such that:

$$\mathcal{J} = \sum_{t=0}^T \left(\|\Delta \mathbf{x}(t)\|_Q^2 + \|\mathbf{u}(t)\|_R^2 \right) + \|\Delta \mathbf{x}(T)\|_S^2 \quad (5.16)$$

$$\mathcal{J}_p = \sum_{t=0}^T \|d(t) - F_PIDP_{ref}(t)\|_P \quad (5.17)$$

where $\mathcal{J}$ is the classical SMPC cost to reach the reference state $\Delta \mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_{ref}(t)$, with weighting matrices $Q, S \in \mathbb{R}^{3 \times 3}$ and $R \in \mathbb{R}^{2 \times 2}$. $d(t)$ is the inter-distance after applying the new control inputs $\mathbf{u}(t) \in \mathbf{U}$, while $F_PIDP_{ref}(t)$ is the reference safety distance (in Section 5.2). The variable $P \in \mathbb{R}$ penalizes the extent to which the reference cost is minimized. By minimizing $\mathcal{J}$ and $\mathcal{J}_p$, SMPC is able to ensure comfortable navigation while minimizing collision risk. It is important to mention that negligible uncertainties are considered in the state of the PLEV using the SMPC algorithm in order to focus on analyzing the contributions of the proposed F-PIDP_{ref} to multi-risk assessment and management during autonomous navigation. The objective cost using F-sPIDP_{ref} together with the constraint (5.15) serves as adaptive constraints for the AV motion, instead of the hard constraints when using MPC.

5.3.2 Road Boundary Constraint

The position of the AV is expected to stay within the lane of the road (cf. Figure 5.4), such that the following constraint is satisfied,

$$y_{min}^{lane} \leq y^{AV}(t) \leq y_{max}^{lane} \quad (5.18)$$

This ensures that the lateral position of the AV denoted $y^{AV}(t)$ at each time t , is bounded by y_{min}^{lane} and y_{max}^{lane} based on the width of the road and the vehicle.

5.3.3 Control Input Constraint

The motion of the AV is also constrained by the physical limits on the velocity, acceleration, and steering angle of the vehicle's wheels. These limits also have a direct influence on the comfort of the passengers. They are formulated as

$$\mathbf{u}_{min}^{AV} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}^{AV} \quad (5.19)$$

where the vector $\mathbf{u}_{min}^{AV}$ and $\mathbf{u}_{max}^{AV}$ are respectively the bounds on the minimum and maximum values of the control inputs.

5.3.4 F-PIDP+SMPC Simulation Results and Discussion

The proposed F-PIDP approach is evaluated through the simulation of two scenarios involving an ego-vehicle and a jaywalking PLEV. The initial state of the ego-vehicle is set to be 15 meters away from the PLEV, with a linear velocity of 5 m/s and 8 m/s, respectively. In both scenarios, it is assumed that the ego-vehicle does not have enough time to brake and is expected to perform a swerving maneuver in order to avoid collision with PLEV, ensure the comfort of passengers, and also prevent exhibiting highly conservative behaviors. (cf. **video shown through this link** <https://youtu.be/IS0qhT0YbGc?list=PL0zhSFDHifx2ruWjKJ1gggCKfnzertLLu>)

5.3.5 Scenario 1: Jaywalking PLEV Switching from Slow to Fast motion

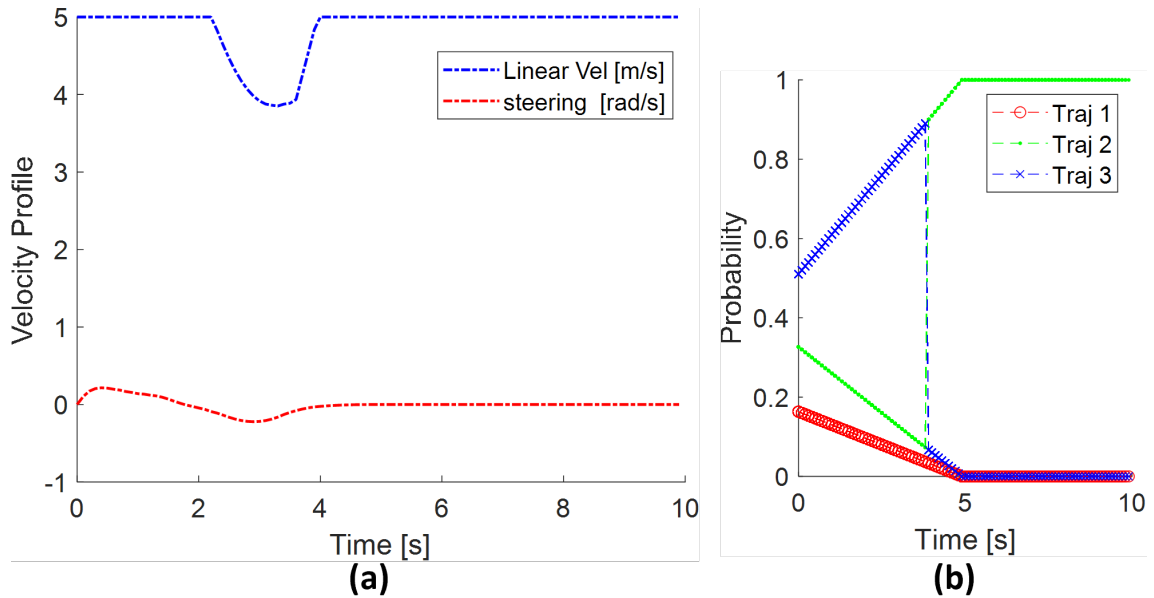


Figure 5.3: Scenario 1: (a) Velocity plot showing a smooth linear velocity and steering (b) Probability plot with a switching from Traj 3 (blue) to Traj 2 (green) after $t = 3.9s$

In the first scenario, a jaywalking PLEV begins to cross the road at an initial linear velocity of $0.5m/s$ and at an angle equivalent to a right turn (cf. Traj 3 in

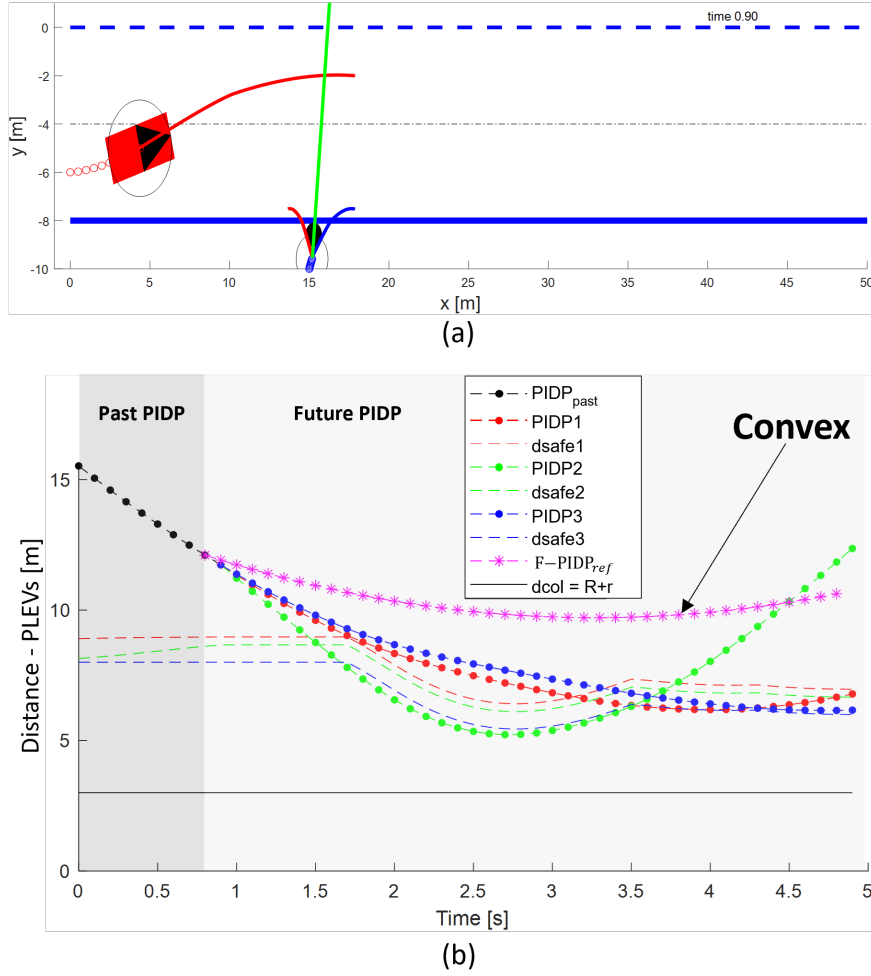


Figure 5.4: Scenario 1 at $t = 0.9$ s (a) Jaywalking PLEV moving slowly (b) Inter-distance plot of ego-vehicle and PLEV: The F-PIDP_{ref} plot exhibits a convex curve, indicating current safety is satisfactory. However, the steep curve suggests a high risk of future safety distance violations due to PIDP₂

Figure 5.4). After approximately 3.9 s, there is an abrupt change in the PLEV's behavior, and the velocity increases to 5 m/s (cf. Traj 2 in Figure 5.5). The progress of the probability of each trajectory is shown in Figure 5.3 b. The sudden switch in behavior is a dangerous situation that could lead to a collision if the ego-vehicle predicts the PLEV's motion solely based on the most probable trajectory. This sudden change in motion from Traj 3 (blue) to 2 (green) (probability plot cf. Figure 5.3 (b)) introduces uncertainty and necessitates an anticipatory assessment of the situation, which is achieved using the F-PIDP. Observe that at $t = 0.9$ s the F-PIDP plot (cf. Figure 5.4) exhibits a convex curve, indicating current safety is satisfactory.

However, the high position of the curve from the PIDPs suggests a high risk of future safety distance violations due to Traj 2(green), since it has the minimum PIDP and hence the most dangerous trajectory (cf. Figure 5.4). The shape of the F-PIDP results from the fact that Traj 2 has a high probability (cf. Figure 5.3 (b)), hence more influence on the position of the pCOM of all the predicted trajectories.

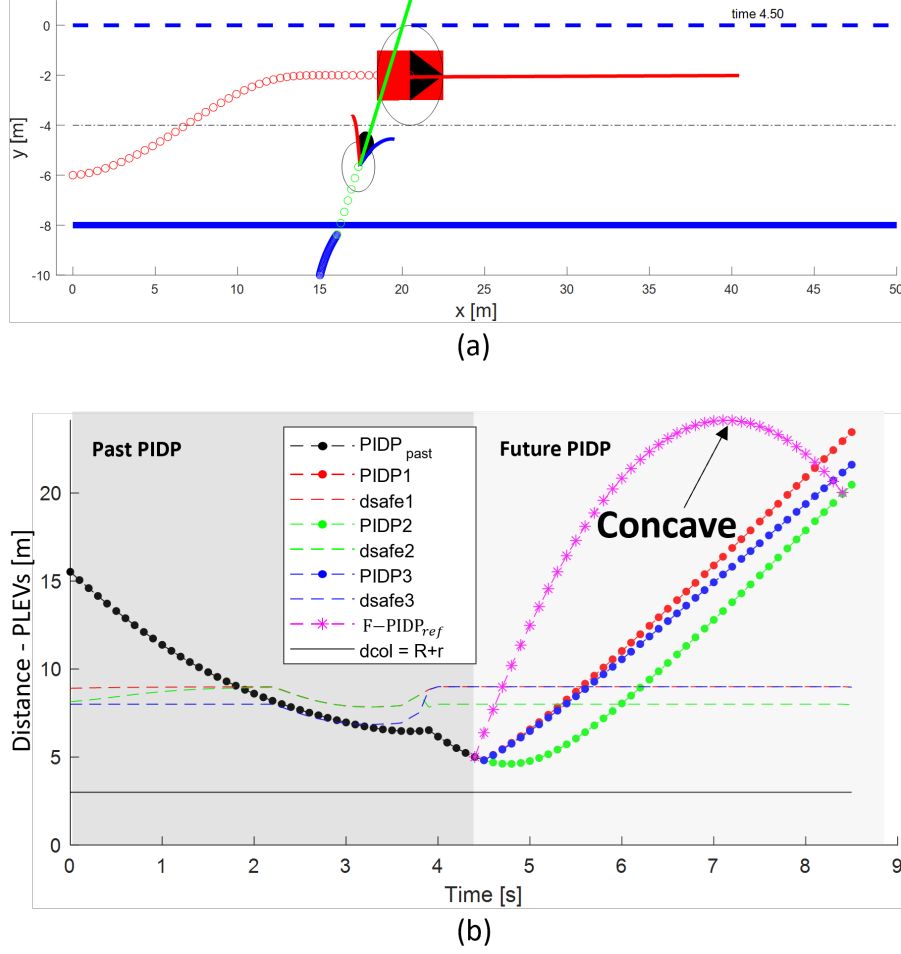


Figure 5.5: Scenario 1 at $t = 4.5s$ (a) Jaywalking PLEV moving Fast (b) Inter-distance plot of ego-vehicle and PLEV: As the vehicle moves closer to the PLEV, the F-PIDP set-point takes a concave form even before Traj 2 enters the collision zone d_{col}

As the vehicle moves closer to the PLEV, the F-PIDP takes a concave form even before Traj 2 enters the collision zone (cf. Figure 5.5). This is because the current inter-distance between the AV and the PLEV falls below d_{safe} , translating the minimum point of the $F-PIDP_{ref}$ curve upward to exceed d_{safe} (as described in Section 5.2.5) introduces a singularity in the solution of the quadratic equation (5.7). This singularity leads to the concave profile of the resulting $F-PIDP_{ref}$ curve. The concave form increases the objective cost $\mathcal{J}_P$ (cf. Section 5.3.1) of the SMPC, thereby suggesting a solution that quickly moves the ego-vehicle away from the collision zone. Therefore, collision is avoided with a minimum inter-distance of approximately 1.6 meters away from the PLEV (cf. Figure 5.6). The ego-vehicle does not decrease its velocity at the beginning because the most dangerous trajectory (Traj 2) has a low probability, hence the velocity plot (cf. Figure 5.3 (a)) shows a smooth velocity profile for the passenger's comfort without being overly conservative (e.g., stopping) even when the ego-vehicle decelerates and then accelerates to avoid a collision.

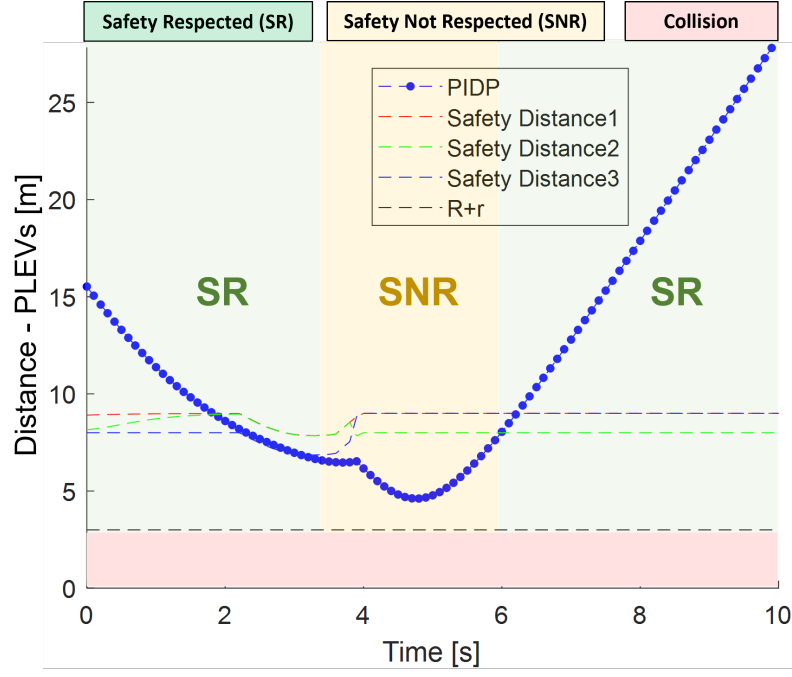


Figure 5.6: The overall inter-distance plot of ego-vehicle and PLEV for scenario 1: shows the region where Safety is Not Respected (SNR) when the PLEV changes its trajectory from Traj 3 to 2

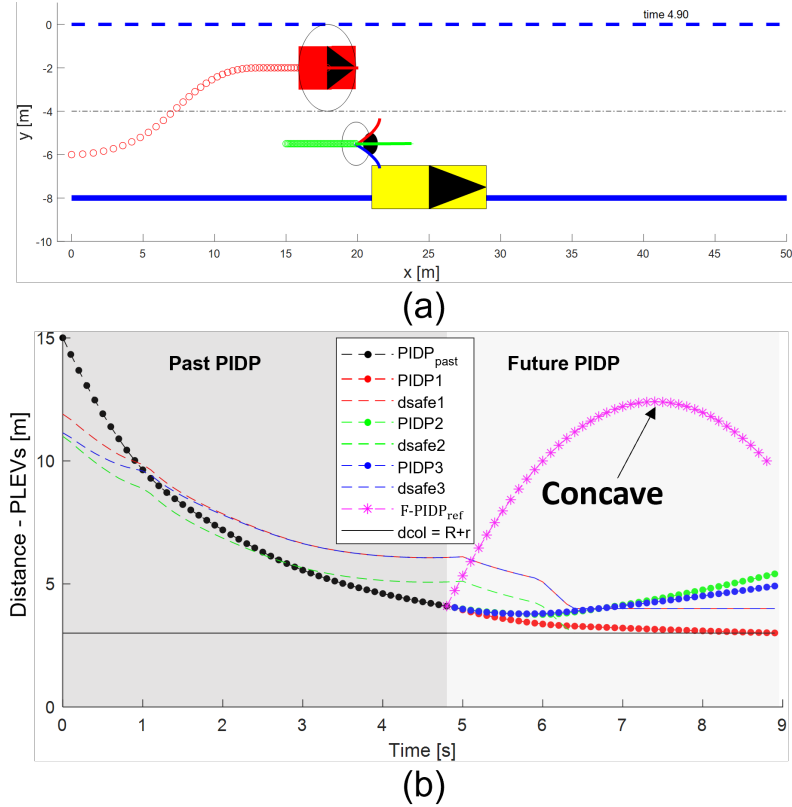


Figure 5.7: (a) Scenario 2: PLEV overtaking a parked vehicle at $t = 4.9s$ (b) Inter-distance plot of ego-vehicle and PLEV overtaking a stationary truck: the F-PIDP plot shows a concave curve that encourages escaping from the risky scenario

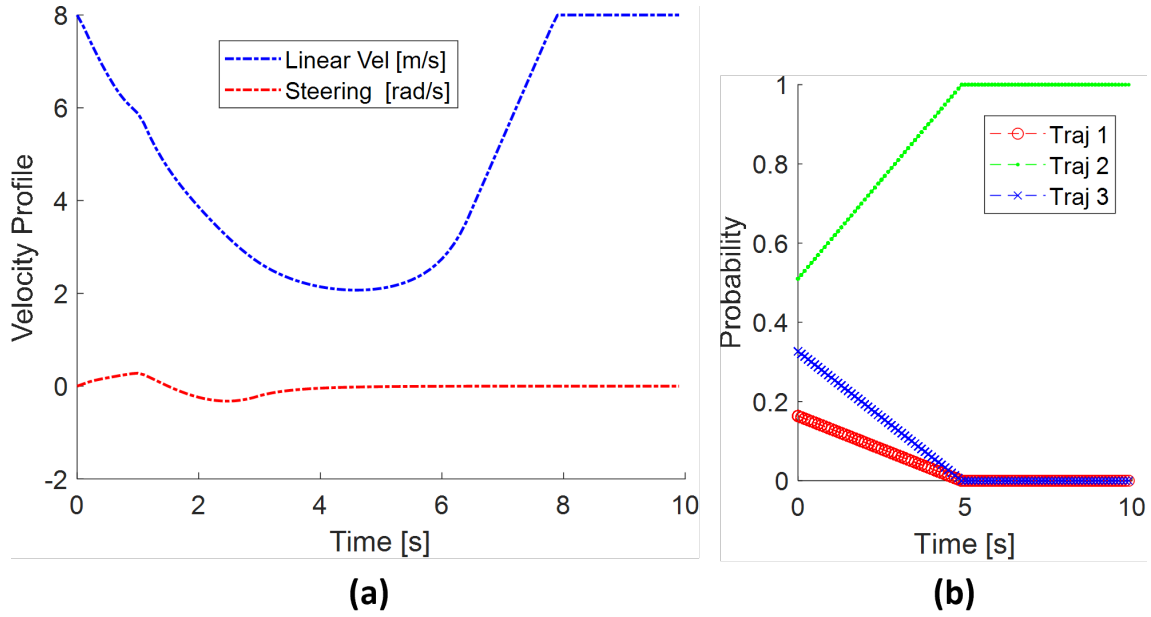


Figure 5.8: Scenario 2: (a) Velocity plot showing a smooth linear velocity and steering (b) Probability plot with increasing probability for Traj 2 (green)

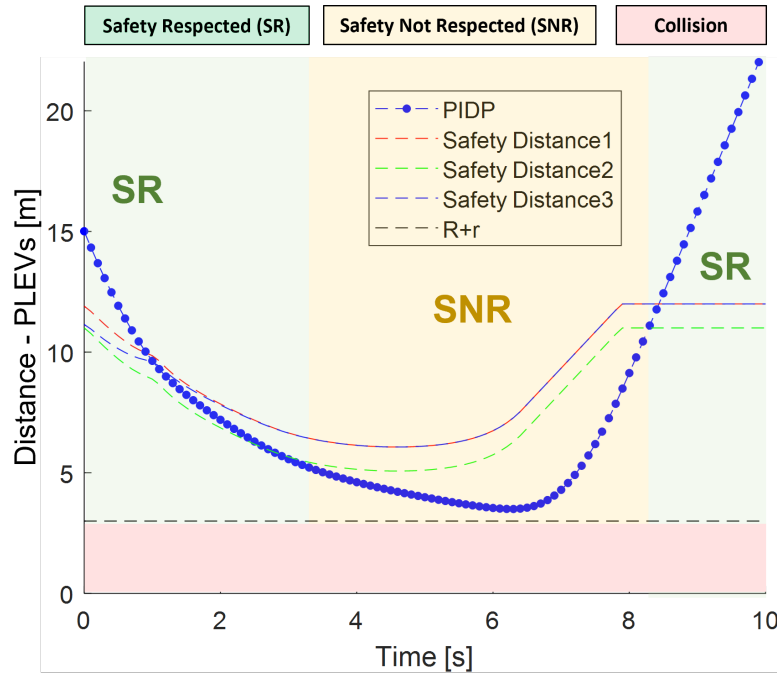


Figure 5.9: Inter-distance plot showing the overall PIDP: collision is avoided and the vehicle is able to move into the Safety Respected region (SR)

5.3.6 Scenario 2: PLEV overtaking a parked vehicle

In this scenario, the PLEV attempts to overtake a stationary truck, and in the process, comes dangerously close to the ego vehicle in the adjacent lane. An initial assessment of the scene using PIDP suggests a high risk of collision as indicated by $PIDP_1$ falling below the collision margin (cf. Figure 5.7 (b)). However, the $F-PIDP_{ref}$ setpoint does not only consider the most dangerous trajectory but also

considers the most probable trajectory of the PLEV based on the likelihood of each trajectory shown in the probability plot (cf. Figure 5.8 (b)). Therefore, the inter-distance plot indicates that the safety distance is respected in the initial few seconds as the ego-vehicle approaches the PLEV (cf. Figure 5.9).

Whereas, during the time interval from $t = 2.6s$ and $t = 8.1s$, the safety distance is temporarily not respected due to the parallel motion of the ego-vehicle and the PLEV (cf. Figure 5.9). To mitigate this risk, the adaptive behavior of the $F\text{-PIDP}_{ref}$ prompts the ego vehicle to further reduce its speed and gradually approach the PLEV. Again, the $F\text{-PIDP}_{ref}$ has a concave profile that encourages the AV to move away from the dangerous situation, as shown in Figure 5.7. This ensures that the AV neither comes to a complete halt nor makes abrupt maneuvers, but instead, decelerates and then carefully moves past the risky zone. This adaptive response is visualized in the velocity profile plot (cf. Figure 5.8). Observe that the velocity of the ego-vehicle decreases to about $2m/s$ at time $t = 4.5s$ before it accelerates to move past the PLEV. More simulation with statistical results can be found in Section 6.2.4, where more uncertainties are introduced.

5.4 THE CURSE OF DIMENSIONALITY IN PREDICTIVE CONTROL

As presented in the previous section (cf. Section 5.3) and in subsequent sections with more simulation results (cf. Section 6.2.9), predictive control algorithms, such as SMPC, can effectively incorporate the F-PIDP and other constraints for risk assessment and management in autonomous driving scenarios involving a PLEV. However, the presence of multiple PLEVs in the environment introduces additional complexity to the navigation problem. Each PLEV generates its own set of predicted trajectories, leading to an exponential increase in the number of PIDPs that need to be considered. This phenomenon is commonly referred to as the "curse of dimensionality," [167], [243] where the safe state space becomes increasingly sparse and difficult to navigate as the number of dimensions (in this case, PLEVs) increases. The curse of dimensionality poses significant challenges for predictive control algorithms, as they must account for the interactions between multiple PLEVs and their associated uncertainties. This section discusses the challenges posed by the curse of dimensionality in the context of predictive control for autonomous vehicles and presents the proposed mitigation strategy to address these challenges.

For example, consider a scenario where an autonomous vehicle is navigating in an environment with N PLEVs. Applying MPC to this scenario requires solving the following optimization problem at each time step:

$$\min_U \sum_{t=0}^{N-1} \left(\|\Delta \mathbf{x}^{AV}(t)\|_Q^2 + \|\mathbf{u}^{AV}(t)\|_R^2 \right) + \|\Delta \mathbf{x}^{AV}(N)\|_S^2 \quad (5.20)$$

such that:

$$\mathbf{x}^{AV}(t+1) = f(\mathbf{x}^{AV}(t), \mathbf{u}^{AV}(t)), \quad t \in \mathbb{N}^N \quad (5.21)$$

$$\mathbf{x}^{PV}(t+1) = f(\mathbf{x}^{PV}(t), \mathbf{u}^{PV}(t)), \quad t \in \mathbb{N}^N \quad (5.22)$$

$$\mathbf{u}^{AV}(t) \in \mathcal{U}(t), \quad t \in \mathbb{N}^N, \quad (5.23)$$

$$\mathbf{x}_{p,j}^{PV}(t) \in \Xi_{safe}, p \in \mathbb{N}^{N_p} \times j \in \mathbb{N}^{N_{traj}} \times t \in \mathbb{N}^N \quad (5.24)$$

The control input to the AV is $U = (u_0, \dots, u_{N-1})^T$ and $\|x\|_M^2 = x^T M x$. The cost to reach the reference state $\Delta \mathbf{x}^{AV}(t) = \mathbf{x}^{AV}(t) - \mathbf{x}_{ref}^{AV}(t)$, with weighting matrices $Q, S \in \mathbb{R}^{3 \times 3}$ and $R \in \mathbb{R}^{2 \times 2}$. The prediction model of the AV and PLEV are respectively $f(\mathbf{x}^{AV}(t), \mathbf{u}^{AV}(t))$ and $f(\mathbf{x}^{PV}(t), \mathbf{u}^{PV}(t))$. The constraints on the inputs and collision-safe states are $\mathcal{U}(t)$ and Ξ_{safe} , respectively. Observe that the size of the state collision avoidance constraint is a product of the number of agents N_p , the number of predicted trajectories N_{traj} , and the size of the prediction steps N . This leads to an exponential growth in the number of constraints that need to be considered as the number of PLEVs increases. This causes the controller to behave conservatively, as the optimization problem becomes too complex to solve efficiently. As a result, the algorithm may either fail to find any feasible solution that ensures safety or settle for a sub-optimal trajectory that compromises performance or increases collision risk.

The curse of dimensionality presents several challenges for predictive control algorithms, including:

- **Computational Complexity:** As the number of PLEVs increases, the computational burden on the predictive control algorithm grows exponentially. This can lead to longer computation times, making real-time control infeasible.
- **Increased Uncertainty:** The presence of multiple PLEVs introduces additional sources of uncertainty, as each PLEV may exhibit different motion patterns and behaviors. This can complicate the prediction process and make it more challenging to accurately assess and manage risk.
- **Scalability Issues:** As the number of PLEVs increases, the scalability of the predictive control algorithm becomes a concern. Algorithms that work well in low-dimensional spaces may struggle to maintain performance in higher dimensions involving multiple constraints.

5.4.1 Priority-based Strategy for Risk Management

As earlier mentioned, the need to satisfy multiple safety constraints simultaneously further contributes to the conservatism due to the curse of dimensionality in classical approaches such as MPC, particularly when road space is limited or multiple agents occupy adjacent lanes. To address this challenge, we introduce a priority-based risk management strategy [4] that dynamically identifies and focuses on the most dangerous agent at each planning step. This approach allows the AV to focus its computational resources and control actions on avoiding the

agent that poses the greatest risk, thereby improving efficiency and reducing unnecessary conservatism.

5.4.2 Selection of Most Dangerous PLEV

The priority-based strategy operates under the assumption that a lane change maneuver is feasible, and systematically evaluates all surrounding agents to determine which poses the greatest threat. The selection process considers both the minimum inter-distance and the time when safety not respected t_{SNR} (cf. Figure 5.10 & Section 4.5) for each agent. By prioritizing the agent with either the closest proximity or the earliest predicted safety violation, the AV can proactively plan evasive maneuvers and maintain a safe trajectory.

Algorithm 1 outlines the steps for single priority target selection. This method enhances the AV's ability to navigate complex, multi-risk environments while maintaining safety and operational efficiency (cf. Figure 5.10). This Algorithm can be characterized with the following 4 steps.

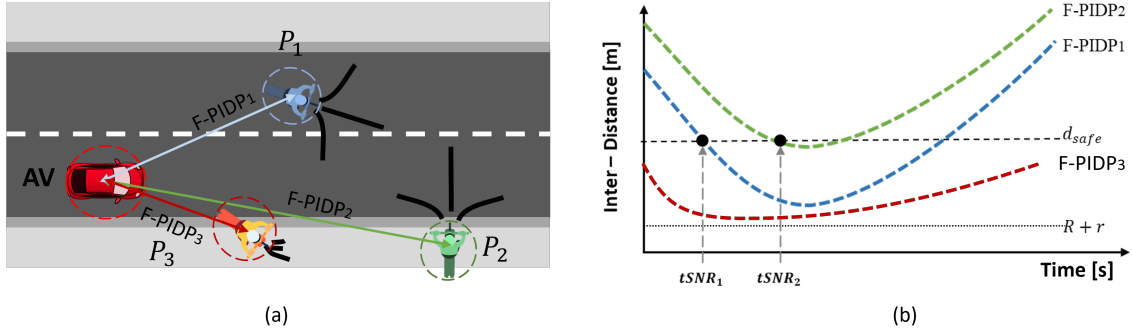


Figure 5.10: Priority-based target selection (a) Example of scenario with multiple PLEVs (b) F-PIDP plot of AV and multiple PLEVs: Priority-Based Target (P_T):= PLEV₃ ($P_{dmin} < d_{safe}$) else (P_T):= PLEV₁ the minimum ($t_{SNR1} < t_{SNR2}$)

- **Step 1:** Compute the F-PIDP of all the agents (as given in Section 5.2) and their t_{SNR} (if any, cf. Figure 5.10),
- **Step 2:** Find P_{dmin} : the PLEV with the current minimum inter-distance to the AV, and P_{tSNR} : the PLEV with the minimum t_{SNR} ,
- **Step 3:** If $P_{dmin} < d_{safe}$, then set the priority target PLEV P_T , to avoid collision with P_{dmin} (cf. PLEV₃ in Figure 5.10), else set the priority target PLEV P_T as P_{tSNR} (cf. PLEV₁ in Figure 5.10). This implies that P_{dmin} will be chosen as priority target since it is closest to the AV even if P_{tSNR} has the smallest t_{SNR} . This also allows the prioritization of agents that suddenly appear in the field of view of the AV, such as occluded PLEVs. The priority target P_T is considered to be the most dangerous agent.
- **Step 4:** Compute collision-free trajectory between target PLEV P_T and AV (using e.g. SMPC), goto **Step 1**.

Algorithm 1 Priority-based Target Selection Algorithm using F-PIDP [4]

Require: $\mathbf{x}^{AV}(t, \dots, N), \mathbf{x}^{PV}(t, \dots, N)$

```

1: while Goal not reached do
2:   Compute F-PIDP of all the PLEVs
3:   Compute  $P_{dmin}$ : the PLEV with current minimum inter-distance to the AV
4:   Compute  $P_{tSNR}$ : the PLEV with the minimum  $t_{SNR}$ 
5:   if  $P_{dmin} < d_{safe}$  then
6:      $P_T := P_{dmin}$ 
7:   else
8:      $P_T := P_{tSNR}$ 
9:   end if
10:   $\mathbf{U}^{AV} \Leftarrow \text{F-PIDP+MPC}(P_T)$ 
11: end while

```

5.5 SIMULATION RESULTS FOR SCENARIO WITH A PRIORITIZED AGENT

The priority-based risk management strategy is evaluated through multiple simulations. The setup features an AV navigating alongside multiple PLEVs within a dual-lane traffic scenario. A series of challenging and dynamic situations were evaluated to test the system's robustness (cf. Figure 5.11). These scenarios introduce two primary challenges:

- the uncertainty in the actual trajectory that each agent will follow, and
- the potential for sudden behavioral changes, particularly in terms of speed, by agents such as PLEVs.

Using agents' parameters in (cf. Table 5.1), PLEV₁ in Yellow & PLEV₂ in Blue. PLEV₁, for instance, may accelerate significantly from an initial speed of 2 m/s up to 10 m/s, adding to the unpredictability of the environment. To simplify collision-checking and minimize computational overhead, each agent is modeled using a single enclosing circle. While this approximation introduces some conservatism, it provides a practical balance between accuracy and efficiency. However, a less conservative approach could involve modeling agents with multiple circumcircles, as proposed in [16], to better capture their shape. (cf. **video shown through this link** <https://youtu.be/oH6yPuocRJK?list=PL0zhSFDHifx2ruWjKJ1ggBCKfnzertLLu>)

In this simulation setup, PLEV₁ accelerates its speed after approximately 1 second. A range of five speed levels is considered: $v \in \{2, 4, 6, 8, 10\} \text{ m/s}$. The first level represents a constant-speed case, where PLEV₁ maintains its initial velocity of 2 m/s throughout the trajectory. The remaining speed levels correspond to cases of sudden acceleration, introducing variability in agent behavior. The total simulation duration is set to 7.5 seconds, with a prediction time step of $\Delta t = 0.05$ seconds. For $N = 40$ prediction steps, a prediction horizon of $t_{horizon} = \Delta t \times N = 2 \text{ s}$ is used at each step to predict future agent positions.

Table 5.1: Agents Parameters and initial states

AV Parameters	Value	Unit
radius	2	m
initial position	[0, -6]	m
initial velocity	[8, 0]	m/s
$\mathbf{u}_{max}^{AV}$	[8, 35]	[m/s, °]
$\mathbf{u}_{min}^{AV}$	[0, -35]	[m/s, °]

PLEV1 Parameters	Value	Unit
radius	0.5	m
initial position	[10, -9]	m
initial velocity	[2, 0]	m/s
v_{max}	10	m/s

PLEV2 Parameters	Value	Unit
radius	0.5	m
initial position	[16, -6]	m
initial velocity	[2, 0]	m/s
v_{max}	2	m/s

5.5.1 Result using Standard MPC formulation

We first evaluate the simulation results using standard MPC. At PLEV1 initial speed levels of 2 m/s and 4 m/s, the MPC effectively manages these small variations in agent speeds without violating collision constraints, as illustrated in Figure 5.11. The PIDP remains above the safe distance threshold, d_{safe} , indicating safe navigation throughout the prediction horizon. This can be observed in the inter-distance plot between the AV and the two PLEVs at time $t \leq 1s$ (cf. Figure 5.12 (a) & (c)).

The MPC approach, however, adopts an overly conservative strategy, nearly halting the motion of the AV to avoid a potential collision with PLEV2. This caution stems from the PIDP indicating a risk of collision due to a possible left-turn trajectory by PLEV2 (cf. Figure 5.12 (d), in red). To mitigate this risk, the MPC algorithm prioritizes maintaining a safe distance behind PLEV2 rather than attempting to overtake it. Consequently, even at the end of the simulation, $t = 7.5$, the AV remains at a significant distance from the final goal position $\mathbf{x}_{goal} = [-2, 60]m$ (cf. Figure 5.11 (b)).

At higher speeds of 6 m/s and above, the MPC algorithm could not find an optimal solution that satisfies all safety constraints. Notably, as shown in Figure 5.13, the AV violates the road boundary constraint in an attempt to avoid a collision with other agents following the sudden acceleration of the PLEV1.

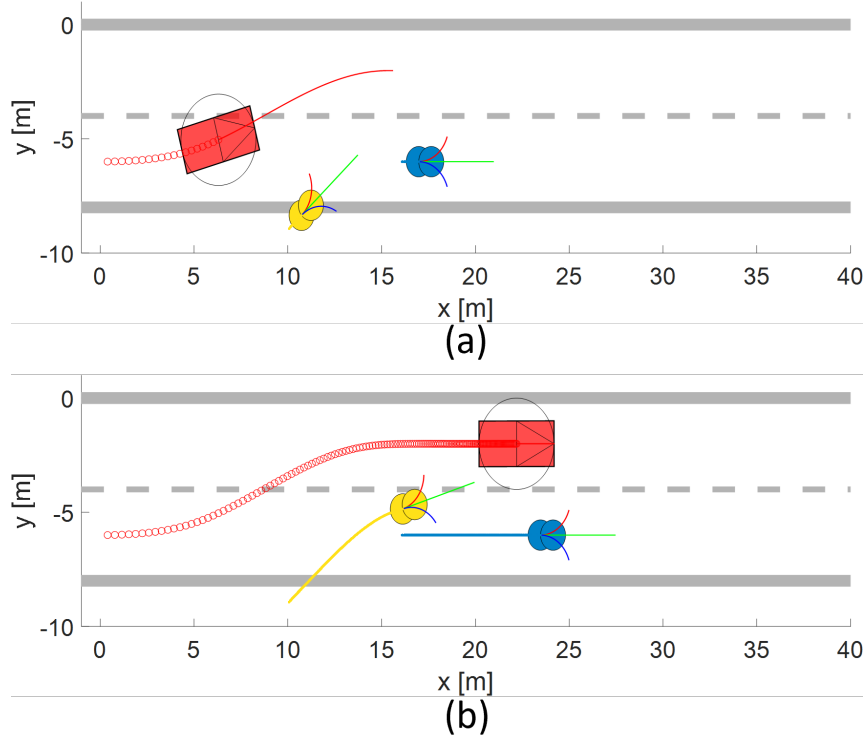


Figure 5.11: A scenario using MPC showing an AV (in red) and multi-modal PLEVs (1 in Yellow & 2 in Blue): (a) At $t = 1$ s, the actual speed of both PLEVs is 2 m/s, (b) At $t = 7.5$ s, AV overtakes PLEV1 but stays beside PLEV2 to avoid a possible collision with a possible Left turn trajectory of PLEV2

5.5.2 Result using Priority-based F-PIDP+MPC

The approach proposed, which integrates F-PIDP+MPC, outperforms standard MPC across all speed variations, provided the alternative lane remains collision-free. By utilizing F-PIDP as a risk metric and prioritizing the target PLEV identified as the most dangerous, the AV can compute a less conservative trajectory that effectively meets all safety constraints.

For example, at $t = 1$ s, PLEV1 is identified as the priority target P_T , as shown in Figure 5.14 (a). After the fusion of the PIDPs, the F-PIDP set-point indicates that the AV should increase its distance from PLEV1 to prevent a potential collision (cf. Figure 5.15 (a)). By focusing solely on the priority target, PLEV1, the F-PIDP+MPC algorithm finds a solution that enables the AV to safely overtake PLEV1. Subsequently, at $t = 5$ s, PLEV2 becomes the new priority target, which the AV overtakes after passing PLEV1 (cf. Figure 5.14 (b)).

This priority-based behavior enables the AV to quickly leave the dangerous region by focusing on the most critical agent, rather than simultaneously accounting for all multi-modal agents' motion (using standard MPC), which often results in overly conservative behavior. Consequently, the AV covers greater distances in less time compared to the standard MPC method, which remains alongside PLEV2 after 7.5 seconds (cf. Figure 5.14 (b)). Throughout the simulation, the proposed F-PIDP+MPC method ensures that the PIDPs of all agents consistently

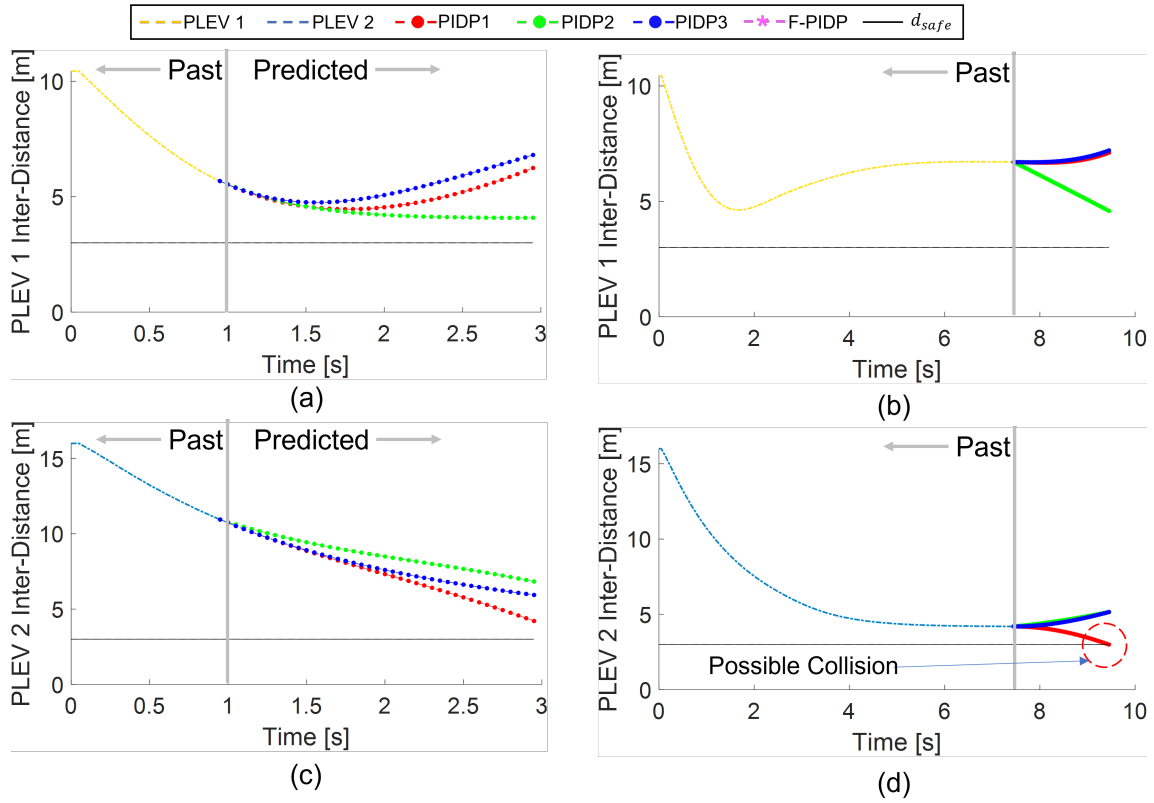


Figure 5.12: PIDP plot between the AV and the PLEVs using MPC: (a) & (c) At $t = 1$ s, all PIDPs (red, blue, green) of PLEV₁ & 2 are above safety distance d_{safe} , (b) & (d) At $t = 7.5$ s, the PIDPs of PLEV₁ are above d_{safe} while a PIDP of PLEV₂ may lead to collision, reproduced from [4]

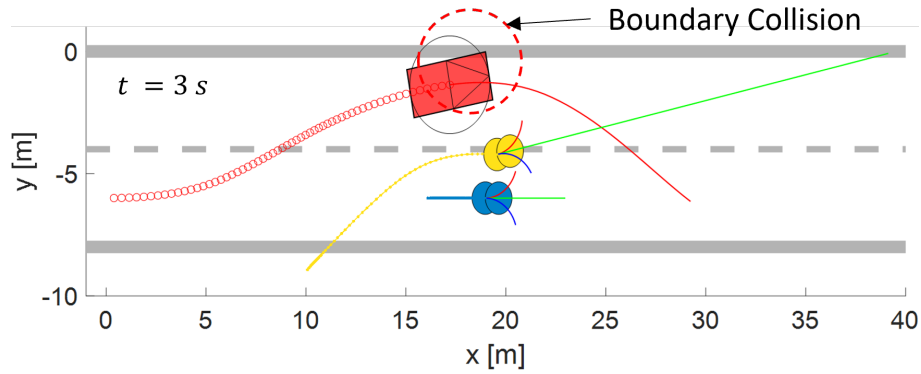


Figure 5.13: At $t = 3$ s, MPC violates the boundary constraint due to the sudden change in speed of PLEV

remain above the safety distance threshold, as illustrated in Figure 5.15. This strategy is effective for both PLEVs and other road vehicles.

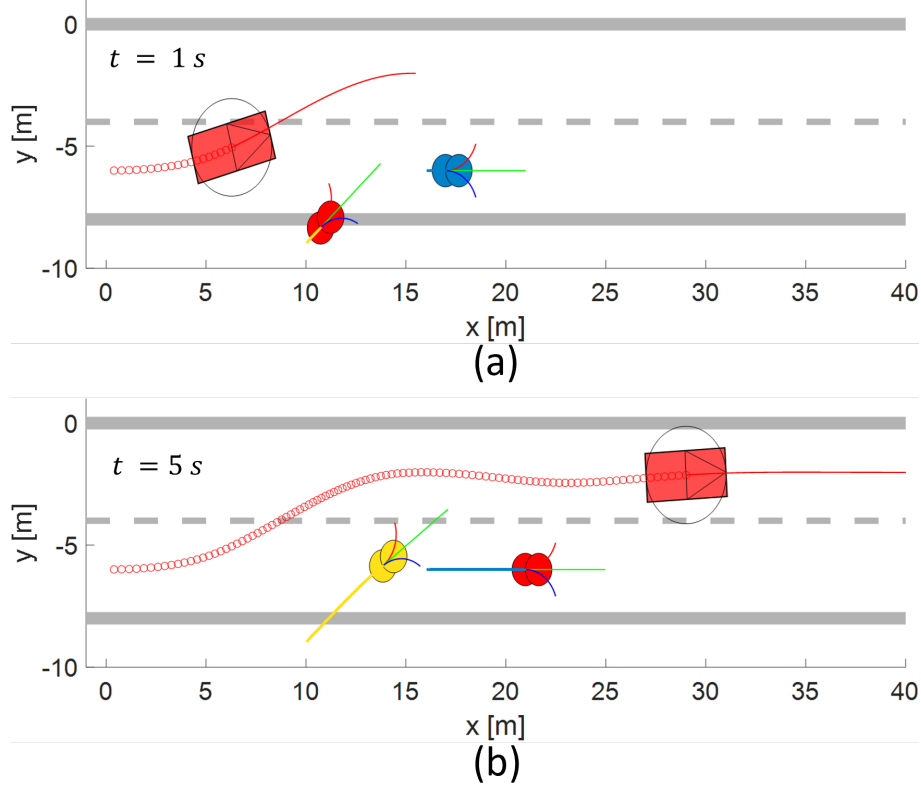


Figure 5.14: A scenario using the F-PIDP+MPC reproduced from [4]: (a) At $t = 1$ s, AV overtakes the priority target PLEV1 (in red), (b) then at $t = 5$ s, AV overtakes the new priority target PLEV2 (in red)

5.6 CONCLUSION

This chapter addresses the challenge of multi-risk assessment and safe control in the presence of multi-modal, highly uncertain micromobility agents (PLEVs). It starts by identifying the core problem: classical scalar metrics (e.g., TTC) do not capture the spatio-temporal evolution and multi-modality of inter-agent separation, and treating each predicted trajectory independently produces either brittle or overly conservative behavior.

To overcome this, the Predictive Inter-Distance Profile (PIDP) is used to represent the distance evolution over the prediction horizon. Building on PIDP, Fusion of PIDPs (F-PIDP) is introduced: characteristic (start, minimum, and end) points from each mode are fused via a probability-weighted center-of-mass and fitted with a smooth quadratic curve. F-PIDP reduces conservatism by combining likelihood and danger into a single continuous safety set and is integrated as a setpoint into a Stochastic MPC (SMPC) formulation. A priority-based selection method is also proposed to mitigate the curse of dimensionality by focusing computation on the most dangerous agent.

Overall, the chapter presents a multi-risk assessment pipeline from PIDP to F-PIDP, together with multi-risk management integrations (SMPC, priority selection) that progressively incorporate multi-modality, producing practical, computationally tractable solutions for multi-risk assessment and management

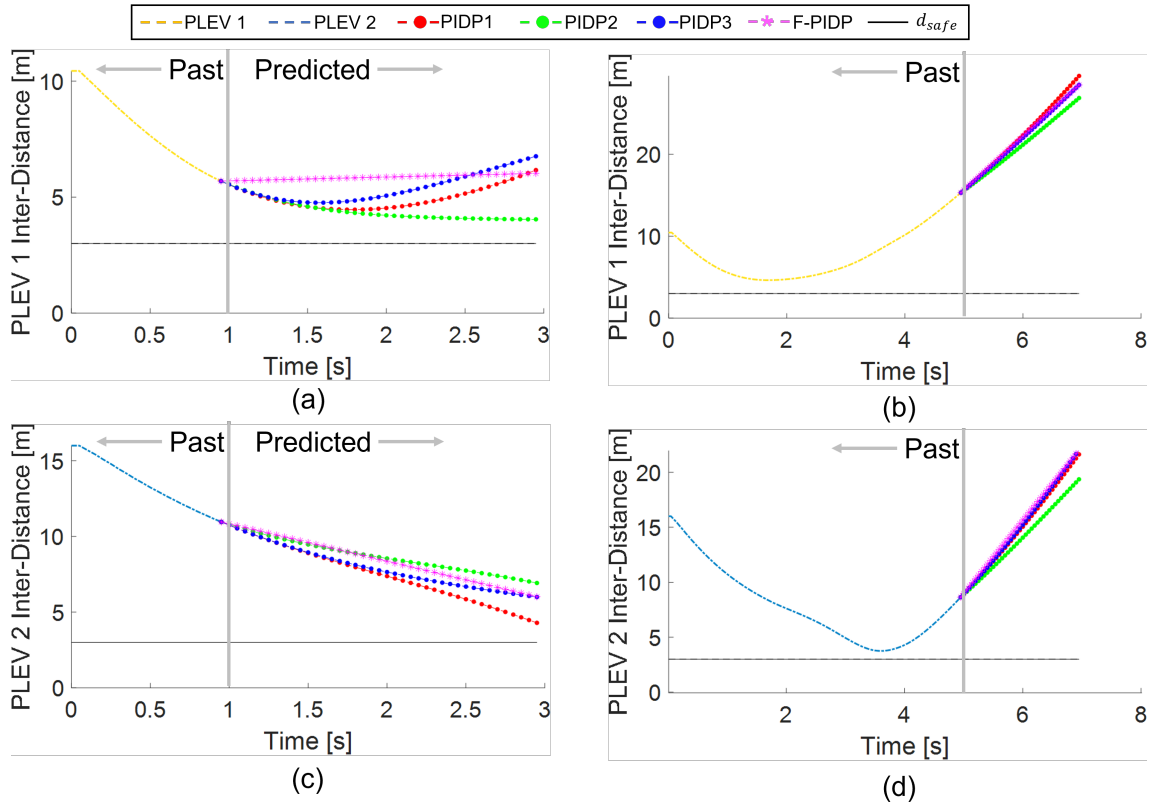


Figure 5.15: PIDP plot between the AV and the PLEVs using F-PIDP+MPC: (a) At $t = 1$ s, F-PIDP setpoints suggest creating more distance between AV and PLEV₁, (b), (c), & (d) All PIDPs are above d_{safe} as presented in [4]

around PLEVs. The next chapter will extend this approach to situations with estimation uncertainties in the state of the PLEVs.

This chapter presents a stochastic Predictive Inter-Distance Profile (sPIDP) to capture estimation uncertainties arising from sensor and model errors. It extends deterministic PIDP with an additive stochastic component derived from propagated state-estimation uncertainties. A Fusion of sPIDP (F-sPIDP) extends this to multi-modal predictions, integrated within the overall Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) architecture. MiRA-MOC combines a sampling-based global planner in the Frenet frame with a robust local controller formulated as a direct optimization problem using F-sPIDP for multi-risk Assessment and management of multi-modal agent behaviors with other sources of uncertainties. Vehicle dynamics, actuator limits, and safety margins are explicitly enforced, while a multi-priority target selection strategy further reduces conservatism in dense traffic by focusing on the most relevant agents.

6.1 INTRODUCTION

The need for a stochastic formulation of the Predictive Inter-Distance Profile (sPIDP) arises from the unavoidable uncertainty in the estimated state of surrounding agents (cf. Figure 6.1). Deterministic PIDP and its fusion, F-PIDP (cf. Section 5.2) neglect estimation noise and can therefore underestimate collision risk when sensor or prediction errors are significant. By explicitly modeling the inter-distance as a sum of a deterministic term and a stochastic perturbation, sPIDP (and its probabilistic fusion, F-sPIDP cf. Section 6.2.4) enables a principled propagation of state prediction uncertainties into safety margins and chance-constrained decision-making.

Building on this, a Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) strategy is proposed to perform multi-risk assessment and management (cf. Figure 4.1). The overall motion and control optimization problem is divided into two main tasks (cf. Figure 6.8): Safe global trajectory selection which takes into account the F-sPIDP for collision avoidance and a local control optimization. This approach efficiently assesses and manages risk while accounting for uncertainties (perceived position, velocity) in the motion of surrounding agents in the traffic environment. Figure 6.8 shows the steps of the multi-level motion optimization procedure, which are presented in detail in this chapter. Therefore, the remainder of the chapter details the sPIDP uncertainty propagation, the F-sPIDP fusion, the multi-level motion optimization strategy, and simulation results illustrating the benefits of the proposed approach.

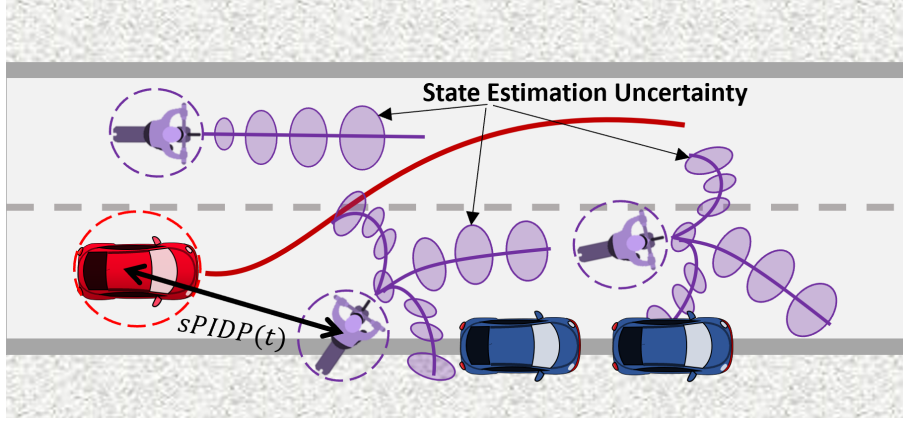


Figure 6.1: Multimodal predictions with state estimation uncertainties

6.2 STOCHASTIC PIDP (SPIDP) TO DEAL WITH MOTION ESTIMATION UNCERTAINTY

Fusion of Predictive Inter-Distance Profile (F-PIDP) presented in Section 5.2 effectively combines multiple PIDPs to provide a comprehensive multi-risk assessment for an AV navigating in an environment with multi-modal PLEVs. F-PIDP method can handle PLEVs with high dynamic multi-modal motion provided the estimated state (positions and velocities) of the agents are very accurate.

Therefore, both PIDP and F-PIDP assume that the states (intentions, positions, velocities, etc.) of the traffic participants are deterministic, which is not true in a real-world scenario since only an estimate of the current state of the traffic participants is known during motion. This implies that estimation uncertainties due to model inaccuracies, or sensor noises, need to be considered besides the multiple hypotheses on the future trajectories of a PLEV [133] (cf. Figure 6.2).

For instance, consider a scenario where the AV is navigating alongside a jaywalking PLEV. The AV estimates the current position and velocity of the PLEV using its onboard sensors, but these estimates are subject to uncertainties due to sensor noise and environmental factors. As shown in (cf. Figure 6.2 (a) & (b)), the estimated PIDP indicates that the safety distance is respected, while the actual PIDP, which accounts for the true state of the PLEV, shows that the safety distance is violated. This discrepancy highlights the importance of considering estimation uncertainties during the multi-risk assessment and management.

This multi-risk problem with estimation uncertainty is addressed in this section by computing a Fusion of stochastic PIDPs (F-sPIDP). F-sPIDP explicitly incorporates the estimated uncertainties in the predicted instantaneous states of the agents, rather than neglecting these uncertainties as done using PIDP and F-PIDP (cf. Chapter 5).

The fundamental idea of stochastic PIDP (sPIDP) is to consider the predicted inter-distance as a combination of two components (cf. Figure 6.2 (c) & (d)):

- A deterministic component equivalent to the expected/standard PIDP which uses the current best estimate of the agent's state trajectories, and

- A stochastic component that accounts for the uncertainty in the estimated states of the agents.

The stochastic PIDP is parameterized as:

$$sPIDP(t) = PIDP(t) + w(t) \quad (6.1)$$

Where the expected $PIDP(t) := \mathbb{E}[sPIDP] \in \mathbb{R}^+$ is the estimated inter-distance of the position vectors of the interacting agents (also known as the mathematical expectation or mean value), $w(t)$ denotes the uncertainties in the system, over the prediction time interval $0 \leq t \leq t_{hor}$. The sPIDP is required in Block “3” of the global architecture (cf. Figure 4.1).

F-sPIDP computes the stochastic PIDP of all the predicted maneuvers that the PLEVs can make, together with their prediction uncertainties. Then, a weighted fusion of the sPIDPs is computed using a quadratic fit. The subsequent sections present a comprehensive explanation of the procedures involved in calculating the F-sPIDP. In sub-section (6.2.2), the mean of the sPIDP is computed. Sub-section (6.2.3) evaluates the variance of the uncertainty of the sPIDP.

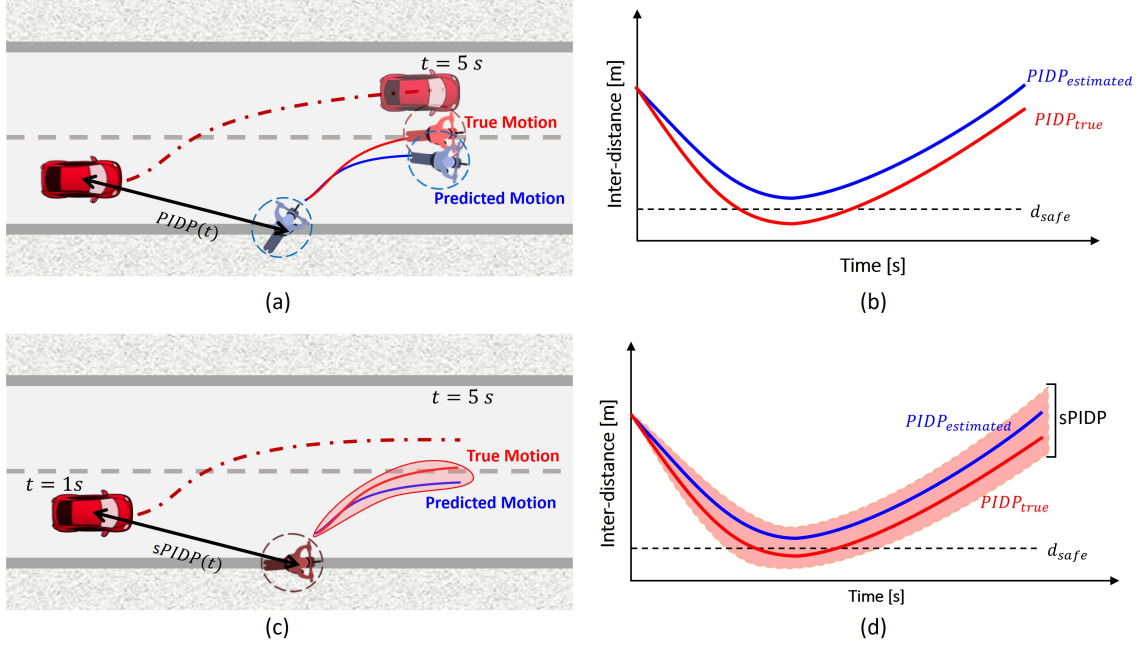


Figure 6.2: (a)&(b) Estimated PIDP shows safety distance is respected, while the Actual/True PIDP shows safety distance is not respected (c)&(d) Representing estimation uncertainties using stochastic PIDP (sPIDP)

6.2.1 Uncertainty-based Motion Prediction of PLEV

For instance, if the predicted state of the PLEV at next time step $t + 1$ is represented as a Gaussian distribution:

$$\tilde{\mathbf{x}}_j(t + 1) = A_j(t) \cdot \tilde{\mathbf{x}}_j(t) + w_j \quad (6.2)$$

$$z(t + 1) = C(t) \cdot \tilde{\mathbf{x}}_j(t) + \varepsilon \quad (6.3)$$

where $\tilde{\mathbf{x}}_j(t+1)$ denotes the predicted state of the PLEV's j th maneuver at time step t , $A_j(t)$ is the state transition matrix, the measurement $z(t+1)$ is same for all maneuvers, and $C(t)$ is the measurement matrix. Both $w_j \sim \mathcal{N}(0, \Sigma_j^w)$ and $\varepsilon \sim \mathcal{N}(0, \Sigma^\varepsilon)$ are assumed to be Gaussian additive noises with covariance matrices Σ_j^w and Σ^ε respectively. This assumption is reasonable for small deviations in the agent's instantaneous motion and can be obtained from an appropriate estimation algorithm, like in [134], which gives the predicted state of the PLEV and its covariance. The estimation of the PLEV's state is obtained from the perception Block "P" in the control architecture shown in Figure 4.1. They are inputs to our architecture, and do not correspond to the investigations in this research (cf. Section 4.3). Significant changes in PLEV motion are typically captured by a different mode within the multi-modal prediction framework.

6.2.2 Deterministic Component of sPIDP

The deterministic component of sPIDP is the value of the inter-distance when the uncertainty in (6.1) is zero. This implies that given the estimated state trajectory of the position vector $\tilde{\mathbf{p}}(t) = (\tilde{x}(t), \tilde{y}(t))$, the deterministic part of the sPIDP of two interacting agents is:

$$\mathbb{E}[sPIDP] = \mathbb{E}[\|\tilde{\mathbf{p}}_a - \tilde{\mathbf{p}}_j\|] \quad (6.4)$$

where $\tilde{\mathbf{p}}_a$ and $\tilde{\mathbf{p}}_j$ are the current estimates of the position of the AV and the i -th traffic participant. Indeed, (6.4) is the expected PIDP that corresponds to the traditional PIDP in [5] and [17] (cf. Section 4.5 & 5.2).

6.2.3 Stochastic Component of sPIDP

The uncertainty in the motion of the agents represented as $w(t) \in \mathbb{R}$ in (6.1) often has a significant impact on the behavior of the AV. Furthermore, the value of $w(t)$ is typically not provided by state-of-the-art motion prediction algorithms. Therefore, we project the predicted state covariance uncertainty of the agents to obtain the predicted inter-distance uncertainty, $w(t) : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$. Let the variance of the inter-distance uncertainty be:

$$\mathbb{V}[sPIDP] = \mathbb{E}[d^2] - (\mathbb{E}[d])^2 \quad (6.5)$$

where $\mathbb{E}[d] =: \mathbb{E}[sPIDP]$ is the expected value of sPIDP given in (6.4), therefore we present below the derivation of $\mathbb{E}[d^2]$.

$$\begin{aligned} \mathbb{E}[d^2] &= \mathbb{E}[(\tilde{x}_r - \tilde{x}_i)^2 + (\tilde{y}_r - \tilde{y}_i)^2] \\ &= \mathbb{E}[\tilde{x}_r^2] + \mathbb{E}[\tilde{x}_i^2] + \mathbb{E}[\tilde{y}_r^2] + \mathbb{E}[\tilde{y}_i^2] - 2(\mathbb{E}[\tilde{x}_r \tilde{x}_i] + \mathbb{E}[\tilde{y}_r \tilde{y}_i]) \end{aligned} \quad (6.6)$$

The formula for defining the expected values of the agents' squared position component is $\mathbb{E}[x^2] = \mathbb{V}[x] + \mathbb{E}[x]^2$. The variance of each position component $\mathbb{V}[\cdot]$ derived from the uncertainty provided by the motion prediction algorithm

[50], [81], [134] can be utilized to compute $\mathbb{E}[x^2]$. Hence, given the covariance matrix $P_j(t+1)$ as shown in [134], defined in (6.7):

$$P_j(t+1) = A_j(t) \cdot P_j(t) \cdot A_j^T(t) + \Sigma_j^w \quad (6.7)$$

where $A_j(t)$ and Σ_j^w are defined in (6.3). Putting (6.4) and (6.6) in the variance equation (6.5) will return the uncertainty of the sPIDP.

The sPIDP of a possible PLEV trajectory is depicted in Figure 6.2 (c & d), where a 2D surface represents the deterministic and stochastic components of sPIDP between an AV and the PLEV. The proposed approach does not rely on a specific assumption about the distribution of uncertainty, making it applicable to most motion prediction algorithms.

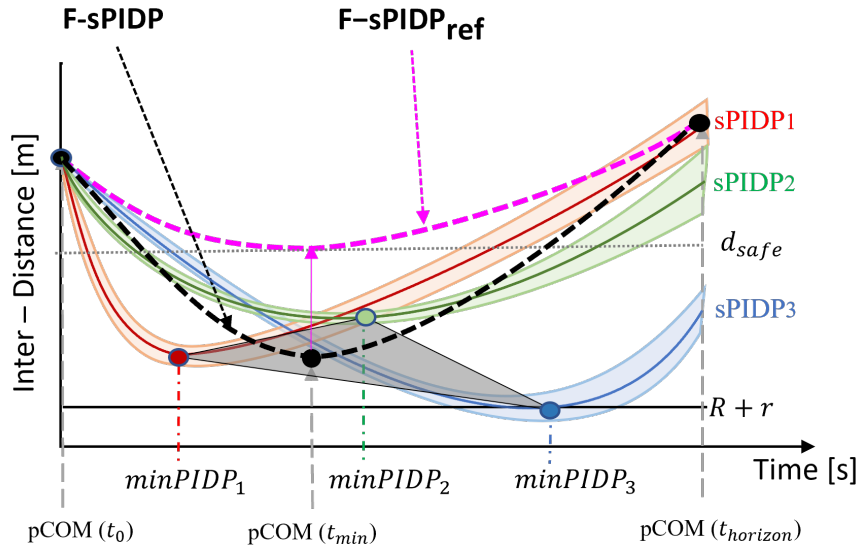


Figure 6.3: Fusion of stochastic PIDPs (F-sPIDP) of an AV and a PLEV with three (3) possible multi-modal maneuvers and their uncertainties (variance)

6.2.4 Fusion of Stochastic PIDP (F-sPIDP) for multi-modal Agents Multi-Risk Assessment

PLEVs are known to exhibit multi-modal behaviors, which increases the complexity of the navigation problem since a single agent may have multiple predictions from the same or multiple motion prediction algorithms. Similar to the F-PIDP, the Fusion of sPIDP (F-sPIDP) is proposed to fuse the sPIDPs of all the predicted maneuvers of a PLEV.

Like F-PIDP, we extract the same characteristic parameters from each sPIDP that capture the significant points on the sPIDP curve:

1. $sPIDP_j(t_0)$: the sPIDP value at time (t_0) , has the same value for all the sPIDPs of a given agent.
2. $sPIDP_j(t_{min})$: the sPIDP value at time (t_{min}) , the minimum point of each sPIDP curve.

3. $sPIDP_j(t_{hor})$: the sPIDP value at time (t_{hor}) , the end point of each sPIDP curve.

Consequently, at each time step, all sPIDP parameters are merged into a single quadratic polynomial curve, weighted by their probability $Pr(j)$. These parameters are integrated by calculating the probabilistic center of mass (pCOM) (5.6) for all sPIDPs, this becomes:

$$F_sPIDP(t) = \sum_{j=1}^{N_{traj}} Pr(j) \times sPIDP_j(t) \quad (6.8)$$

where, each maneuver $j \in \mathbb{N}^{N_{traj}}$ has a probability $Pr(j)$, and $sPIDP_j(t)$ represents the sPIDP at any of the feature points $t \in \{t_0, t_{min}, t_{hor}\}$. N_{traj} indicates the total number of maneuvers considered. The $F_sPIDP(t) = pCOM(t)$, a weighted average of the feature points based on the probability of the PLEV's maneuver. As a result, the F-sPIDP curve can be parameterized as:

$$\begin{bmatrix} F_sPIDP(t_0) \\ \vdots \\ F_sPIDP(t_{hor}) \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 \\ \vdots & \vdots & \vdots \\ 1 & t_{hor} & t_{hor}^2 \end{bmatrix} \cdot \begin{bmatrix} q_0 \\ q_1 \\ q_2 \end{bmatrix} \quad (6.9)$$

where $[q_0, q_1, q_2]$ are unknown variables that describe the F-sPIDP curve and are obtained by solving the above linear equation by taking the matrix (or pseudo) inverse at $t \in [t_0, t_{min}, t_{hor}]$. As illustrated in Figure 6.3, the outcome is a singular, cohesive F-sPIDP curve that approximates the information from all of the sPIDPs. The definition of the appropriate F-sPIDP and a coherent F_sPIDP_{ref} setpoint required by the multi-risk management algorithm for safety can be derived as shown in Section 5.2.

6.2.5 F-sPIDP in Uncertainty-aware MPC (U-MPC)

The F_sPIDP_{ref} is integrated into an Uncertainty-aware MPC (U-MPC) framework for multi-risk management which enhance the AV's ability to navigate safely in environments with multiple PLEVs exhibiting uncertain behaviors. The U-MPC formulation incorporates the F-sPIDP as a dynamic safety constraint, allowing the AV to adapt its trajectory based on the assessed multi-risk from surrounding agents. The U-MPC optimization problem is formulated as follows:

$$\min_{\mathbf{U}} \mathcal{F}(\tilde{\mathbf{x}}(t), \mathbf{x}_{ref}, \mathbf{U}, \mathbf{U}_{ref}) \quad (6.10)$$

$$\text{such that } \begin{cases} \forall t & \in \{0, \dots, N\} \\ \tilde{\mathbf{x}}(t+1) & = f(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \\ \tilde{\mathbf{x}}_j(t+1) & = f(\tilde{\mathbf{x}}_j(t), w) \\ PIDP(t) & \geq \mathcal{D}_{safe}(\delta_v, d_w(t)) \\ \mathbf{U}_{min} & \leq \mathbf{U}(t) \leq \mathbf{U}_{max} \end{cases} \quad (6.11)$$

based on the current estimation of the state of the ego-vehicle $\tilde{\mathbf{x}}(t+1)$ and the PLEV most probable trajectory $\tilde{\mathbf{x}}_j(t+1)$. The lower and upper bounds of the acceptable control inputs are denoted as $\mathbf{U}_{min}$ and $\mathbf{U}_{max}$, respectively. While $\mathbf{U} = [\mathbf{u}(t), \dots, \mathbf{u}(N-1)]$ are the desired safe control inputs for the ego-vehicle, $\mathcal{D}_{safe}(\cdot)$ denotes the safety distance constraint function. Therefore, the output of the U-MPC algorithm is the value of $\mathbf{U}$ that minimizes the objective function $\mathcal{F}$, defined in the following sub-section 6.2.6. Sub-Section 6.2.7 defines the uncertainty and speed safety constraint.

6.2.6 Definition of U-MPC Objective Function

It is required that the motion of the ego-vehicle must be as smooth as possible and also safe as it optimizes its tracking of the defined F-sPIDP set-point (cf. Section 6.2.4 and Figure 6.3). Therefore, the U-MPC optimizes a combination of multiple objective functions. For simplicity, the multi-objective functions are considered to be least-squares functions:

1. The stage cost to reach the target destination $\mathbf{x}_{ref}$: $f_x = \frac{1}{2} \|\tilde{\mathbf{x}}(t) - \mathbf{x}_{ref}\|^2$
2. The terminal cost to reach the target destination $\mathbf{x}_{ref}$ of the ego-vehicle: $f_{x_N} = \frac{1}{2} \|\tilde{\mathbf{x}}(N) - \mathbf{x}_{ref}\|^2$
3. The energy cost to reach the reference control action $\mathbf{U}_{ref}$: $f_u = \frac{1}{2} \|\mathbf{U}(t) - \mathbf{U}_{ref}\|^2$
4. The deviation cost from the reference F-sPIDP set-point: $f_p = \frac{1}{2} \|PIDP(t) - F_sPIDP(t)\|^2$

Therefore, the overall multi-objective function $\mathcal{F}$ is given by (6.12)

$$\mathcal{F} = \gamma_1 f_x + \gamma_2 f_{x_N} + \gamma_3 f_u + \gamma_4 f_p; \quad \sum_{i=1}^4 \gamma_i = 1 \quad (6.12)$$

with $\gamma_i \in \mathbb{R}^+$ representing global weights to balance the influence of the cost functions on the overall multi-objective function.

6.2.7 Uncertainty-Based Safety Constraint

The risk of collision with other traffic agents is assumed to be proportional to the linear velocity of the AV and the uncertainty in the inter-distance between the AV and PLEV. This is because the faster the motion of the AV, the more cautious it needs to be around other road users. Similarly, the more the level of uncertainty in the predicted state of the PLEV, the higher the risk of collision. Therefore, the safety distance constraint function is assumed to be dynamic and defined as

$$\mathcal{D}_{safe}(\delta_v, d_w(t)) = R + r + \delta_v + d_w(t) \quad (6.13)$$

The braking distance δ_v depends on the vehicle velocity and

$$d_w(t) := \mathcal{N}(0, \mathbb{V}ar[sPIDP]) \quad (6.5)$$

is the predicted uncertainty of the current most probable sPIDP. This implies that the safety distance constraint function $\mathcal{D}_{safe}$ in (6.11) is linear with respect to the sum of the radii of the AV (R) and PLEV (r), the AV's braking distance, and the sPIDP uncertainty.

a) Collision Avoidance as a Chance constraint

We employ sPIDP to guarantee safety between the AV and the surrounding road users. The approach is similar to the traditional PIDP (cf. Section 4.5 & 5.2), here the collision constraint is expressed as

$$\mathcal{D}_{safe}(\delta_v, d_w(t)) \leq sPIDP(t) \quad (6.14)$$

The uncertainty due to estimation errors in the AV and PLEVs' position (and radius) is addressed through the addition of the noise in the sPIDP to the safety distance and by formulating (6.14) as a chance-constraint problem with a desired probability α to be satisfied,

$$Pr(d_{safe} + d_w(t) \leq sPIDP(t)) \geq \alpha \quad (6.15)$$

with $0.5 \leq \alpha \leq 1$, $d_{safe} = R + r + \delta_v$

α is a tunable risk parameter. When α is large, the behavior of the AV becomes very conservative, which can make the chance-constraint optimization problem infeasible. However, a small value of α may lead to a high risk of collision. A value of $0.7 \leq \alpha \leq 0.95$ provides a good balance between operationality and risk. The following section presents a method to tighten the chance constraint to reduce the impact of large levels of estimation uncertainties.

b) Collision Avoidance Constraint Tightening

Directly adding uncertainty to the safety distance may lead to highly conservative behavior, since the estimated uncertainty may be very large. We address this

problem by performing constraint tightening using the desired risk parameter α . Hence, the constraint in (6.15) is separated into a deterministic and stochastic component with a limit on the inter-distance, such that d_{safe} will have a high probability of being below the PIDP. This implies that:

$$\begin{aligned} d_{safe} + \sigma(t) &\leq d(t) \\ Pr(d_w(t) \leq \sigma(t)) &\geq \alpha \end{aligned} \quad (6.16)$$

since the uncertainty in $d_w(t) \sim \mathcal{N}(0, \text{Var}[sPIDP])$ is known (cf. Section 6.2.3) and $d(t) = \mathbb{E}[\|\tilde{\mathbf{p}}_a - \tilde{\mathbf{p}}_j\|]$ is the current inter-distance as in (6.4), the value of $\sigma(t)$ can be computed explicitly during every iteration from the quantile function of a Gaussian distribution as, $\sigma(t) = \sqrt{2 * \text{Var}[sPIDP]} \times \text{erf}^{-1}(2\alpha - 1)$. Where the function $\text{erf}^{-1}(\cdot)$ denotes the inverse error function.

6.2.8 Simulation and Results using F-sPIDP + SMPC

The motion of the AV is subjected to the following constraints in addition to the safety constraint

$$y \in [0, -8]m, \delta \in [-35, 35]^\circ, v \in [0, 8]m/s, d_{safe} = 3m$$

The minimum safety distance d_{safe} between the AV and a PLEV is set to 3 m, with values below this indicating a collision. The PLEV has three possible maneuvers with their associated probabilities for slow forward motion (red), fast forward motion (green) and turning right (blue) (cf. Figure 6.6). The predicted trajectories of the PLEV and their associated probabilities are assumed to be provided by an appropriate perception/prediction algorithm (cf. Section 4.3). However, for simplicity, the evolution of the probability of the trajectories is assumed to be the same for the tested scenarios depending on the PLEV trajectory. (cf. **video shown through this link** <https://youtu.be/mzyxZo0jUB8?list=PL0zhSFDHifx2ruWjKJ1ggbCKfnzertlLu>)

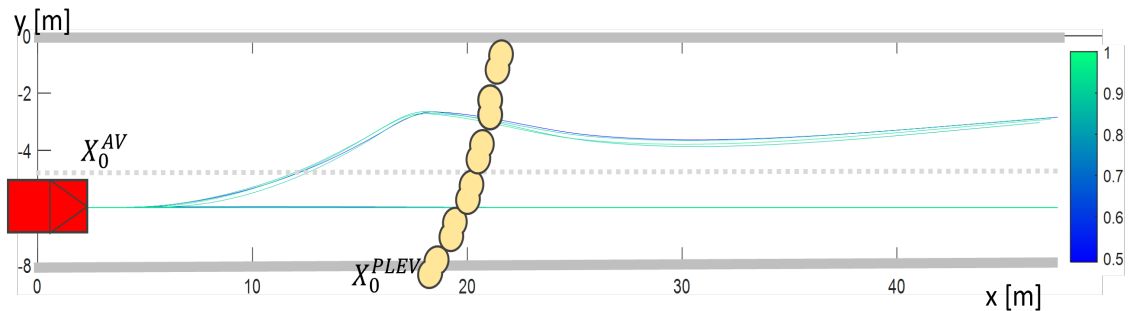


Figure 6.4: Test Scenario of the AV and a single PLEV crossing the road: showing the different trajectories of the AV for Varying Risk Factor $\alpha \in [0.5, 1]$

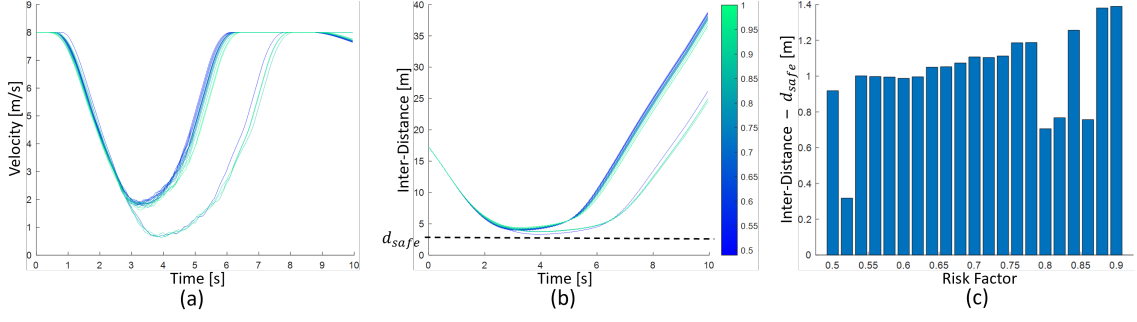


Figure 6.5: Varying Risk Factor $\alpha \in [0.5, 1]$: (a) Velocity, (b) Inter-Distance, and (c) Safety Distance

6.2.9 Chance Constraint Risk Factor Test

The first scenario involves an AV and a PLEV crossing the road at constant velocity. The results show that even lower values of the risk factor $\alpha \in [0.5, 1]$ yield safe trajectories for the AV, either by decelerating or performing a lane change (cf. Figure 6.4). The velocity plot shows a deceleration to low velocity of about 1 m/s for high values of the risk factor to create more distance, while for lower values of the risk factor the AV decelerates to an average speed of 2 m/s (cf. Figure 6.5 (a) & (b)). The difference between the safety distance and the actual inter-distance for varying parameters of the risk factor is shown in Figure 6.5 (c), with large values when the risk factor is high which indicates a conservative behavior.

We set $\alpha = 0.7$, and simulated multiple agents with two PLEVs and two parked vehicles. The F-sPIDP of each PLEV is computed independently. The result shows that safety is respected: all stochastic PIDPs and the ground truth $PIDP_{true}$ of the agents are always above d_{safe} without constraint violations (cf. Figure 6.6).

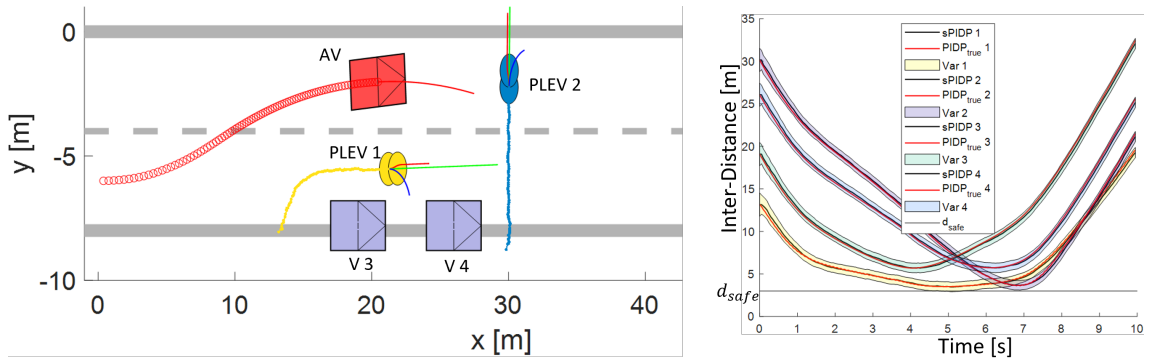


Figure 6.6: Scenario with multiple agents: Left, shows the 2D scenario of the AV trying to avoid collision with the surrounding vehicles. Right, shows the plot of the estimated sPIDP for each agent and their corresponding variance ($V[sPIDP]$), all stochastic PIDP above d_{safe}

6.2.10 Statistical Comparison with SMPC

In this section, we compare the proposed method using (F-sPIDP + U-MPC) with classical SMPC. The scenario involves a PLEV that suddenly enters the traffic environment at an average speed of 4 m/s while trying to overtake two parked vehicles. This scenario is very challenging because the PLEV starts from a position only a few meters from the AV. Furthermore, different levels of multi-object detection and tracking errors were considered, starting from low translation errors for LiDAR-based tracking algorithms to high levels of uncertainty in camera-based tracking algorithms ($\Sigma_T \in [0.02, 0.1, 0.5, 1]m$) [138].

The following performance metrics are considered for comparison of the results (cf. Table 6.1):

1. Minimum inter-distance minPIDP (cf. Figure 6.3): this measures the smallest safety distance or risk of collision between the AV and all the other agents. A high minPIDP indicates a safer motion.
2. Cost to reach the reference position (Cost to Ref.): measures the algorithm's efficiency to reach the desired goal quickly. It measures the distance between the final position of the AV and the reference position ($\|x_N - x_{ref}\|$, where $x_{ref} = 100m$) A small value of Cost to Ref. implies an efficient motion towards the reference position (AV's destination).
3. Computation Time: measures the time it takes for the ego-vehicle to compute the control actions required by the AV.

Report of the mean (over 20 runs) of the minimum inter-distance (minPIDP), cost to reach the reference position (Cost to Ref.), and Computation Time are shown in Table 6.1.

To ensure clarity, just the outcomes of the interaction between the AV and PLEV are studied. Due to the high rate of constraint violation using SMPC with only the most probable trajectory, we considered SMPC with all 3 trajectories as collision constraints. This approach reduces the collision rate, but have a significant impact on the efficiency of the method. Therefore, when the uncertainty is low ($\Sigma_T \leq 0.1m$), the resulting motion is smooth using all the methods considered. SMPC gives high values of minPIDP due to its conservative behavior since it tries to avoid all three predicted trajectories of the agents. Whereas, the proposed approach using only the most probable trajectory and the $F\text{-}sPIDP_{ref}$ maintains a safety distance above $d_{safe} = 3m$, with the lowest Cost to Ref. and computation time, which can be further optimized for real-time application. A low Cost to Ref. of 43.80 m using the proposed approach, compared to 67.54 m using SMPC, shows that the final position of the AV using the proposed approach is about 24 m ahead of the SMPC-based method (cf. Figure 6.7).

For high levels of simulated position errors ($\Sigma_T \geq 0.5m$) often observed in camera-based object detection algorithms [138], SMPC becomes too conservative while trying to increase the safety distance (min PIDP) between the AV and the PLEVs. This behavior in such a narrow space often results in poor performance that violates the constraints and keeps the AV far from the reference position as

indicated by the high value of the Cost to Ref. (cf. Table 6.1). Moreover, when the maximum object tracking error is 1 m, SMPC couldn't find a safe solution, thereby violating the constraint by moving too close to the road boundary (cf. Figure 6.7). A limitation of the proposed method, common to most MPC-based approaches, is its high computational time. This can be improved by strategies such as increasing the sampling interval Δt , reducing the number of prediction steps N , or leveraging sampling-based optimization techniques that are capable of utilizing GPU acceleration for faster computation.

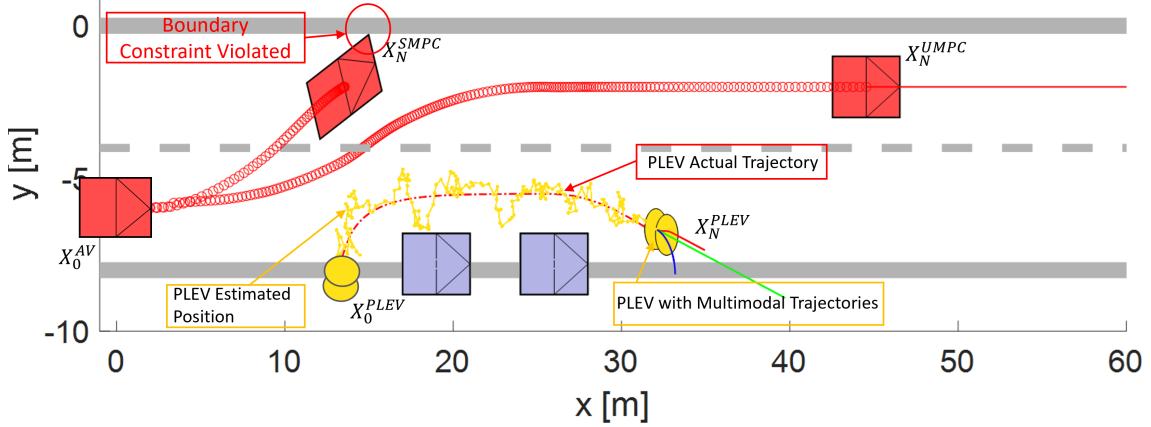


Figure 6.7: Final Position of AV at Final Time N for both the proposed U-MPC and SMPC methods with tracking error of 1m: SMPC violates the constraint by moving too close to the boundary, U-MPC maintains a safe distance without being too conservative

Table 6.1: Comparison with varying Object Tracking Errors. Report of the mean (over 20 runs) of (min PIDP $\uparrow$, Cost to Ref. $\downarrow$, and Computation Time $\downarrow$)

Method	Metric / Σ_T	0.02 m	0.1 m	0.5 m	1 m
SMPC	minPIDP [m]	4.64	4.67	4.15	5.50
	Cost to Ref. [m]	67.54	67.83	68.72	86.47
	Computation Time [s]	0.17	0.17	0.20	0.26
U-MPC	min PIDP [m]	3.28	3.32	3.46	3.45
	Cost to Ref. [m]	43.80	43.90	51.64	55.07
	Computation Time [s]	0.11	0.13	0.16	0.16

6.3 MULTI-LEVEL MOTION OPTIMIZATION AND CONTROL

The main idea is that MPC and its variants, such as SMPC, struggle to handle multiple moving agents, particularly in roads with nonlinear curvatures, as discussed in Section 4. Although MPC allows adding multiple constraints, the optimization problem becomes more complex with each additional constraint (road or agent), leading to longer computation times and potential infeasibility. This is

particularly problematic in urban environments where real-time decision-making is crucial for safety. To overcome these limitations, a Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) strategy is proposed to perform sampling-based global trajectory optimization in the Frenet coordinate frame to handle road boundaries and agent constraints efficiently. Meanwhile, the local control optimization tracks the reference trajectory, accounting for vehicle kinematics and the assessed level of uncertainty. This approach enables more efficient multi-risk assessment and management while accounting for unforeseen disturbances in the motion of surrounding agents in the traffic environment.

1. Sampling-based Optimal Trajectory Selection 2. Local Trajectory Optimization

Compute trajectory samples

Select Safe and Optimal Global Path using Fusion of stochastic PIDP (F-sPIDP)

Compute Local trajectory respecting vehicle kinematic constraints and motion uncertainties obtained from F-sPIDP

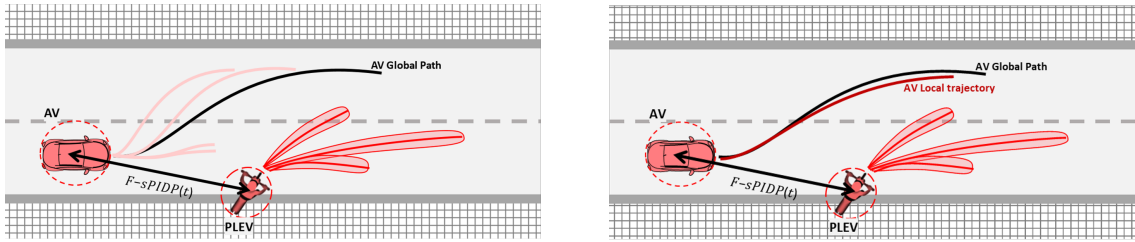


Figure 6.8: An overview of the multi-level motion optimization steps: (Left) Sampling-based Optimal Trajectory Selection, (Right) Local Trajectory Optimization

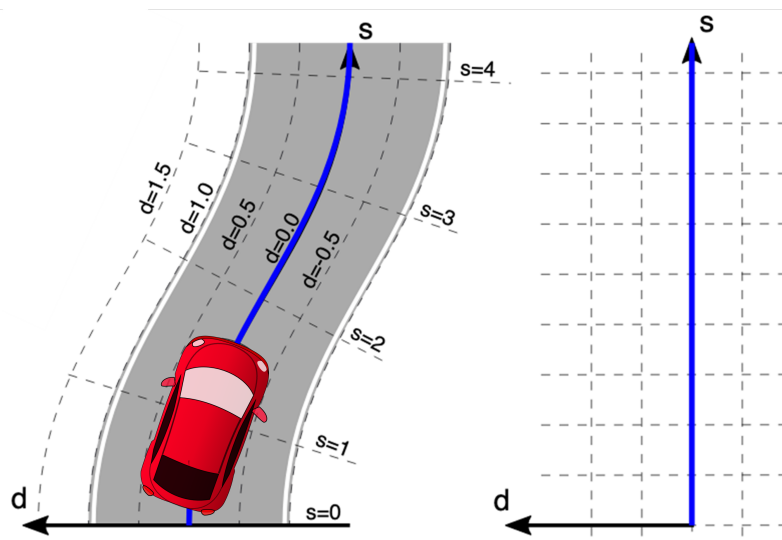


Figure 6.9: Global path in Frenet coordinates

6.4 SAMPLING-BASED SAFE OPTIMAL TRAJECTORY SELECTION

Generating traffic-adapted trajectories is challenging in highly dynamic and uncertain environments, due to the coupled effects of the vehicle's lateral and longitudinal dynamics in a Cartesian frame. Furthermore, during collision avoidance, the search for an optimal safe trajectory is a non-convex problem, which is very challenging for most optimization solvers. To address this challenge, a sampling-based trajectory optimization approach is proposed in the Frenet coordinate frame to generate lane-disciplined trajectories that better describe the relationship between vehicle dynamics and road geometry. Representing the road environment in the Frenet frame enables transforming the curvilinear road into a linear representation (cf. Figure 6.9) instead of the straight line assumption in Section 5.3.2. The algorithm efficiently explores the solution space based on the road's shape and identifies optimal, safe trajectories that minimize a defined cost function while satisfying safety constraints. The main steps of the sampling-based trajectory optimization approach are presented in the following sections.

6.4.1 Lateral Trajectory Generation

The lateral trajectory generation is based on the quintic polynomial model, widely used in autonomous driving applications for its ability to ensure smoothness and continuity in position, velocity, and acceleration profiles. Its main focus is on obstacle avoidance and lane change. The quintic polynomial is defined as follows:

$$d(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (6.17)$$

where $d(t)$ represents the lateral position as a function of the longitudinal distance t , and $a_0, a_1, a_2, a_3, a_4, a_5$ are the coefficients of the polynomial. These coefficients are determined based on boundary conditions, which typically include the initial and final lateral positions, velocities, and accelerations. The boundary conditions can be expressed as:

$$\begin{aligned} d(t_0) &= d_0, & d(t_f) &= d_f \\ \dot{d}(t_0) &= v_{d0}, & \dot{d}(t_f) &= v_{df} \\ \ddot{d}(t_0) &= a_{d0}, & \ddot{d}(t_f) &= a_{df} \end{aligned}$$

where d_0 and d_f are the initial and final lateral positions, v_{d0} and v_{df} are the initial and final lateral velocities, and a_{d0} and a_{df} are the initial and final lateral accelerations. By solving this system of equations, the coefficients $a_0, a_1, a_2, a_3, a_4, a_5$ can be determined, enabling the generation of a smooth lateral trajectory that satisfies the specified boundary conditions. This approach ensures that the vehicle can transition smoothly between lanes or navigate around obstacles while maintaining comfort and safety.

Therefore, given the initial state of the ego vehicle and a set of desired final lateral positions, velocities, and accelerations, a set of candidate lateral trajectories can be generated by varying the final conditions. This allows for the exploration

of different maneuver options, such as lane changes or obstacle avoidance, while ensuring that each trajectory adheres to road limits and shape constraints. For a given start point at time t_0 and end point at time t_f , the matrices of the initial and final boundary conditions can be expressed as follows:

$$\mathbf{Y}_0 = \begin{bmatrix} d_0 \\ v_{d0} \\ a_{d0} \end{bmatrix} \quad \text{and} \quad \mathbf{Y}_f = \begin{bmatrix} d_f \\ v_{df} \\ a_{df} \end{bmatrix} \quad (6.18)$$

The coefficients of the quintic polynomial can then be computed by solving the following linear system:

$$\begin{bmatrix} 1 & d_f & d_f^2 & d_f^3 & d_f^4 & d_f^5 \\ 0 & 1 & 2d_f & 3d_f^2 & 4d_f^3 & 5d_f^4 \\ 0 & 0 & 2 & 6d_f & 12d_f^2 & 20d_f^3 \\ 1 & d_0 & d_0^2 & d_0^3 & d_0^4 & d_0^5 \\ 0 & 1 & 2d_0 & 3d_0^2 & 4d_0^3 & 5d_0^4 \\ 0 & 0 & 2 & 6d_0 & 12d_0^2 & 20d_0^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} y_f \\ v_{df} \\ a_{df} \\ d_0 \\ v_{d0} \\ a_{d0} \end{bmatrix} \quad (6.19)$$

By solving this system, the coefficients $a_0, a_1, a_2, a_3, a_4, a_5$ can be determined, allowing for the generation of a smooth lateral trajectory that meets the specified boundary conditions. This approach ensures that the vehicle can transition smoothly between lanes or navigate around obstacles while maintaining comfort and safety.

6.4.2 Longitudinal Trajectory Generation

The longitudinal trajectory generation is based on the quartic polynomial model which ignores the final position constraint. This model is particularly useful to ensure a constant motion along the reference lane. The quartic polynomial is defined as follows:

$$s(t) = b_0 + b_1t + b_2t^2 + b_3t^3 + b_4t^4 \quad (6.20)$$

where $s(t)$ represents the longitudinal position as a function of time t , and $(b_0, b_1, b_2, b_3, b_4)$ are the coefficients of the polynomial. These coefficients are determined based on boundary conditions, which typically include the initial and final longitudinal positions, velocities, and accelerations. The boundary conditions can be expressed as:

$$\begin{aligned} s(t_0) &= s_0, & \dot{s}(t_0) &= v_{s0}, & \ddot{s}(t_0) &= a_{s0} \\ \dot{s}(t_f) &= v_{sf}, & \ddot{s}(t_f) &= a_{sf} \end{aligned}$$

where s_0 is the initial longitudinal position, v_{s0} and v_{sf} are the initial and final longitudinal velocities, and a_{s0} and a_{sf} are the initial and final longitudinal accelerations. By solving this system of equations, the coefficients $(b_0, b_1, b_2, b_3, b_4)$

can be determined, allowing for the generation of a smooth longitudinal trajectory that meets the specified boundary conditions. This approach ensures that the vehicle can maintain a consistent speed along the reference lane while adapting to changes in traffic conditions.

Therefore, given the initial state of the ego vehicle and a set of desired final longitudinal velocities and accelerations, a set of candidate longitudinal trajectories can be generated by varying the final conditions. This allows for the exploration of different speed profiles, such as acceleration or deceleration, while ensuring that each trajectory adheres to speed limits and traffic conditions. For a given start point at time t_0 and end point at time t_f , the matrices of the initial and final boundary conditions can be expressed as follows:

$$\mathbf{X}_0 = \begin{bmatrix} s_0 \\ v_{s0} \\ a_{s0} \end{bmatrix} \quad \text{and} \quad \mathbf{X}_f = \begin{bmatrix} v_{sf} \\ a_{sf} \end{bmatrix} \quad (6.21)$$

The coefficients of the quartic polynomial can then be computed by solving the following linear system:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} s_0 \\ v_{s0} \\ a_{s0} \\ v_{sf} \\ a_{sf} \end{bmatrix} \quad (6.22)$$

By solving this system, the coefficients $(b_0, b_1, b_2, b_3, b_4)$ can be determined, allowing for the generation of a smooth longitudinal trajectory that meets the specified boundary conditions. This approach ensures that the vehicle can maintain a consistent speed along the reference lane while adapting to changes in traffic conditions.

6.5 F-SPIDP FOR OPTIMAL SAFE TRAJECTORY SELECTION

Combining the lateral and longitudinal trajectory generation methods described in Sections 6.4.1 and 6.4.2, a set of candidate trajectories can be generated by varying the final conditions for both lateral and longitudinal movements. Each candidate trajectory is evaluated based on a cost function that considers factors such as comfort, efficiency, and F-sPIDP for safety. To transform the candidate trajectories from the Frenet frame back to the Cartesian frame, the following equations are used:

$$x(t) = x_{ref}(s(t)) - d(t) \sin(\theta_{ref}(s(t))) \quad (6.23)$$

$$y(t) = y_{ref}(s(t)) + d(t) \cos(\theta_{ref}(s(t))) \quad (6.24)$$

where $x_{ref}(s)$ and $y_{ref}(s)$ are the reference path coordinates, and $\theta_{ref}(s)$ is the heading angle of the reference path at position s . The derivatives of the trajectory

in the Cartesian frame can be computed using the chain rule, taking into account the derivatives of $s(t)$ and $d(t)$. This transformation allows for the evaluation of the candidate trajectories in the context of the actual road geometry and surrounding environment, facilitating the selection of the optimal safe trajectory for the autonomous vehicle in the following sections.

6.5.1 Sampling Objective Cost Function

Given a starting position, the optimization problem aims to find a safe trajectory $\mathbf{x}_{ref}[t, t_{hor}]$ from a set of trajectory samples $\mathcal{X}_{traj}$ that leads the AV to the target destination without colliding with other traffic agents. The trajectory samples are computed in the road-aligned Frenet frame, this allows the treatment of curved roads as straight trajectories and implicitly accounts for motion primitives like lane change and lane keeping. Table 6.2 summarizes the cost functions used to evaluate the generated trajectory samples. The cost functions are categorized into three main aspects: comfort, efficiency, and safety. Each cost function is designed to quantify specific performance metrics that are crucial for safe and efficient autonomous driving. The overall cost function in (cf. Section 4.6) for a given trajectory sample is computed as a weighted sum of the individual cost functions:

$$\mathcal{J}(\mathbf{x}_{traj}) = w_c J_{comfort} + w_e J_{efficiency} + w_s J_{safety} \quad (6.25)$$

The optimal safe trajectory $\mathbf{x}_{ref}$ is then selected by minimizing the overall cost function across all trajectory samples that satisfy the safety constraints, as follows:

$$\mathbf{x}_{ref}[t_0, t_f] = \arg \min_{\mathbf{x}_{traj} \in \mathcal{X}_{safe}} \mathcal{J}(\mathbf{x}_{traj}) \quad (6.26)$$

subjected to:

$$g(\ddot{x}(t), \ddot{y}(t), x(t), y(t)) \leq \mathbf{0} \quad \forall t \in [t, t_f], \quad (6.27)$$

$$\mathbf{x}(t) \in \mathcal{X}_{safe} \quad \forall t \in [t, t_f], \quad (6.28)$$

where w_c , w_e , and w_s are the weights assigned to each aspect, reflecting their relative importance in the overall evaluation. The trajectory sample with the lowest total cost $\mathcal{J}(\mathbf{x}_{traj})$ is selected as the optimal safe trajectory for the autonomous vehicle to follow. This approach ensures that the selected trajectory balances comfort, efficiency, and safety considerations, leading to a more robust and reliable autonomous driving experience. The various constraints on the kinematics of the system are represented by the nonlinear function in (6.27) as proposed in [226]. The following subsection presents the proposed method to satisfy the safety constraint in (6.28) to avoid the surrounding agents.

6.5.2 Uncertainty-aware Collision Check

The candidate trajectories are transformed from the Frenet frame to the Cartesian coordinates to prevent collisions with other traffic agents. Instead of considering

Table 6.2: Global Objective Cost Functions

	Cost Function	Description
Comfort	$J_{accel} = \int_{t_0}^{t_f} a^2 dt$	Penalizes large accelerations, by evaluating total squared acceleration a
	$J_{J_{lat}} = \int_{t_0}^{t_f} \dot{a}_{lat}^2 dt$	Penalizes sudden changes in lateral acceleration, by evaluating total squared lateral jerk $\dot{a}_{lat}$
	$J_{J_{lon}} = \int_{t_0}^{t_f} \dot{a}_{lon}^2 dt$	Penalizes sudden changes in longitudinal acceleration, by evaluating total squared longitudinal jerk $\dot{a}_{lon}$
Efficiency	$J_{time} = \int_{t_0}^{t_f} 1 dt$	Calculates the time taken to complete the maneuver
	$J_{goal} = \int_{t_0}^{t_f} s dt$	Penalizes the longitudinal distance covered to encourage shorter trajectories
	$J_{vel} = \int_{t_s}^{t_f} (v(t) - v_{ref}(t))^2 dt$	Penalizes deviations from a reference velocity $v_{ref}(t)$ over a given period from t_s to t_f
Safety	$J_{lane} = \int_{t_0}^{t_f} (d(t) - d_{ref}(t))^2 dt$	Penalizing deviations from the lane center $d_{ref}(t)$ for lane disciplined driving and safety
	$J_{F\text{-}SPIDP} = \sum_{N_{agent}} \int_{t_0}^{t_f} (F\text{-}SPIDP_{ref} - d(t)) dt$	Evaluates the total collision risk for N_{agent} obstacles over time to be maximized

multiple trajectories of each agent, we compute the F-sPIDP of all PLEVs. The F-sPIDP of each candidate trajectory is computed as shown in Section 6.2.4, effectively reducing the multiple PIDPs to a single abstraction of the Multi-Risk Assessment (the F-sPIDP). The set of trajectories that satisfy constraint (6.28) for all i -th PLEVs are defined as:

$$\mathcal{X}_{safe} := \left\{ \mathbf{x}_{traj} : F\text{-}SPIDP_i \geq d_{safe} + \mathbb{V}[sPIDP] \right\} \quad (6.29)$$

where $\mathcal{X}_{safe}$ is the safe set. Similar to the traditional PIDP, safety is guaranteed when $\min(F\text{-}SPIDP_i) \geq d_{safe}$ for all the agents. We define $d_{safe} = R + r + \delta_v$, where the radius of the AV and PLEV are R and r respectively, δ_v is a braking distance. This formulation of the collision avoidance constraint allows for easy adaptation for different shapes of the AV and PLEVs (e.g., rectangular, circular, elliptical) by adjusting the safety distance d_{safe} . The optimal trajectory $\mathbf{x}_{ref}$ and its $F\text{-}SPIDP_{ref}$ is the global trajectory that minimizes the objective function (6.26) and satisfies the constraints (6.27) and (6.29) (cf. Figure 6.10). Algorithm 2 gives a summary of the optimal trajectory selection process.

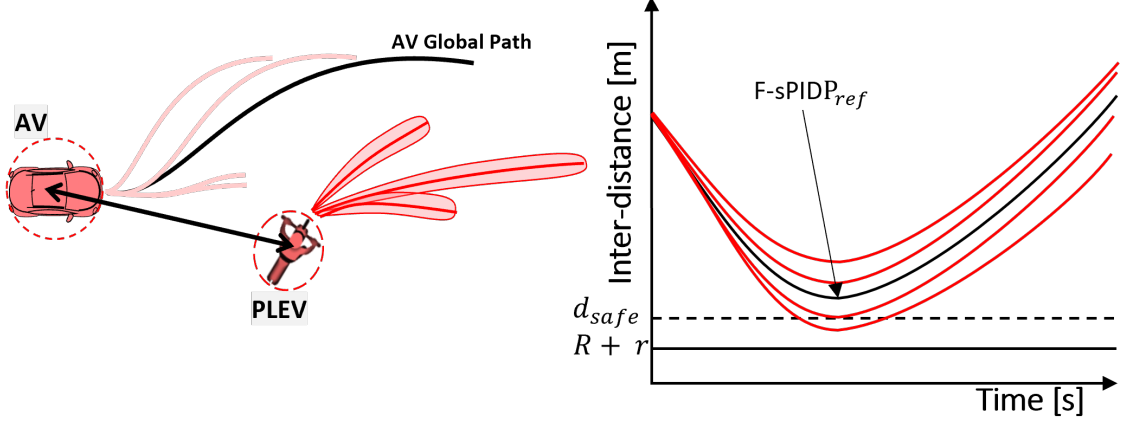


Figure 6.10: The candidate trajectories and the corresponding F-sPIDP between an AV and the possible trajectories of an agent: each F-sPIDP curve represents the fusion of inter-distances between AV's candidate trajectory and all the maneuvers of the PLEV

Algorithm 2 Safe Optimal Trajectory Selection

Given set of AV trajectory samples $\mathcal{X}_{traj}$ (cf. Figure 6.10)
Initialize empty set of safe trajectory $\mathcal{X}_{safe}$
for each trajectory $\mathbf{x}_{traj} \in \mathcal{X}_{traj}$ **do**
 Compute F-sPIDP between $\mathbf{x}_{traj}$ and all trajectories of the i -th Agent
 if $\mathbf{x}_{traj}$ satisfy Equation (6.27) & (6.29) **then**
 Add $\mathbf{x}_{traj} \rightarrow \mathcal{X}_{safe}$
 Compute objective cost $\mathcal{J}(\mathbf{x}_{traj})$ in Equation (6.26)
 end if
end for
Return $\mathbf{x}_{ref} \leftarrow \arg \min \mathcal{J}(\mathbf{x}_{traj})$, where $\mathbf{x}_{ref} \in \mathcal{X}_{safe}$

6.6 LOCAL CONTROL OPTIMIZATION

Once the global optimal safe trajectory $\mathbf{x}_{ref}$ is selected, a local control optimization problem is formulated to track the reference trajectory while integrating the vehicle's dynamic and kinematic constraints, along with the assessed level of uncertainty in the motion of surrounding agents (e.g., PLEVs). It is formulated as a robust optimization problem to ensure that the autonomous vehicle can adapt its control inputs in real-time, maintaining safety and performance even in the presence of disturbances and uncertainties. The local control optimization problem requires frequent updates and adjustments to the control inputs based on the latest state estimate of the vehicle and its environment. The main components of the local control optimization problem are presented in the following subsections.

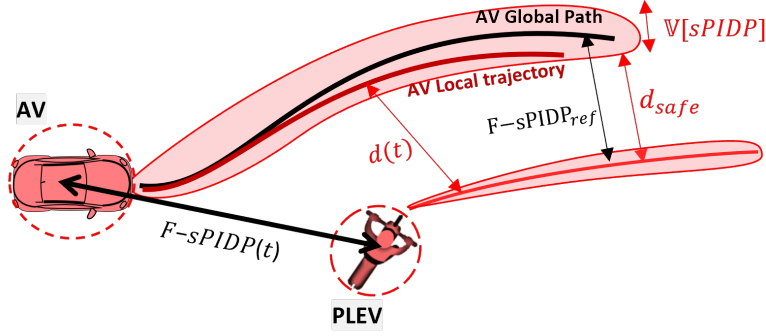


Figure 6.11: The local trajectory of the AV while respecting the uncertainty constraint $\mathbb{V}[sPIDP(t)]$

6.6.1 Control Objective and Cost Function

The objective of the local controller is to minimize deviations from the global reference trajectory (cf. Section 6.5.2) while maintaining smooth control actions and ensuring safety. The control problem is expressed as the following optimization:

$$\min_{\mathbf{u}[t, t_{hor}]} \sum_{t=0}^{t_{hor}} \left(\|\mathbf{x}(t) - \mathbf{x}_{ref}(t)\|_Q^2 + \|\mathbf{u}(t)\|_R^2 + \delta^2 \right), \quad (6.30)$$

where $\|\mathbf{x}\|_A := \mathbf{x}^T A \mathbf{x}$ denotes a weighted quadratic norm with $A = A^T$. Here, $\mathbf{x}(t)$ and $\mathbf{u}(t)$ represent the state and control input vectors of the AV, respectively. The first term in the cost function penalizes deviations from the reference trajectory, ensuring accurate tracking of the system's state. The second term penalizes excessive control inputs, promoting smooth and efficient driving behavior. The third term penalizes deviations from the safe distance defined by F-sPIDP, ensuring that the vehicle maintains a safe distance from surrounding PLEVs. This term is critical in dynamic and uncertain environments, as it helps to mitigate the multi-risks associated with unpredictable agent behaviors. By minimizing this cost function, the local control optimization problem seeks to achieve a balance between performance and safety, enabling the autonomous vehicle to navigate effectively in complex traffic scenarios.

6.6.2 Control Constraints

The optimization problem in MPC typically prioritizes tracking a reference trajectory while respecting constraints. However, it often struggles to react promptly to obstacles due to the non-convex formulation of the obstacle avoidance constraints, particularly when the AV requires significant deviations from the planned path. By computing a safe reference path in the previous section, the optimization problem is only subjected to the vehicle dynamics, the safety threshold, and the actuator constraints:

$$f(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{x}(t+1), \quad \forall t \in [t, t_{hor}], \quad (6.31)$$

$$\|F_sPIDP_{ref} - d(t)\| \leq \delta, \quad \forall t \in [t, t_{hor}], \quad (6.32)$$

$$\mathbb{V}[sPIDP] \geq \delta, \quad \forall t \in [t, t_{hor}], \quad (6.33)$$

$$\mathbf{u}(t) \in \mathcal{U}, \quad \forall t \in [t, t_{hor}]. \quad (6.34)$$

The first constraint (6.31) enforces the system dynamics, ensuring state evolution follows the nonlinear kinematic or dynamic model $f(\cdot)$. The second and third constraints (6.32)–(6.33) ensure that the inter-distance error δ between the AV and the most probable predicted trajectory of the surrounding agent remains within the uncertainty bounds $\mathbb{V}[sPIDP]$. The inter-distance is defined as:

$$d(t) = \|\tilde{\mathbf{p}}_{AV}(t) - \mathbf{p}_i(t)\|_2, \quad (6.35)$$

where $\mathbf{p}_i(t) = (\tilde{x}_i(t), \tilde{y}_i(t))$ denotes the predicted position of the i^{th} agent at time t . Finally, constraint (6.34) restricts the control input $\mathbf{u}(t)$ (e.g., steering, throttle, braking) within the admissible bounds $\mathcal{U}$ imposed by physical or comfort limits.

6.6.3 Overall Control Optimization Problem

The complete optimization problem integrates tracking, minimization of control effort, and uncertainty-aware safety constraints. It can be summarized as:

$$\min_{\mathbf{u}[t, t_{hor}]} J(\mathbf{x}, \mathbf{u}, \delta) = \sum_{t=0}^{t_{hor}} \left(\|\mathbf{x}(t) - \mathbf{x}_{ref}(t)\|_Q^2 + \|\mathbf{u}(t)\|_R^2 + \delta^2 \right) \quad (6.36)$$

$$\text{s.t. } f(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{x}(t+1), \quad (6.37)$$

$$\|F_sPIDP_{ref} - d(t)\| \leq \delta \leq \mathbb{V}[sPIDP], \quad (6.38)$$

$$\mathbf{u}(t) \in \mathcal{U}. \quad (6.39)$$

This formulation enables the AV to track the locally planned trajectory $\mathbf{x}_{ref}$ and its corresponding probabilistic reference F_sPIDP_{ref} (as obtained in Section 6.5.2) while maintaining robustness to prediction uncertainty. Given the quadratic structure of the objective and convex constraints, the problem can be efficiently solved using real-time optimization solvers such as CasADi optimization framework [10], allowing for implementation within model predictive control (MPC) frameworks.

6.6.4 Multi-priority-based Collision Avoidance Strategy

In this work, we also address scenarios where there are many potential conflicts with other vehicles on the road. This often leads to conservative multi-risk management. Here, the decision-making and control problem regarding the appropriate maneuver that the AV can execute is divided into two parts:

1. Decision on possible lane maneuver
2. Evaluation of the vehicle control inputs

The decision on a potential lane change maneuver depends on whether the adjacent lane is safe for the AV to move into. Hence, instead of a single priority agent, a set of multiple priority agents is considered based on the level of danger posed to the AV. To be less conservative, this approach can be used to consider a priority agent on each lane of the road. Therefore, Algorithm 1 given in Section 5.4.1 is modified to identify multiple priority agents in the environments as shown in Algorithm 3.

Algorithm 3 Multi-priority-based Target Selection Algorithm using F-sPIDP

Require: $\mathbf{x}^{AV}(t, \dots, N), \mathbf{x}^{PV}(t, \dots, N)$

```

1:  $P_{multi} \leftarrow \emptyset$ 
2: while Goal not reached do
3:   Compute F-sPIDP of all the PLEVs
4:   for Each Road Lane do
5:     Compute  $P_{dmin}$ : the PLEV with the minimum inter-distance to the AV
6:     Compute  $P_{tSNR}$ : the PLEV with the minimum  $t_{SNR}$ 
7:     if  $P_{dmin} < d_{safe}$  then
8:        $P_{multi} \leftarrow P_{dmin}$  : Add to multi-priority agents list
9:     else
10:       $P_{multi} \leftarrow P_{tSNR}$  : Add to multi-priority agents list
11:    end if
12:  end for
13: end while

```

6.7 RESULTS AND DISCUSSION

6.7.1 Comparison between F-PIDP and F-sPIDP using Multi-level Motion Optimization

F-sPIDP was evaluated against the earlier proposed F-PIDP [5] (cf. Section 5.2) in a dangerous scenario, where a PLEV was suddenly detected in front of the AV. Figure 6.12 (a) shows the current state of the agents, the predicted motion of the AV, and the predicted multi-modal motion that the PLEV can execute in this scenario. The plot of the inter-distance between their future motion shows a possibility of violating the safety constraint since both the F-sPIDP and F-PIDP curves are similar and below the safety distance (d_{safe}) (cf. Figure 6.12 (b)). This implies that both methods can account for the multi-modal uncertainties/probability in the motion of the detected agent (PLEV). F-PIDP assumes a deterministic state estimation, which is a liability in the presence of highly dynamic agents. As shown in Figure 6.12 ((c) & (d)), the reference trajectory using F-PIDP ($F\text{-PIDP}_{ref}$) is invariant to the degree of noise Σ_T in the estimated inter-distance, as compared to the reference F-sPIDP trajectories ($F\text{-sPIDP}_{ref}$) that varies with respect to the magnitude of the estimated state uncertainty. As a result, F-PIDP risks generating paths that are less conservative but close to the safety distance, potentially

resulting in dangerous situations in real-world applications. In contrast, F-sPIDP incorporates the stochastic uncertainty ($V[sPIDP]$), resulting in safer trajectories.

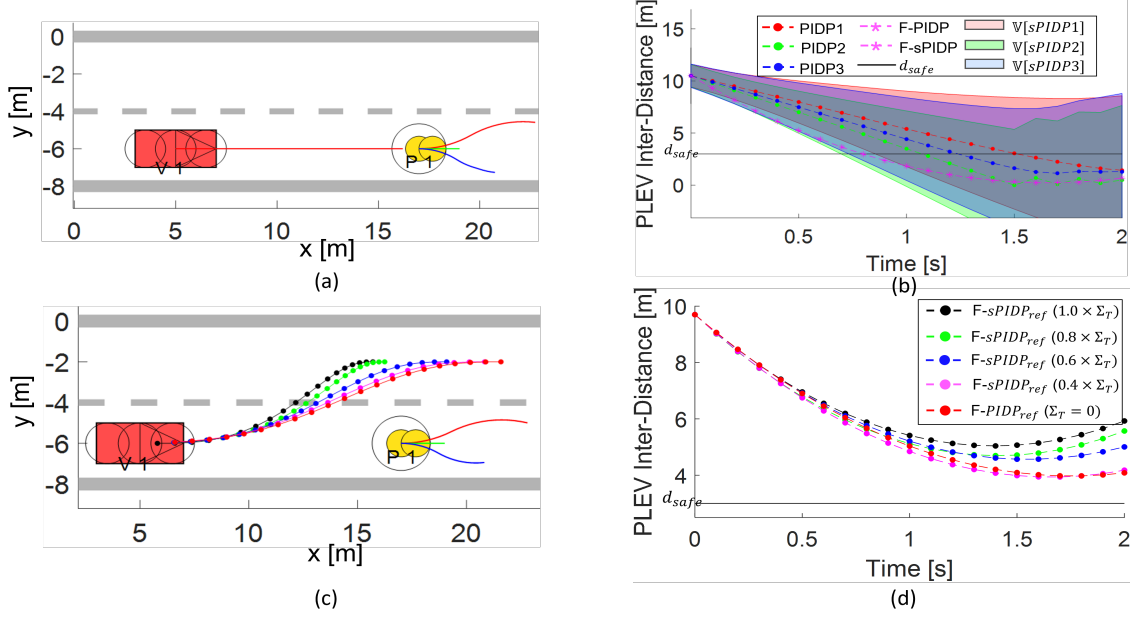


Figure 6.12: Comparison between F-PIDP and F-sPIDP: (a) the current state of the agents, the predicted motion of the AV, and the predicted multi-modal motion of the PLEV, (b) the PIDPs, F-PIDP, F-sPIDP, and estimated inter-distance uncertainty plot, showing possible collision with PLEV (all PIDPs $< d_{safe}$), (c) & (d) sampled trajectories for varying levels of uncertainty, where $F-PIDP_{ref}$ is invariant to the level of uncertainty

6.7.2 Scenario with Multiple Priority PLEVs

In this simulation setup, we incorporate an additional vehicle, V1, moving in the adjacent lane, which influences the feasibility of a lane change maneuver based on V1's motion (cf. Figure 6.13). This introduces the possibility of multiple priority agents depending on the lane, contrary to the situation in [4] (cf. Section 5.4.1) that uses F-PIDP+MPC. To address this, our proposed multi-priority-based trajectory sampling method first evaluates whether a safe lane change is viable. If feasible, the algorithm proceeds with the MPC-based trajectory tracking to follow the reference trajectory.

We evaluate three speed levels for vehicle V1: $v \in \{2, 4, 6\} m/s$. At $t = 0s$, when V1 operates at lower speeds of 2 m/s or 4 m/s, the algorithm identifies two priority PLEVs: PLEV 1 (on the same lane as AV) and V1 (on the adjacent lane). The sampling-based planner then generates a safe lane change maneuver x_{ref} to avoid collision with both priority agents (cf. Figure 6.13 (a), black dotted lines). By $t = 4.0s$, the AV successfully tracks the reference lane change trajectory (cf. Figure 6.13 (b)). Subsequently, the priority target shifts from PLEV 1 to PLEV 2, and the reference trajectory transitions to a lane-keeping motion.

However, when vehicle V1 moves at a high speed of 6 m/s, the proposed multi-priority-based trajectory sampling algorithm could not find a safe lane

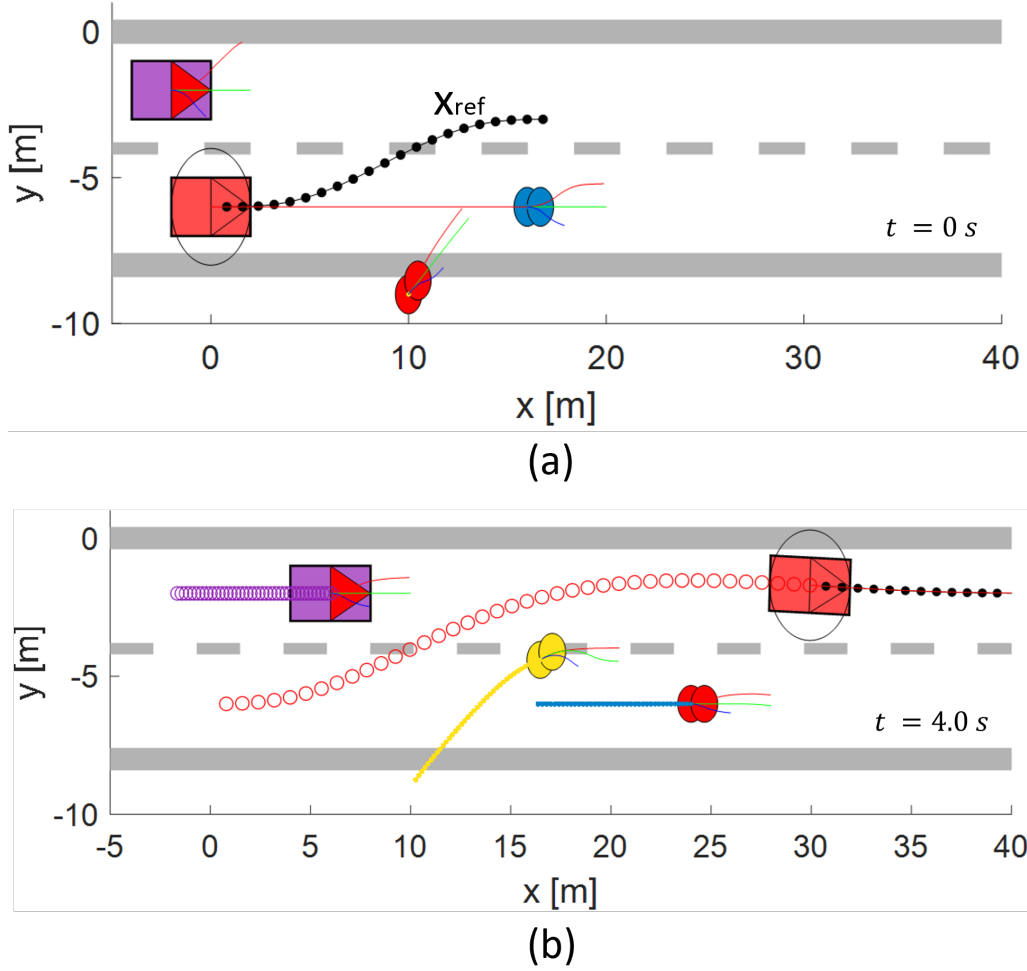


Figure 6.13: A scenario using the proposed multi-priority collision avoidance strategy: (a) At $t = 0$ s, AV overtakes the priority targets PLEV 1 (on the same lane as AV) and vehicle, V1 (on the adjacent lane), (b) then by $t = 4$ s, AV overtakes the new priority target PLEV 2 (in red)

change maneuver. As seen in Figure 6.14 (a), at $t = 1.5$ s, the AV decelerates to wait for V1 to pass to avoid collision with the two priority agents as indicated by the predicted F-PIDP and PIDP of the agents on the inter-distance plot (cf. Figure 6.14 (b)). Then, at $t = 6.0$ s the AV is able to complete the lane change maneuver (cf. Figure 6.14 (c)) with the inter-distance plot showing no predicted collision (all PIDP are above d_{safe}) (cf. Figure 6.14 (d)).

6.7.3 Comparative Analysis of the proposed MiRA-MOC

A cut-in scenario was simulated, where a PLEV swiftly enters the road as it tries to overtake another vehicle (cf. Fig. 6.15). This scenario emulates a realistic challenge for autonomous navigation, characterized by sudden and unpredictable maneuvers from vulnerable road users. We focus on PLEV's motion modeled with a spectrum of object detection and tracking errors up to ($\Sigma_T = 1.0$ m) [138], introducing significant uncertainty into PLEV's state estimation. The proposed algorithm, utilizing the multi-level optimization with F-sPIDP, was

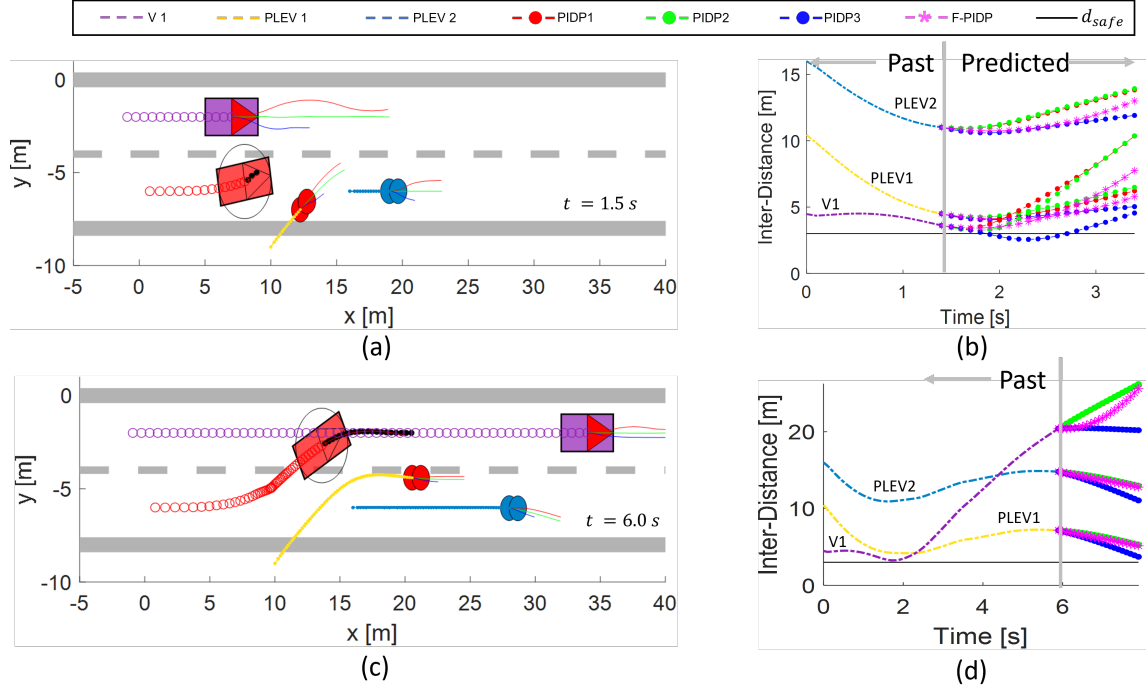


Figure 6.14: A scenario using the proposed multi-priority collision avoidance strategy: (a) At $t = 1.5$ s, AV cannot overtake the priority targets PLEV 1 and V1 (in red), (b) predicted F-PIDP and PIDP of the PLEVs on the inter-distance plot indicates possible collision, (c) then at $t = 6.0$ s, the AV is able to complete the lane change maneuver, (d) with the inter-distance plot showing no predicted collision (all PIDP are above d_{safe})

benchmarked against the SMPC-based Chance Constraint Optimization (CC-Opt) method, a well-established baseline for obstacle avoidance under uncertainty [160]. (cf. video shown through this link <https://youtu.be/Ak68s3a9Mos?list=PL0zhSFDHifx2ruWjKJlggBCKfnzertlLu>)

Trajectory Performance: Both methods successfully avoid collisions with the PLEV by computing a local optimal trajectory that accounts for all possible PLEV trajectories within the uncertainty bounds. However, the baseline (CC-Opt) approach results in a reactive AV motion, excessively influenced by the PLEV's uncertain behavior. The AV's trajectory exhibits pronounced oscillations (wavy motion cf. Fig. 6.15 (a)) as it adjusts to the broad envelope of potential PLEV trajectories, compromising ride comfort and lane discipline.

In contrast, MiRA-MOC employs sampling-based optimization to generate collision-free trajectories that inherently respect desired lane-keeping and lane-changing maneuvers. By integrating stochastic uncertainty into the F-sPIDP model, the algorithm selects an optimal trajectory and an $F\text{-sPIDP}_{ref}$ that balances safety and smoothness, which are then refined via local control optimization to adhere to vehicle dynamics. This results in a significantly less erratic AV motion, maintaining smoother transitions despite the sudden cut-in and uncertainties in the motion of the PLEV (cf. Fig. 6.15 (b)).

Safety Distance Metrics: A critical metric for assessing safety is the minimum inter-distance between the AV and other vehicles. For clarity, we focus on the

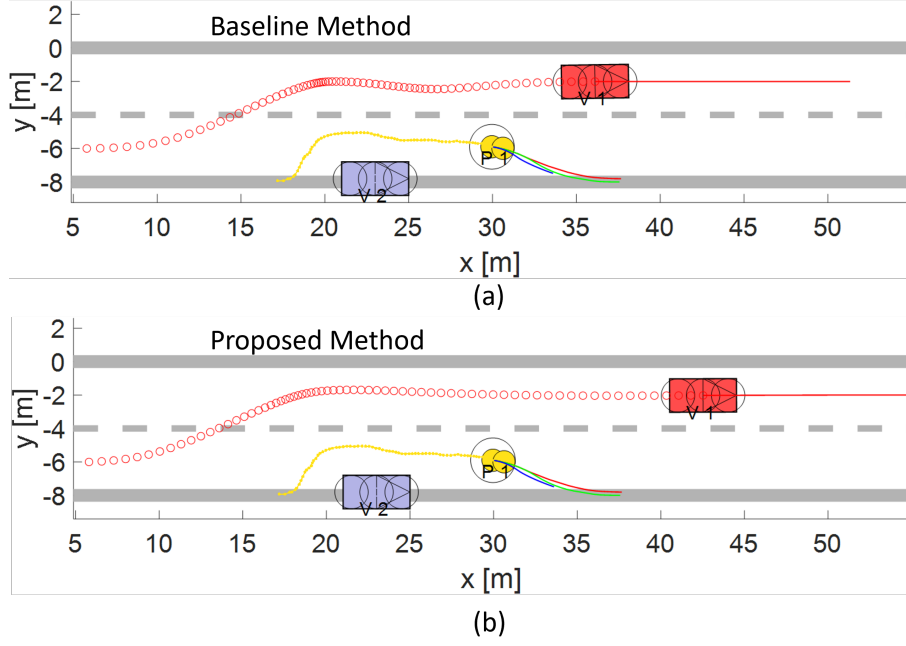


Figure 6.15: Comparison of the trajectory performance: (a) Baseline results in oscillatory AV motion, excessively influenced by the PLEVs' uncertain behavior, (b) Proposed method shows little oscillation by generating collision-free trajectories that respect desired lane-changing and lane-keeping maneuvers while accounting for uncertainties in the PLEV motion

interaction between the AV and PLEV. We performed multiple 10-second simulation runs and set the desired safety distance to 3 meters. Fig. 6.16 ((a) & (b)) shows the minimum inter-distance between the AV and the PLEV for different levels of uncertainty across multiple simulation runs. For the baseline method, the AV maintains an average safety clearance around the 3-meter threshold across most runs; however, the minimum inter-distance occasionally drops below the set 3-meter threshold (cf. Fig. 6.16(a)). This breach occurs because the chance constraints are optimized locally against a broad uncertainty, resulting in reactive adjustments to the AV motion. Increasing the desired safety distance (e.g., to 5 meters) to mitigate this issue often renders the problem infeasible due to the narrow road geometry in the cut-in scenario, highlighting a limitation in the baseline's adaptability.

In contrast, the proposed method (MiRA-MOC) (cf. Fig. 6.16(b)) consistently achieves a robust safety margin, maintaining a minimum inter-distance that exceeds the 3-meter threshold by up to 1 meter across the varying uncertainty levels. This large clearance stems from the sampling-based optimization stage, which leverages the F-sPIDP to select a reference trajectory that inherently respects safety constraints while accounting for uncertainties in the motion of all the PLEVs (cf. Fig. 6.12). By sampling trajectories aligned with lane-keeping and lane-changing intents, the proposed method avoids the reactive corrections, ensuring a stable and safe path even in narrow conditions. The subsequent local optimization further refines this trajectory to maintain vehicle dynamics without

compromising the established clearance. (cf. **video shown through this link** <https://youtu.be/LdUf5fq4fPs?list=PL0zhSFDHifx2ruWjKJ1gggCKfnzertlLu>)

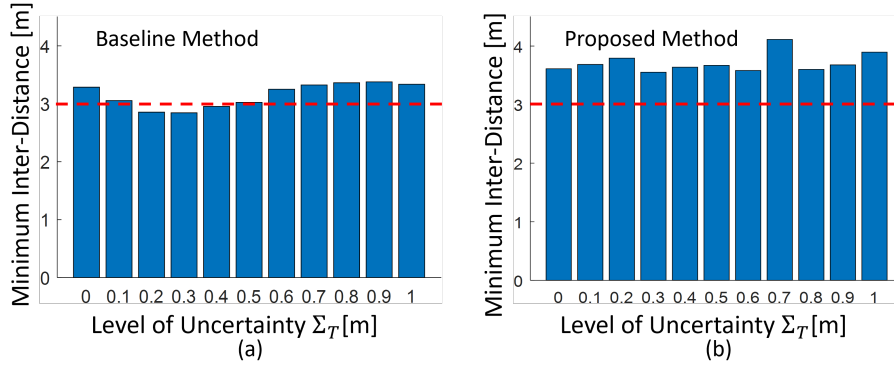


Figure 6.16: Minimum inter-distance between AV and PLEV over time for varying level of uncertainty $\Sigma_T = [0, 1]m$: (a) Baseline showing occasional drops below ($d_{safe} = 3$ m), (b) Proposed method maintaining clearance up to 1 m above desired d_{safe}

The average computation time of the sampling algorithm is 75 ms, while the direct local control refinement optimization takes 20 ms. Hence, the average computation time for the proposed MiRA-MOC is about 95 ms, compared to $\geq 170ms$ for SMPC-based collision avoidance (cf. Table 6.1). The sampling-based algorithm can achieve significant speedups by parallelizing the processing of sampled trajectories. Furthermore, other sub-optimal trajectory samples can serve as alternative solutions for evasive maneuvers in extreme situations.

6.8 CONCLUSION

This chapter presented the proposed Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) framework for uncertainty-aware safe trajectory planning and control of autonomous vehicles in dynamic environments. In this framework, F-sPIDP captures the multi-modal uncertainties in the motion of surrounding agents, enabling the generation of safe and efficient trajectories. MiRA-MOC combines sampling-based lateral and longitudinal trajectory generation in the Frenet frame with an optimal safe trajectory selection process based on F-sPIDP. The local control optimization problem is then formulated as an uncertainty-aware tracking optimization problem that integrates vehicle dynamics, kinematic constraints, and uncertainty-aware safety margins. Simulation results demonstrate the effectiveness of the proposed methods in ensuring safe and efficient navigation in various traffic scenarios, outperforming SMPC-based collision avoidance methods. Additional simulation results and experimental validations of the methods are currently in progress (cf. Appendix A & B).

GENERAL CONCLUSION AND PROSPECTS

7.1 SUMMARY AND CONCLUSIONS

This research investigated in this PhD thesis has presented a comprehensive exploration of risk-aware motion planning and control for autonomous vehicles operating in complex urban environments populated by vulnerable road users, including pedestrians using or not Personal Light Electric Vehicles (PLEVs). Through progressive chapters, this dissertation has developed a framework that integrates uncertainty modeling, probabilistic multi-modal motion prediction, and hierarchical optimization into a single unified approach capable of addressing the uncertainty, multi-modality, and social behaviors inherent to modern urban mobility.

The early chapters established the contextual foundations. Chapter 1 motivated the research by identifying the emerging challenge posed by PLEVs: agents that blur the traditional dichotomy between pedestrians and large road vehicles and thus expose the limits of deterministic autonomous driving systems. Chapter 2 situated the study within the broader landscape of autonomous urban driving research, reviewing architectures, uncertainty quantification, multi-modal prediction, and the evolving role of infrastructure support (RSUs, V2X). Chapter 3 refined this foundation by analyzing planning, risk assessment & management, and control methodologies: spanning classical algorithms, probabilistic formulations, and learning-based hybrids, while emphasizing the necessity of explicit uncertainty treatment and compliance with safety standards.

Building on these insights, the subsequent chapters progressively designed, formalized, and validated a set of complementary contributions:

- Chapters 4 and 5 introduced the **Fusion of (stochastic) Predictive Inter-Distance Profiles** (F-PIDP and F-sPIDP), which unify multi-modal motion predictions and estimation uncertainties into an interpretable, quantitative risk metric for multi-risk assessment and management. This family of constructs bridges the gap between worst-case conservatism and single-hypothesis optimism by probabilistically weighting multiple potential futures.

These metrics were embedded within risk-aware control formulations, including Stochastic and Uncertainty-aware Model Predictive Controllers (SMPC and U-MPC), which dynamically adapt safety margins based on perception noise and prediction variance. The proposed **Multi-Priority-based Target Selection Algorithm** ensured tractability for scenarios with multiple PLEVs without compromising safety.

- Chapter 6 extended the framework to a **Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC)** architecture that

couples a sampling-based global planner (in Frenet coordinates) with an uncertainty-aware local control optimizer. This decoupled design reconciles global exploration with real-time feasibility, maintaining robust and comfortable motion even under partial observability.

Across simulation and mixed-reality evaluations, the integrated framework demonstrated consistent improvements in safety margins, trajectory smoothness, and computational efficiency relative to baseline planners. The findings collectively validate the central thesis: Explicitly modeling multi-modal, stochastic agent behaviors and embedding these representations into a multi-level planning and control hierarchy enables safer, smoother, and more human-compatible autonomous navigation in mixed-traffic urban environments with PLEVs.

From a broader perspective, this research advances the adaptability of autonomous driving systems to probabilistic decision-making, acknowledging the inherently uncertain characteristics of personal mobility systems. It also highlights that practical autonomy requires not only technical sophistication but also interpretability and certifiability.

7.2 PROSPECTS

Building on the proposed Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) architecture presented in this thesis, several promising research and development directions emerge. These prospects emphasize deeper integration between perception, prediction, decision-making, and human–infrastructure cooperation:

Extensive Experimentation and Testing of Rare Cases

For safe deployment, theoretical and simulation-based validation must be complemented by extensive experimental campaigns targeting rare, high-risk, or edge-case scenarios that are statistically underrepresented in current datasets. These include unpredictable PLEV maneuvers, multi-agent conflicts, or degraded sensing environments. Future research could therefore develop comprehensive scenario libraries and mixed-reality testbeds combining simulated agents with real autonomous platforms. Such setups will enable controlled reproduction of rare events, safe stress testing of algorithms, and data collection for continual model improvement. Therefore, testing the proposed MiRA-MOC architecture in long field trials conducted across varied environments, seasons, and cultural contexts will provide crucial insights into long-term reliability, behavioral consistency, and user trust. Through systematic experimentation, the framework’s probabilistic safety principles can evolve into evidence-backed certification pathways aligned with international standards for autonomous systems.

Advanced Prediction through Infrastructure and Cooperative Autonomy

Future progress in autonomous navigation will rely on cooperative perception and prediction, extending beyond the ego-vehicle's onboard sensors. Integrating roadside units (RSUs), vehicle-to-everything (V2X) communication, and shared situational awareness frameworks (cf. Section 2.4.2) can substantially reduce perceptual uncertainty, particularly in occluded or congested urban environments populated by PLEVs and other vulnerable road users. By combining infrastructure-based sensing with inter-vehicle cooperation, prediction modules can access broader spatial context and more accurate intent estimation, leading to safer and smoother navigation decisions. Future research could therefore extend the application of the proposed MiRA-MOC architecture to distributed prediction architectures, multi-agent data fusion protocols, and real-time synchronization mechanisms that jointly model the behavior of all participants in a shared traffic ecosystem. Such cooperative autonomy will transform the current vehicle-centric paradigm into a networked intelligence, enabling predictive anticipation of hazards beyond line of sight and supporting collective risk minimization across agents.

Learning Adaptive Risk Fusion Online

While this thesis introduced the Fusion of stochastic Predictive Inter-Distance Profiles (F-sPIDP) as a principled framework for multi-risk-aware decision-making, future work could move towards continuously learning risk fusion mechanisms online. This is because during deployments, uncertainty distributions, sensor fidelity, and environmental conditions evolve dynamically. Hence, pre-deployment safety thresholds or confidence margins may be insufficient. An emerging direction is to embed online learning and meta-adaptive controllers that continuously adjust the parameters of the F-sPIDP model based on contextual cues, such as traffic density, weather, mission criticality, or observed behavioral deviations of other agents. Leveraging learning methods could enable self-calibrating risk fusion, where the system autonomously refines its conservatism-to-efficiency balance through real-world experience. Such adaptability is essential for maintaining both robustness and performance as AVs encounter novel conditions beyond their pre-training domain.

Human-Centered and Societal Integration

Ultimately, technological progress in autonomy must align with human expectations, ethical norms, and societal acceptance. As autonomous vehicles interact with pedestrians, cyclists, and users of PLEVs, their behavior must not only be safe but also perceived as understandable, courteous, and predictable. Future work should therefore adopt a human-centered design philosophy, combining engineering rigor with behavioral and social sciences. Conducting user studies and participatory evaluations of the proposed MiRA-MOC architecture will help determine acceptable balances between assertiveness and caution, and inform the

design of decision policies that reflect local driving culture and public comfort. Furthermore, collaboration with policymakers and urban planners is needed to ensure that algorithmic choices such as probabilistic risk thresholds or cooperation priorities align with societal values and regulatory frameworks. Such interdisciplinary engagement will be pivotal for fostering trust and ensuring that risk-aware autonomy becomes a socially integrated, ethically responsible technology rather than a purely technical achievement.

SIMULATION ENVIRONMENT

A.1 INTRODUCTION

The development and validation of autonomous vehicle (AV) systems, particularly those incorporating risk assessment and management architectures, require rigorous testing in controlled environments to ensure safety, reliability, and efficacy. Real-world experiments pose significant challenges, including high costs, logistical complexities, and potential risks to human participants and infrastructure. The validation process follows a progressive testing methodology: starting with fully virtual simulations for rapid prototyping and algorithm evaluation, followed by mixed hardware-in-the-loop (HIL) [77] (in Appendix B.1) experiments that bridge simulation and real-world implementation. The objective is to ensure that the proposed methods operate reliably under realistic dynamic scenarios, while maintaining safety and repeatability during testing.

A.2 SIMULATION TEST ENVIRONMENT

To evaluate the proposed risk assessment and management architecture, extensive simulations were conducted. These simulations served dual purposes: (1) to validate the algorithmic components in isolation under controlled conditions, and (2) to assess the system's behavior in dynamic, interactive scenarios mimicking real-world shared spaces. The simulation framework was designed to support testing in fully virtual environments as well as mixed setups that integrate virtual elements with physical hardware. This approach allowed for iterative refinement of the algorithms without the immediate risks associated with physical deployments [184].

A.2.1 *The 2D Simulation Environment*

The majority of initial tests were implemented in a custom-built 2D environment using MATLAB, chosen for its rapid prototyping capabilities and extensive libraries for numerical computation, optimization, and visualization. This environment facilitated quick iterations on the proposed algorithms, enabling the evaluation of risk assessment metrics, motion planning strategies, and management protocols under various uncertainty levels. The 2D simulation environment consists of two primary components:

- Backend computational module: this module executes the core algorithms for risk assessment and management. It simulates the dynamics of the AV, PLEVs, and other environmental agents using kinematic and dynamic models (cf. Chapter 5 & 6). The backend handles computations such as

multi-level optimization for trajectory generation (cf. Figure 4.1), and real-time updates to agent states based on simulated sensor data. Environmental uncertainties, such as sensor noise and PLEV behavioral variability, were modeled stochastically to reflect realistic conditions.

- Frontend visualization interface: Provides a graphical visualization of agents and the environment, allowing observation of positions, velocities, trajectories, collision avoidance behavior, and dynamic risk levels of the inter-distance plots (cf. Figure A.1).

The 2D MATLAB simulation environment enables rapid iteration and testing of algorithms in controlled conditions, before transitioning to more complex 3D and mixed-reality simulations.

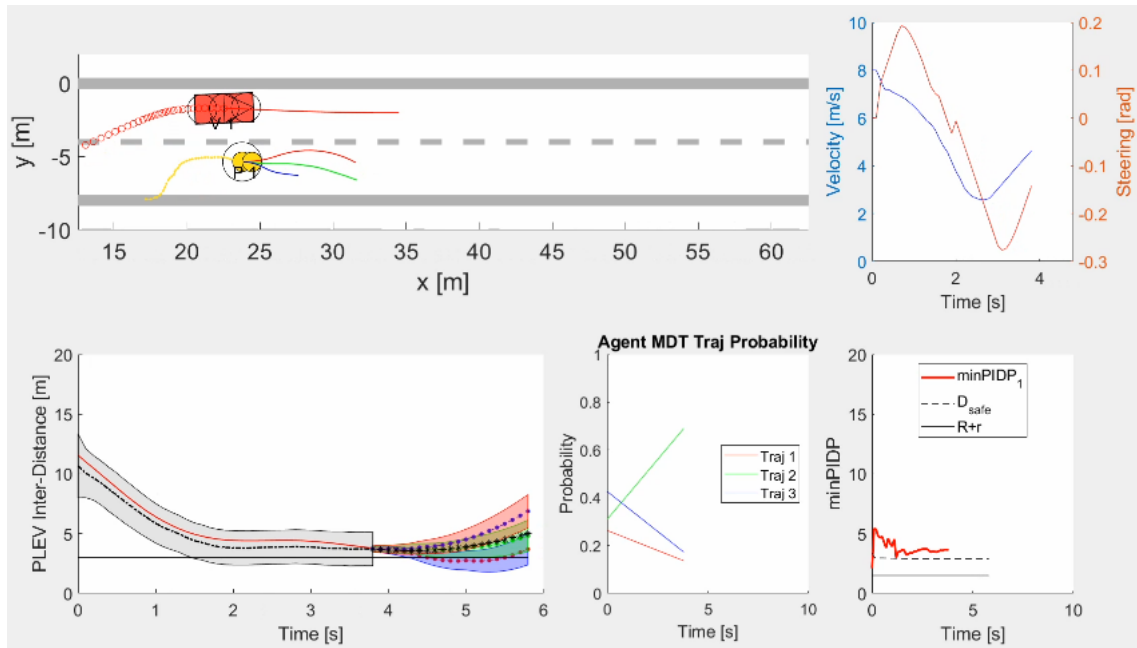


Figure A.1: A snapshot of the 2D simulation environment

A.2.2 State-of-the-Art Simulation Environments

Conducting experiments with physical AVs in real-world settings is inherently challenging due to the complexities of environment setup, ethical considerations, and the potential for harm to human actors or property. For instance, replicating dangerous scenarios, such as near-collision or aggressive maneuvers of PLEV behaviors in a controlled outdoor space requires significant resources and safety protocols. Consequently, realistic simulation environments are essential for safe, repeatable testing of autonomous systems [137].

A.2.3 Review of Simulation Platforms

In recent years, several advanced simulators have been developed to support the design and validation of AV technologies. These tools provide high-fidelity

modeling of vehicle dynamics, sensor suites, and environmental interactions. Key examples include [242]:

- SCANeR Studio¹: A commercial simulator focused on automotive testing, offering detailed vehicle models and integration with hardware-in-the-loop systems. It excels in scenario-based testing for advanced driver-assistance systems (ADAS) but requires proprietary licensing.
- CARLA²: An open-source simulator built on Unreal Engine, designed specifically for autonomous driving research. It supports realistic urban environments, dynamic weather conditions, and multi-agent interactions, with built-in support for sensors like LiDAR, cameras, and GPS.
- IPG CarMaker³: A professional-grade tool emphasizing vehicle dynamics and control systems, widely used in the automotive industry for virtual prototyping and validation.
- Unreal Engine⁴: A general-purpose game engine adapted for AV simulations, providing high-quality graphics and physics but requiring custom scripting for specialized scenarios.
- AWSIM⁵: An open-source simulator leveraging Unity, suitable for cloud-based testing and integration with AWS services for scalable computations.
- LGSVL Simulator⁶: Now part of the LG Electronics ecosystem, it focuses on sensor simulation and integration, making it ideal for perception and control algorithm testing.

A.2.4 *Selected Simulation Platform: CARLA*

In this work, CARLA simulator was selected for the 3D simulation components of this work due to its open-source nature, extensive community support, and compatibility with Robot Operating System 2 (ROS2) middleware [124]. CARLA's ability to simulate complex urban scenarios with pedestrian and vehicle agents aligned well with the focus on urban driving involving PLEVs [57]. Additionally, its Python API enabled seamless integration with the proposed MATLAB-based algorithms, allowing for hybrid simulations where virtual agents interact with real hardware states.

¹ <https://www.avsimulation.com/fr/scaner/>

² <https://carla.org/>

³ <https://www.ipg-automotive.com/en/products-solutions/software/carmaker/>

⁴ <https://www.unrealengine.com/en-US>

⁵ <https://autoware.org/awsim-end-to-end-digital-twin-simulation-platform/>

⁶ <https://hidetoshi-furukawa.github.io/post/lgsvl-simulator/>

EXPERIMENTAL PLATFORM

B.1 INTRODUCTION

To transition from simulation to real-world applicability, a Robotized Autonomous Vehicle (RAV) was used as the experimental platform. This platform served as the foundation for hardware-in-the-loop (HIL) testing [77], ensuring that the proposed Multi-Risk Assessment and Multi-level Motion Optimization and Control (MiRA-MOC) architecture could operate within physical constraints while minimizing risks.

B.1.1 *Robotized Vehicle Platform*

The primary experimental vehicle is a robotized Renault Zoe electric car housed in the Heudiasyc Laboratory. This platform enables autonomous operation through the control of various actuators for steering, acceleration, and braking, controlled via a custom onboard computer. The vehicle is equipped with a comprehensive sensor suite to support perception, localization, and navigation:

- **Localization Sensors:** High-precision GPS (RTK-enabled) and inertial measurement units (IMUs) for accurate positioning in outdoor environments.
- **Perception Sensors:** LiDAR (e.g., 360° Velodyne) for 3D point cloud mapping and obstacle detection, multiple cameras (RGB and stereo) for visual odometry and object recognition.
- **Control Interface:** the vehicle is controlled using ROS2, a middleware framework that facilitates modular integration of software components developed in diverse languages (e.g., MATLAB, Python, C++). This setup forms a complete autonomous navigation stack, including modules for path planning, obstacle avoidance, and low-level control.

The Renault Zoe was chosen for its compact size, electric propulsion (reducing emissions during indoor/outdoor tests), and compatibility with laboratory infrastructure. All experiments were conducted in a controlled environment to ensure safety.

B.1.2 *Test Software Architecture*

The test software architecture extends the simulation environment to incorporate the real robotized autonomous vehicle (RAV) in a hardware-in-the-loop (HIL) configuration. This hybrid setup bridges the gap between virtual simulations and

physical deployments, allowing the AV to interact with simulated obstacles as if they were real.

As depicted in Figure B.1, the architecture comprises three interconnected components:

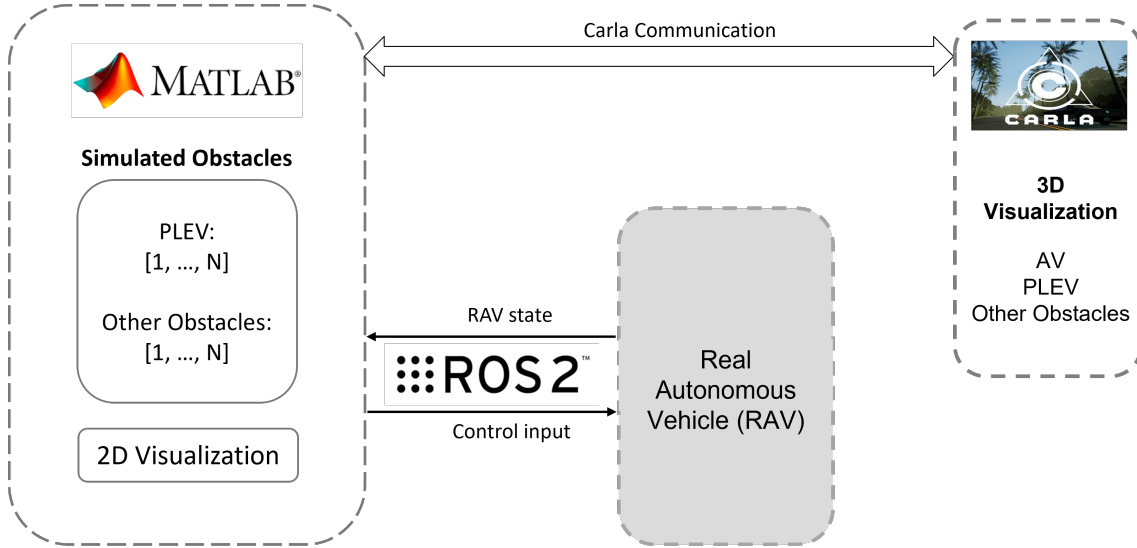


Figure B.1: Software Architecture for Real test of the proposed MiRA-MOC architecture

- **MATLAB Computational Module:** the risk assessment and management algorithms run here, processing inputs from both virtual and real sources to generate optimized trajectories.
- **Real/Robotized Autonomous Vehicle (RAV):** provides real state data (e.g., position, velocity) as inputs to the MiRA-MOC and takes computed control commands via ROS2 topics.
- **CARLA Simulator:** used for visualization and simulation of environmental elements, such as PLEVs and static obstacles. The states of these virtual agents are updated based on the models from Chapter 5, while the AV's state is synchronized with the RAV.

In this HIL setup, virtual obstacles are provided to the RAV system, enabling safe testing of collision avoidance maneuvers. ROS2 bridges facilitate data exchange, ensuring low-latency communication between the RAV and the simulator. This configuration allows evaluation of the multi-level motion optimization and control architecture in scenarios where the AV navigates around virtual PLEVs, mitigating risks associated with using real human actors.

B.1.3 Implementation Results

The mixed-reality setup was used to replicate some of the test scenarios evaluated previously in simulation (cf. Chapter 5 & 6). The experiments demonstrated that the proposed MiRA-MOC algorithm behaves consistently across both domains.

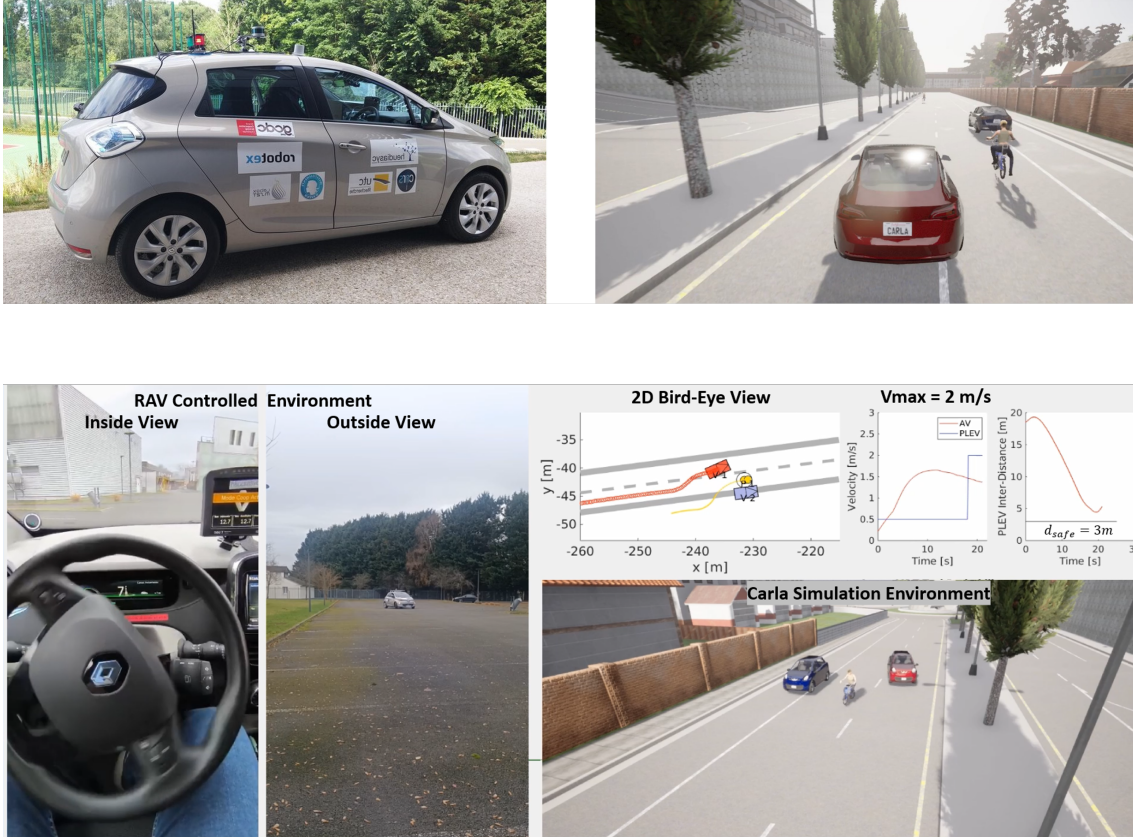


Figure B.2: Visualization of Real Autonomous Vehicle (available in Heudiasyc Lab) and Virtual Environment (cf. **video shown through this link** <https://youtu.be/zcYiv6yFLZ0?list=PL0zhSFDHifx2ruWjKJ1ggbCKfnzert1Lu>)

Figure B.2(Top Left) shows the physical RAV in a controlled laboratory environment, while Figure B.2(Top Right) illustrates the corresponding virtual scene in CARLA, with the red vehicle representing the synchronized RAV state and other elements depicting virtual agents. Figure B.2(Bottom) shows various perspectives of the RAV and other traffic participants as their states evolve in the environment.

During tests, the RAV successfully detected virtual obstacles and adapted its trajectory to avoid collisions. The results (Figure B.2) indicate a high correspondence between the simulated trajectories and real vehicle responses, confirming the accuracy and robustness of the risk assessment and management framework.

This consistency validates both the transferability of the proposed algorithms and the effectiveness of the HIL setup in approximating real-world dynamics. The system demonstrated reliable risk mitigation behaviors, such as safe lane disciplined lateral avoidance, aligning closely with theoretical predictions and simulation results. Work is currently in progress to perform further experimentation to validate and refine these findings.

BIBLIOGRAPHY

- [1] L. Adouane, "Reactive versus cognitive vehicle navigation based on optimal local and global pelc*," *Robotics and Autonomous Systems*, vol. 88, pp. 51–70, 2017 (cit. on pp. 44, 49).
- [2] L. Adouane, "Toward fully autonomous vehicle navigation: from behavioral to hybrid multi-controller architectures," in *2017 11th International Workshop on Robot Motion and Control (RoMoCo)*, IEEE, 2017, pp. 85–98 (cit. on p. 49).
- [3] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971 (cit. on pp. 39, 40).
- [4] E. Alao, L. Adouane, and P. Martinet, "Multi-risk assessment and management in the presence of personal light electric vehicles," in *21st International Conference on Informatics in Control, Automation and Robotics*, vol. 1, 2024, pp. 137–145 (cit. on pp. 93, 95, 98–100, 123).
- [5] E. Alao, L. Adouane, and P. Martinet, "Reliable risk assessment and management using probabilistic fusion of predictive inter-distance profile for urban autonomous driving," in *2024 European Control Conference (ECC)*, IEEE, 2024, pp. 3747–3753 (cit. on pp. 23, 24, 54, 57, 104, 122).
- [6] E. Alao, L. Adouane, and P. Martinet, "Reliable multi-level optimization for safe predictive control of autonomous vehicles to avoid uncertain multimodal plevs," *International Conference on Intelligent Robots and Systems (IROS)*, 2025 (cit. on p. 76).
- [7] B. L. Allen, B. T. Shin, and P. J. Cooper, "Analysis of traffic conflicts and collisions," *Tech. Rep.*, 1978 (cit. on pp. 56, 57).
- [8] F. Altché and A. de La Fortelle, "An lstm network for highway trajectory prediction," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 353–359 (cit. on p. 60).
- [9] O. Amidi and C. E. Thorpe, "Integrated mobile robot control," in *Mobile Robots V*, SPIE, vol. 1388, 1991, pp. 504–523 (cit. on p. 53).
- [10] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, 2019 (cit. on p. 121).
- [11] S. A. Bagloee, M. Tavana, M. Asadi, and T. Oliver, "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies," *Journal of modern transportation*, vol. 24, no. 4, pp. 284–303, 2016 (cit. on p. 29).

- [12] N. L. Baisa, "Derivation of a constant velocity motion model for visual tracking," *arXiv preprint arXiv:2005.00844*, 2020 (cit. on p. 36).
- [13] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018 (cit. on p. 60).
- [14] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A robust scenario mpc approach for uncertain multi-modal obstacles," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 947–952, 2020 (cit. on p. 54).
- [15] F. Batsch, A. Daneshkhah, M. Cheah, S. Kanarachos, and A. Baxendale, "Performance boundary identification for the evaluation of automated vehicles using gaussian process classification," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 419–424 (cit. on p. 59).
- [16] K. Bellingard, L. Adouane, and F. Peyrin, "Risk assessment and management based on neuro-fuzzy system for safe and flexible navigation in unsignalized intersection," in *IEEE Intelligent Vehicles Symposium (IV)*, 2023, pp. 1–7 (cit. on pp. 57, 72, 80, 95).
- [17] K. Bellingard, L. Adouane, and F. Peyrin, "Safe and adaptive roundabout insertion for autonomous vehicle based limit-cycle and predicted inter-distance profiles," in *European Control Conference (ECC)*, 2023. DOI: 10.23919/ECC57647.2023.10178129 (cit. on pp. 24, 52, 71, 104).
- [18] M. Benrabah, C. Orou Mousse, E. Randriamiarintsoa, R. Chapuis, and R. Aufrère, "A review on traversability risk assessments for autonomous ground vehicles: methods and metrics," *Sensors*, vol. 24, no. 6, p. 1909, 2024 (cit. on p. 60).
- [19] G. Bishop, G. Welch, et al., "An introduction to the kalman filter," *Proc of SIGGRAPH, Course*, vol. 8, no. 27599-23175, p. 41, 2001 (cit. on p. 30).
- [20] M. Bojarski, C. Chen, J. Daw, A. Değirmenci, J. Deri, B. Firner, B. Flepp, S. Gogri, J. Hong, L. Jackel, et al., "The nvidia pilotnet experiments," *arXiv preprint*, 2020 (cit. on p. 34).
- [21] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al., "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016 (cit. on pp. 34, 55, 60).
- [22] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv preprint arXiv:1704.07911*, 2017 (cit. on p. 60).
- [23] S. R. Bonela and B. R. Kadali, "Review of traffic safety evaluation at t-intersections using surrogate safety measures in developing countries context," *IATSS research*, vol. 46, no. 3, pp. 307–321, 2022 (cit. on p. 56).

- [24] M. Bouton, A. Nakhaei, K. Fujimura, and M. J. Kochenderfer, "Cooperation-aware reinforcement learning for merging in dense traffic," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3441–3447 (cit. on p. 31).
- [25] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic mdp-behavior planning for cars," in *2011 14th international IEEE conference on intelligent transportation systems (ITSC)*, 2011, pp. 1537–1542 (cit. on p. 63).
- [26] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous pomdps," in *17th international IEEE conference on intelligent transportation systems (ITSC)*, 2014, pp. 392–399 (cit. on p. 63).
- [27] R. F. Brena, A. A. Aguilera, L. A. Trejo, E. Molino-Minero-Re, and O. Mayora, "Choosing the best sensor fusion method: a machine-learning approach," *Sensors*, vol. 20, no. 8, p. 2350, 2020 (cit. on p. 30).
- [28] A. Broadhurst, S. Baker, and T. Kanade, "Monte carlo road safety reasoning," in *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, 2005, pp. 319–324 (cit. on pp. 58, 59).
- [29] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 22–36, 2021 (cit. on p. 54).
- [30] L. D. Burns and C. Shulgan, *Autonomy: The quest to build the driverless car—and how it will reshape our world*. HarperCollins, 2018 (cit. on p. 29).
- [31] N. Burt and Z. Ahmed, "E-scooter attitudes and risk-taking behaviours: an international systematic literature review and survey responses in the west midlands, united kingdom," *Frontiers in public health*, vol. 11, p. 1 277 378, 2023 (cit. on p. 29).
- [32] S. Burton, L. Gauerhof, and C. Heinzemann, "Making the case for safety of machine learning in highly automated driving," in *International Conference on Computer Safety, Reliability, and Security*, Springer, 2017, pp. 5–16 (cit. on p. 60).
- [33] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "Nuscenes: a multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631 (cit. on p. 30).
- [34] A. Caillot, S. Ouerghi, P. Vasseur, R. Bouteau, and Y. Dupuis, "Survey on cooperative perception in an automotive context," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 204–14 223, 2022 (cit. on p. 41).
- [35] M. Castillo-Lopez, P. Ludvig, S. A. Sajadi-Alamdari, J. L. Sanchez-Lopez, M. A. Olivares-Mendez, and H. Voos, "A real-time approach for chance-constrained motion planning with dynamic obstacles," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3620–3625, 2020 (cit. on p. 54).

- [36] C. Chai, X. Zeng, X. Wu, and X. Wang, "Evaluation and optimization of responsibility-sensitive safety models on autonomous car-following maneuvers," *Transportation research record*, vol. 2674, no. 11, pp. 662–673, 2020 (cit. on p. 56).
- [37] M. Charest-Finn and S. Pejhan, "Model predictive control used in passenger vehicles: an overview," *Machines*, vol. 12, no. 11, 2024 (cit. on p. 54).
- [38] J. Chen, Y. Shimizu, L. Sun, M. Tomizuka, and W. Zhan, "Constrained iterative lqg for real-time chance-constrained gaussian belief space planning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5801–5808 (cit. on p. 54).
- [39] W. Chen, F. He, and H. Dong, "Maneuvering target tracking based on an adaptive variable structure interactive multiple model filtering and smoothing algorithm," *AIP Advances*, vol. 13, no. 4, 2023 (cit. on p. 37).
- [40] X. Chen, M. Treiber, V. Kanagaraj, and H. Li, "Social force models for pedestrian traffic—state of the art," *Transport reviews*, vol. 38, no. 5, pp. 625–653, 2018 (cit. on p. 36).
- [41] W. Chi and M. Q.-H. Meng, "Risk-rrt*: a robot motion planning algorithm for the human robot coexisting environment," in *2017 18th International Conference on Advanced Robotics (ICAR)*, IEEE, 2017, pp. 583–588 (cit. on p. 63).
- [42] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, 2018, pp. 4693–4700 (cit. on pp. 34, 55).
- [43] O.-R. A. D. (Committee, *Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles*. SAE international, 2021 (cit. on p. 28).
- [44] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223 (cit. on p. 30).
- [45] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: an automotive application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2006 (cit. on p. 59).
- [46] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Pittsburgh, PA, Tech. Rep. CMU-RI-TR-92-01, 1992 (cit. on p. 53).
- [47] R. David, *The department of transportation just issued a comprehensive policy on self-driving cars*, 2016. [Online]. Available: <https://www.vox.com/2016/9/19/12966680/department-of-transportation-automated-vehicles> (cit. on p. 28).

- [48] M. Dawood, M. E. El Najjar, C. Cappelle, J. Peng, M. Khalil, and D. Pomorski, "Vehicle geo-localization using imm-ukf multi-sensor data fusion based on virtual 3d city model as a priori information," in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, 2012, pp. 7–12 (cit. on p. 37).
- [49] K. De Asis, J. Hernandez-Garcia, G. Holland, and R. Sutton, "Multi-step reinforcement learning: a unifying algorithm," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018 (cit. on p. 51).
- [50] P. Dendorfer, A. Osep, and L. Leal-Taixé, "Goal-gan: multimodal trajectory prediction based on goal position estimation," in *Proceedings of the Asian Conference on Computer Vision*, 2020 (cit. on p. 105).
- [51] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 1468–1476 (cit. on p. 39).
- [52] E. D. Dickmanns et al., "Vehicles capable of dynamic vision," in *IJCAI*, vol. 97, 1997, pp. 1577–1592 (cit. on p. 27).
- [53] N. Distefano, S. Leonardi, and A. Litrico, "Analysis of driving behavior of micromobility vehicle users at mini-roundabouts," *Applied Sciences*, vol. 14, no. 24, p. 11 944, 2024 (cit. on p. 29).
- [54] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *ann arbor*, vol. 1001, no. 48105, pp. 18–80, 2008 (cit. on pp. 44, 49).
- [55] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous driving in unknown environments," in *Experimental Robotics: The Eleventh International Symposium*, Springer, 2009, pp. 55–64 (cit. on pp. 44, 49).
- [56] S. Dominguez, A. Ali, G. Garcia, and P. Martinet, "Comparison of lateral controllers for autonomous vehicle: experimental results," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1418–1423 (cit. on p. 53).
- [57] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: an open urban driving simulator," in *Conference on robot learning*, PMLR, 2017, pp. 1–16 (cit. on p. 135).
- [58] M. G. Earl and R. D'Andrea, "Multi-vehicle cooperative control using mixed integer linear programming," *Cooperative Control of Distributed Multi-Agent Systems*, pp. 231–259, 2007 (cit. on p. 50).
- [59] A. Eidehall, *Tracking and threat assessment for automotive collision avoidance*. Linkopings Universitet (Sweden), 2007 (cit. on p. 29).
- [60] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 2002 (cit. on p. 59).

- [61] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015 (cit. on p. 28).
- [62] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007 (cit. on pp. 31, 54).
- [63] D. D. Fan, K. Otsu, Y. Kubo, A. Dixit, J. Burdick, and A.-A. Agha-Mohammadi, "Step: stochastic traversability evaluation and planning for risk-aware off-road navigation," *arXiv preprint arXiv:2103.02828*, 2021 (cit. on p. 54).
- [64] D. Fan and P. Shi, "Improvement of dijkstra's algorithm and its application in route planning," in *2010 seventh international conference on fuzzy systems and knowledge discovery*, IEEE, vol. 4, 2010, pp. 1901–1904 (cit. on p. 45).
- [65] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "Prm-rl: long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE international conference on robotics and automation (ICRA)*, 2018, pp. 5113–5120 (cit. on pp. 44, 51).
- [66] E. Fiasché, P. Martinet, and E. Malis, "Towards autonomous robot navigation in human populated environments using an universal sfm and parametrized mpc," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 7422–7427 (cit. on p. 54).
- [67] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The international journal of robotics research*, vol. 17, no. 7, pp. 760–772, 1998 (cit. on p. 54).
- [68] J. Firl and Q. Tran, "Probabilistic maneuver prediction in traffic scenarios," in *ECMR*, 2011, pp. 89–94 (cit. on p. 58).
- [69] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE robotics & automation magazine*, vol. 4, no. 1, pp. 23–33, 2002 (cit. on p. 53).
- [70] D. Fox, W. Burgard, S. Thrun, and A. B. Cremers, "A hybrid collision avoidance method for mobile robots," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 2, 1998, pp. 1238–1243 (cit. on p. 53).
- [71] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, et al., "Event-based vision: a survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020 (cit. on p. 33).
- [72] C. Gámez Serna, A. Lombard, Y. Ruichek, and A. Abbas-Turki, "Gps-based curve estimation for an adaptive pure pursuit algorithm," in *Mexican International Conference on Artificial Intelligence*, Springer, 2016, pp. 497–511 (cit. on p. 53).

- [73] J. D. Gammell, T. D. Barfoot, and S. S. Srinivasa, "Batch informed trees (bit*): informed asymptotically optimal anytime search," *The International Journal of Robotics Research*, vol. 39, no. 5, pp. 543–567, 2020 (cit. on p. 47).
- [74] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *2014 IEEE/RSJ international conference on intelligent robots and systems*, 2014, pp. 2997–3004 (cit. on pp. 44, 47).
- [75] M. S. Gandhi, B. Vlahov, J. Gibson, G. Williams, and E. A. Theodorou, "Robust model predictive path integral control: analysis and performance guarantees," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1423–1430, 2021 (cit. on p. 54).
- [76] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012 (cit. on p. 30).
- [77] Ş. Y. Gelbal, S. Tamilarasan, M. R. Cantas, L. Güvenç, and B. Aksun-Güvenç, "A connected and autonomous vehicle hardware-in-the-loop simulator for developing automated driving algorithms," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 3397–3402 (cit. on pp. 133, 137).
- [78] P. Ghorai, A. Eskandarian, Y.-K. Kim, and G. Mehr, "State estimation and motion prediction of vehicles and vulnerable road users for cooperative autonomous driving: a survey," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 10, pp. 16 983–17 002, 2022 (cit. on p. 23).
- [79] A. González-Sieira, M. Mucients, and A. Bugarín-Diz, "An adaptive multi-resolution state lattice approach for motion planning with uncertainty," in *Robot 2015: Second Iberian Robotics Conference: Advances in Robotics, Volume 1*, Springer, 2015, pp. 257–268 (cit. on p. 49).
- [80] M. Götze, M. Witt, N. Willer, and F. Raisch, "Safety in use and automated driving in consideration of the new iso 21448," *ATZ worldwide*, vol. 125, no. 4, pp. 38–43, 2023 (cit. on p. 61).
- [81] P. Gu, Z. Jing, and L. Wu, "A novel robust interacting multiple model filter for manoeuvring target tracking," *Electronics Letters*, vol. 60, no. 6, e13165, 2024 (cit. on pp. 68, 69, 105).
- [82] T. Gu and J. M. Dolan, "Toward human-like motion planning in urban environments," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 350–355 (cit. on p. 53).
- [83] M. Gulzar, Y. Muhammad, and N. Muhammad, "A survey on motion prediction of pedestrians and vehicles for autonomous driving," *IEEE Access*, vol. 9, pp. 137 957–137 969, 2021 (cit. on p. 23).
- [84] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264 (cit. on pp. 39, 69).

- [85] S. Gyawali, S. Xu, Y. Qian, and R. Q. Hu, "Challenges and solutions for cellular based v2x communications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 222–255, 2020 (cit. on p. 40).
- [86] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-aware motion planning and control using cvar-constrained optimization," *IEEE Robotics and Automation letters*, vol. 4, no. 4, pp. 3924–3931, 2019 (cit. on p. 54).
- [87] P. Han, J. Liu, J. He, Z. Zhang, S. Yang, Y. Tang, and P. Roop, "St crf: spatial-temporal conditional random field for human trajectory prediction," *CoRR*, 2023 (cit. on p. 38).
- [88] A. Haridas, K. Hamadeh, and S. C. B. Dash, "Dadagger: disagreement-augmented dataset aggregation," *arXiv preprint arXiv:2301.01348*, 2023 (cit. on p. 50).
- [89] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968 (cit. on pp. 31, 44, 45).
- [90] A. J. Hawkins, *Waymo still doing better than humans at preventing injuries and property damage*, 2024. [Online]. Available: <https://www.theverge.com/2024/12/19/24324492/waymo-injury-property-damage-insurance-data-swiss-re> (cit. on p. 27).
- [91] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using gaussian process regression," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2736–2743, 2019 (cit. on p. 59).
- [92] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious nmpc with gaussian process dynamics for autonomous miniature race cars," in *2018 European control conference (ECC)*, IEEE, 2018, pp. 1341–1348 (cit. on p. 59).
- [93] T. N. N. Hossein, S. Mita, and H. Long, "Multi-sensor data fusion for autonomous vehicle navigation through adaptive particle filter," in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 752–759 (cit. on p. 30).
- [94] J. Hou, G. F. List, and X. Guo, "New algorithms for computing the time-to-collision in freeway traffic simulation models," *Computational intelligence and neuroscience*, vol. 2014, no. 1, p. 761 047, 2014 (cit. on p. 56).
- [95] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ international conference on intelligent robots and systems*, 2013, pp. 4363–4369 (cit. on pp. 37, 38).
- [96] C. Hu, D. Liu, Z. Wang, D. Li, J. Wang, and X. Tang, "Risk-aware reinforcement learning for non-conservative motion planning in uncertain autonomous driving environments," *IEEE Transactions on Intelligent Transportation Systems*, 2025 (cit. on p. 60).
- [97] J. Huang, Z. He, Y. Arakawa, and B. Dawton, "Trajectory planning in frenet frame via multi-objective optimization," *IEEE Access*, vol. 11, pp. 70 764–70 777, 2023 (cit. on p. 53).

- [98] J. Huang, J. Xie, H. Zhai, Z. Li, and W. Feng, "An adaptive constant acceleration model for maneuvering target tracking," *Remote Sensing*, vol. 17, no. 5, p. 850, 2025 (cit. on p. 36).
- [99] W. Huang, X. Wu, Q. Zhang, N. Wu, and Z. Song, "Trajectory optimization of autonomous driving by differential dynamic programming," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, IEEE, 2014, pp. 1758–1763 (cit. on p. 54).
- [100] T. Hülnhagen, I. Dengler, A. Tamke, T. Dang, and G. Breuel, "Maneuver recognition using probabilistic finite-state machines and fuzzy logic," in *2010 IEEE intelligent vehicles symposium*, 2010, pp. 65–70 (cit. on p. 38).
- [101] C. Hupfer, "Deceleration to safety time (dst)-a useful figure to evaluate traffic safety," in *ICTCT conference proceedings of seminar*, vol. 3, 1997, pp. 5–7 (cit. on p. 57).
- [102] D. Iberraken and L. Adouane, "Safe navigation and evasive maneuvers based on probabilistic multi-controller architecture," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, 2022 (cit. on p. 24).
- [103] D. Iberraken, L. Adouane, and D. 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)*, 2018, pp. 3984–3990 (cit. on pp. 49, 56, 57, 71, 72, 79, 80).
- [104] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7087–7094 (cit. on pp. 44, 51).
- [105] B. Ivanovic, K. Leung, E. Schmerling, and M. Pavone, "Multimodal deep generative models for trajectory prediction: a conditional variational autoencoder approach," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 295–302, 2020 (cit. on p. 68).
- [106] B. Ivanovic and M. Pavone, "The trajectron: probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2375–2384 (cit. on pp. 33, 39).
- [107] M. Jaritz, R. De Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-end race driving with deep reinforcement learning," in *2018 IEEE international conference on robotics and automation (ICRA)*, 2018, pp. 2070–2075 (cit. on p. 60).
- [108] A. Javaid, "Understanding dijkstra's algorithm," *Available at SSRN 2340905*, 2013 (cit. on p. 45).
- [109] F. Jiménez, J. E. Naranjo, and F. García, "An improved method to calculate the time-to-collision of two vehicles," *International Journal of Intelligent Transportation Systems Research*, vol. 11, no. 1, pp. 34–42, 2013 (cit. on p. 56).
- [110] Y. Junko, *A wave of av safety standards to hit in 2020*, 2019. [Online]. Available: <https://www.eetimes.com/a-wave-of-av-safety-standards-to-hit-in-2020/> (cit. on p. 61).

- [111] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011 (cit. on pp. 44, 47, 48, 51).
- [112] C. Katrakazas, M. Quddus, and W.-H. Chen, "A new integrated collision risk assessment methodology for autonomous vehicles," *Accident Analysis & Prevention*, vol. 127, pp. 61–79, 2019 (cit. on p. 58).
- [113] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: state-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015 (cit. on p. 53).
- [114] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 2002 (cit. on pp. 44, 47, 48).
- [115] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2019 international conference on robotics and automation (ICRA)*, IEEE, 2019, pp. 8248–8254 (cit. on pp. 31, 55, 60).
- [116] Z. H. Khattak and Z. Lin, "Quantifying automated vehicle benefits in reducing driving stress: a simulation experiment approach," *Frontiers in future transportation*, vol. 4, p. 1 196 629, 2023 (cit. on p. 27).
- [117] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *2017 IEEE 20Th international conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 399–404 (cit. on p. 60).
- [118] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: a survey," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 6, pp. 4909–4926, 2021 (cit. on p. 55).
- [119] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE transactions on robotics*, vol. 21, no. 3, pp. 354–363, 2005 (cit. on pp. 44–46).
- [120] S. Koenig, M. Likhachev, and D. Furcy, "Lifelong planning a*," *Artificial Intelligence*, vol. 155, no. 1-2, pp. 93–146, 2004 (cit. on pp. 44, 46).
- [121] P. Koopman, "Ul 4600: what to include in an autonomous vehicle safety case," *Computer*, vol. 56, no. 05, pp. 101–104, 2023 (cit. on p. 62).
- [122] P. Koopman, U. Ferrell, F. Fratrick, and M. Wagner, "A safety standard approach for fully autonomous vehicles," in *International Conference on Computer Safety, Reliability, and Security*, Springer, 2019, pp. 326–332 (cit. on p. 62).
- [123] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016 (cit. on p. 32).

- [124] J. Kouril, B. Schäufele, I. Radusch, and B. Schnor, "Performance evaluation of a ros2 based automated driving system," *arXiv preprint arXiv:2411.11607*, 2024 (cit. on p. 135).
- [125] N. M. B. Lakhal, L. Adouane, O. Nasri, and J. B. H. Slama, "Risk management for intelligent vehicles based on interval analysis of ttc," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 338–343, 2019 (cit. on p. 56).
- [126] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Yong, J.-D. Yoder, C. Tay, K. Mekhnacha, and A. Nègre, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 4, pp. 4–19, 2011 (cit. on pp. 38, 39, 56).
- [127] S. LaValle, "Rapidly-exploring random trees: a new tool for path planning," *Research Report 9811*, 1998 (cit. on pp. 44, 47).
- [128] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 336–345 (cit. on p. 39).
- [129] Y.-L. Lee, "Using a new circular prediction algorithm to design an imm filter for low update rate radar system," *Sensors*, vol. 20, no. 18, p. 5035, 2020 (cit. on p. 37).
- [130] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, no. 1, p. 1, 2014 (cit. on pp. 31, 35, 36, 38).
- [131] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *2010 IEEE international conference on robotics and automation*, 2010, pp. 4372–4378 (cit. on p. 30).
- [132] J. Li, B. Dai, X. Li, X. Xu, and D. Liu, "A dynamic bayesian network for vehicle maneuver prediction in highway driving scenarios: framework and verification," *Electronics*, vol. 8, no. 1, p. 40, 2019 (cit. on p. 58).
- [133] L. Li, B. Zhou, J. Lian, X. Wang, and Y. Zhou, "Multi-pptp: multiple probabilistic pedestrian trajectory prediction in the complex junction scene," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13 758–13 768, 2021 (cit. on p. 102).
- [134] Q. Li, R. Li, K. Ji, and W. Dai, "Kalman filter and its application," in *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*, IEEE, 2015, pp. 74–77 (cit. on pp. 104, 105).
- [135] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Transactions on aerospace and electronic systems*, vol. 39, no. 4, pp. 1333–1364, 2003 (cit. on p. 36).
- [136] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. part v. multiple-model methods," *IEEE Transactions on aerospace and electronic systems*, vol. 41, no. 4, pp. 1255–1321, 2005 (cit. on p. 37).

- [137] Y. Li, W. Yuan, S. Zhang, W. Yan, Q. Shen, C. Wang, and M. Yang, "Choose your simulator wisely: a review on open-source simulators for autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 5, pp. 4861–4876, 2024 (cit. on p. 134).
- [138] Z. Li, S. Lan, J. M. Alvarez, and Z. Wu, "Bevnext: reviving dense bev frameworks for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 113–20 123 (cit. on pp. 111, 124).
- [139] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara*: anytime a* with provable bounds on sub-optimality," *Advances in neural information processing systems*, vol. 16, 2003 (cit. on pp. 44, 46).
- [140] Q. Liu, X. Li, Y. Tang, X. Gao, F. Yang, and Z. Li, "Graph reinforcement learning-based decision-making technology for connected and autonomous vehicles: framework, review, and future trends," *Sensors*, vol. 23, no. 19, p. 8229, 2023 (cit. on p. 60).
- [141] S. Liu, X. Wang, O. Hassanin, X. Xu, M. Yang, D. Hurwitz, and X. Wu, "Calibration and evaluation of responsibility-sensitive safety (rss) in automated vehicle performance during cut-in scenarios," *Transportation research part C: emerging technologies*, vol. 125, p. 103 037, 2021 (cit. on p. 62).
- [142] W. Liu, C. Liu, G. Chen, and A. Knoll, "Gaussian process based model predictive control for overtaking in autonomous driving," *Frontiers in Neurorobotics*, vol. 15, p. 723 049, 2021 (cit. on pp. 58, 59).
- [143] X. Liu, X. Zhao, G. Liu, Z. Wu, T. Wang, L. Meng, and Y. Wang, "Immot: a novel 3d multi-object tracking framework with interacting multiple model filter," *arXiv preprint arXiv:2502.09672*, 2025 (cit. on p. 36).
- [144] Y. Liu, B. Li, X. Wang, C. Sammut, and L. Yao, "Attention-aware social graph transformer networks for stochastic trajectory prediction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 5633–5646, 2024 (cit. on p. 39).
- [145] A. Loganathan and N. S. Ahmad, "A systematic review on recent advances in autonomous mobile robot navigation," *Engineering Science and Technology, an International Journal*, vol. 40, p. 101 343, 2023 (cit. on p. 48).
- [146] M. Luber, J. A. Stork, G. D. Tipaldi, and K. O. Arras, "People tracking with human motion predictions from social forces," in *2010 IEEE international conference on robotics and automation*, 2010, pp. 464–469 (cit. on p. 36).
- [147] S. Luo and L. Schomaker, "Reinforcement learning in robotic motion planning by combined experience-based planning and self-imitation learning," *Robotics and Autonomous Systems*, vol. 170, p. 104 545, 2023 (cit. on p. 51).
- [148] J. Ma, Z. Cheng, X. Zhang, Z. Lin, F. L. Lewis, and T. H. Lee, "Local learning enabled iterative linear quadratic regulator for constrained trajectory planning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 5354–5365, 2022 (cit. on p. 54).

- [149] R. Mariani, N. Maor, J. Athavale, and K. Gay, "The importance of interoperability in functional safety standards," *Computer*, vol. 54, no. 3, pp. 80–84, 2021 (cit. on p. 62).
- [150] D. Mayne, "A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems," *International Journal of Control*, vol. 3, no. 1, pp. 85–95, 1966 (cit. on p. 54).
- [151] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000 (cit. on p. 31).
- [152] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 935–942 (cit. on p. 37).
- [153] V. Mehranfar and C. Jones, "Exploring implications and current practices in e-scooter safety: a systematic review," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 107, pp. 321–382, 2024 (cit. on p. 29).
- [154] A. Mesbah, "Stochastic model predictive control: an overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016 (cit. on pp. 54, 85).
- [155] D. Meyer-Delius, C. Plagemann, and W. Burgard, "Probabilistic situation recognition for vehicular traffic scenarios," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 459–464 (cit. on p. 58).
- [156] M. M. Minderhoud and P. H. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Analysis & Prevention*, vol. 33, no. 1, pp. 89–97, 2001 (cit. on p. 56).
- [157] Z. H. Mir, N. Dreyer, T. Kürner, and F. Filali, "Investigation on cellular lte c-v2x network serving vehicular data traffic in realistic urban scenarios," *Future Generation Computer Systems*, vol. 161, pp. 66–80, 2024 (cit. on p. 40).
- [158] R. Molina-Masegosa and J. Gozalvez, "Lte-v for sidelink 5g v2x vehicular communications: a new 5g technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017 (cit. on p. 40).
- [159] M. Mouad, L. Adouane, D. Khadraoui, and P. Martinet, "Mobile robot navigation and obstacles avoidance based on planning and re-planning algorithm," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 622–628, 2012 (cit. on p. 54).
- [160] S. H. Nair, H. Lee, E. Joa, Y. Wang, H. E. Tseng, and F. Borrelli, "Predictive control for autonomous driving with uncertain, multimodal predictions," *IEEE Transactions on Control Systems Technology*, 2024 (cit. on pp. 85, 125).
- [161] D. C. K. Ngai and N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 509–522, 2011 (cit. on p. 63).

- [162] Q. P. Nguyen, Z. Dai, B. K. H. Low, and P. Jaillet, "Value-at-risk optimization with gaussian processes," in *International Conference on Machine Learning*, PMLR, 2021, pp. 8063–8072 (cit. on p. 59).
- [163] M. Okada and B. Gallina, "Safety of the intended functionality of external human interfaces: gaps and research agenda," in *2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2024, pp. 578–583 (cit. on p. 61).
- [164] N. C. Onat, J. Mandouri, M. Kucukvar, B. Sen, S. A. Abbasi, W. Alhajyaseen, A. A. Kutty, R. Jabbar, M. Contestabile, and A. M. Hamouda, "Rebound effects undermine carbon footprint reduction potential of autonomous electric vehicles," *Nature Communications*, vol. 14, no. 1, p. 6258, 2023 (cit. on p. 27).
- [165] R. Palin, D. Ward, I. Habli, and R. Rivett, "Iso 26262 safety cases: compliance and assurance," in *6th IET International Conference on System Safety 2011*, IET, 2011, B12 (cit. on p. 61).
- [166] B. Peng, D. Yu, H. Zhou, X. Xiao, and C. Xie, "A motion planning method for automated vehicles in dynamic traffic scenarios," *Symmetry*, vol. 14, no. 2, p. 208, 2022 (cit. on p. 53).
- [167] L. Petrović, "Motion planning in high-dimensional spaces," *arXiv preprint arXiv:1806.07457*, 2018 (cit. on p. 92).
- [168] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, Munich Germany, 2005, pp. 1–7 (cit. on pp. 44, 48).
- [169] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009 (cit. on pp. 44, 49).
- [170] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Optimal, smooth, nonholonomic mobile robot motion planning in state lattices," *Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-07-15*, 2007 (cit. on p. 49).
- [171] D. A. Pomerleau, "Alvin: an autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988 (cit. on pp. 27, 55).
- [172] J. Pyschny, F. Berger, S. Rothen, J. Denker, M. Frantzen, F. Roder, and S. Kneiphof, "A multiple-input multiple-output radar-based rider assistance system for personal light electric vehicles," *SAE Technical Paper, Tech. Rep.*, 2023 (cit. on p. 23).
- [173] S. J. Qin and T. A. Badgwell, "An overview of industrial model predictive control technology," in *Alche symposium series*, New York, NY: American Institute of Chemical Engineers, 1971-c2002., vol. 93, 1997, pp. 232–256 (cit. on p. 54).

- [174] S. J. Qin and T. A. Badgwell, "An overview of nonlinear model predictive control applications," *Nonlinear model predictive control*, pp. 369–392, 2000 (cit. on p. 54).
- [175] M. M. Rahman and J.-C. Thill, "Impacts of connected and autonomous vehicles on urban transportation and environment: a comprehensive review," *Sustainable Cities and Society*, vol. 96, p. 104649, 2023 (cit. on p. 27).
- [176] S. Rahmani, A. Baghbani, N. Bouguila, and Z. Patterson, "Graph neural networks for intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8846–8885, 2023 (cit. on p. 60).
- [177] R. Rajamani, *Vehicle dynamics and control*. Springer, 2006 (cit. on p. 31).
- [178] R. Rajamani, "Vehicle dynamics and control," *Mechanical engineering*, 2012 (cit. on p. 52).
- [179] V. M. Raju, V. Gupta, and S. Lomate, "Performance of open autonomous vehicle platforms: autoware and apollo," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 2019, pp. 1–5 (cit. on p. 32).
- [180] A. Richards and J. How, "Mixed-integer programming for control," in *Proceedings of the 2005, American Control Conference, 2005.*, IEEE, 2005, pp. 2676–2683 (cit. on pp. 44, 50).
- [181] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–764, 2002 (cit. on pp. 44, 50).
- [182] S. Rodriguez, X. Tang, J.-M. Lien, and N. M. Amato, "An obstacle-based rapidly-exploring random tree," in *Proceedings 2006 IEEE international conference on robotics and automation, 2006. ICRA 2006.*, 2006, pp. 895–900 (cit. on p. 47).
- [183] L. A. Rosero, I. P. Gomes, J. A. R. Da Silva, C. A. Przewodowski, D. F. Wolf, and F. S. Osório, "Integrating modular pipelines with end-to-end learning: a hybrid approach for robust and reliable autonomous driving systems," *Sensors*, vol. 24, no. 7, p. 2097, 2024 (cit. on pp. 31, 33, 35, 36).
- [184] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, p. 648, 2019 (cit. on p. 133).
- [185] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings*, 2011, pp. 627–635 (cit. on pp. 44, 51).
- [186] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: a survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020 (cit. on pp. 23, 35, 38).

- [187] C. Sachs, S. Burandt, S. Mandelj, and R. Mutter, "Assessing the market of light electric vehicles as a potential application for electric in-wheel drives," in *2016 6th International Electric Drives Production Conference (EDPC)*, IEEE, 2016, pp. 280–285 (cit. on p. 23).
- [188] A. Sadeghian, V. Kosaraju, A. Sadeghian, N. Hirose, H. Rezatofighi, and S. Savarese, "Sophie: an attentive gan for predicting paths compliant to social and physical constraints," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1349–1358 (cit. on pp. 39, 40).
- [189] K. Saleh, M. Hossny, and S. Nahavandi, "Contextual recurrent predictive model for long-term intent prediction of vulnerable road users," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3398–3408, 2019 (cit. on p. 39).
- [190] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *arXiv preprint arXiv:1704.02532*, 2017 (cit. on p. 55).
- [191] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: dynamically-feasible trajectory forecasting with heterogeneous data," in *European Conference on Computer Vision*, Springer, 2020, pp. 683–700 (cit. on pp. 34, 39, 40).
- [192] N. Schneider and D. M. Gavrila, "Pedestrian path prediction with recursive bayesian filters: a comparative study," in *German conference on pattern recognition*, Springer, 2013, pp. 174–183 (cit. on p. 37).
- [193] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020 (cit. on pp. 36–38).
- [194] R. Schubert, "Evaluating the utility of driving: toward automated decision making under uncertainty," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 354–364, 2011 (cit. on pp. 55, 57, 62, 63).
- [195] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *2008 11th international conference on information fusion*, IEEE, 2008, pp. 1–6 (cit. on p. 36).
- [196] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014 (cit. on pp. 44, 50).
- [197] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization," in *Robotics: science and systems*, Berlin, Germany, vol. 9, 2013, pp. 1–10 (cit. on p. 50).

- [198] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: modelling, exploring, and exploiting functions," *Journal of mathematical psychology*, vol. 85, pp. 1–16, 2018 (cit. on p. 59).
- [199] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Interaction-aware probabilistic behavior prediction in urban environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3999–4006 (cit. on p. 58).
- [200] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Multiple model unscented kalman filtering in dynamic bayesian networks for intention estimation and trajectory prediction," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 1467–1474 (cit. on p. 63).
- [201] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 187–210, 2018 (cit. on pp. 43, 70).
- [202] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016 (cit. on p. 35).
- [203] S. E. Shladover, "Connected and automated vehicle systems: introduction and overview," *Journal of Intelligent Transportation Systems*, vol. 22, no. 3, pp. 190–200, 2018 (cit. on p. 28).
- [204] D. Singh and R. Srivastava, "Graph neural network with rnns based trajectory prediction of dynamic agents for autonomous vehicle," *Applied Intelligence*, vol. 52, no. 11, pp. 12 801–12 816, 2022 (cit. on p. 60).
- [205] D. Singh, P. Das, and I. Ghosh, "Conflict-based safety evaluations at unsignalized intersections using surrogate safety measures," *Heliyon*, vol. 10, no. 5, 2024 (cit. on pp. 57, 79).
- [206] B. Škugor, J. Deur, V. Ivanovic, and H. E. Tseng, "Stochastic model predictive control of an autonomous vehicle interacting with pedestrians at unsignalized crosswalks," in *European Control Conference (ECC)*, 2023, pp. 1–7 (cit. on p. 54).
- [207] K. Stachowicz and S. Levine, "Racer: epistemic risk-sensitive rl enables fast driving with fewer crashes," *arXiv preprint arXiv:2405.04714*, 2024 (cit. on pp. 59, 60, 63).
- [208] H. Taghavifar, L. Taghavifar, C. Hu, C. Wei, and Y. Qin, "Optimal reinforcement learning and probabilistic-risk-based path planning and following of autonomous vehicles with obstacle avoidance," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 238, no. 6, pp. 1427–1439, 2024 (cit. on p. 60).
- [209] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel, "Value iteration networks," *Advances in neural information processing systems*, vol. 29, 2016 (cit. on pp. 44, 51).

- [210] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1168–1175 (cit. on p. 54).
- [211] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al., "Stanley: the robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006 (cit. on pp. 28, 30, 53).
- [212] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proceedings of the 2005, American Control Conference, 2005.*, IEEE, 2005, pp. 300–306 (cit. on p. 54).
- [213] A.-T. Tran, A. Muraleedharan, H. Okuda, and T. Suzuki, "Scenario-based stochastic mpc for vehicle speed control considering the interaction with pedestrians," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 325–15 331, 2020 (cit. on p. 54).
- [214] R. Trauth, K. Moller, G. Würsching, and J. Betz, "Frenetix: a high-performance and modular motion planning framework for autonomous driving," *IEEE Access*, 2024 (cit. on p. 53).
- [215] S. Ulbrich and M. Maurer, "Probabilistic online pomdp decision making for lane changes in fully automated driving," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 2063–2067 (cit. on p. 63).
- [216] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al., "Autonomous driving in urban environments: boss and the urban challenge," *Journal of field Robotics*, vol. 25, no. 8, pp. 425–466, 2008 (cit. on pp. 32, 62).
- [217] S. A. Useche, A. Gonzalez-Marin, M. Faus, and F. Alonso, "Environmentally friendly, but behaviorally complex? a systematic review of e-scooter riders' psychosocial risk features," *PLoS one*, vol. 17, no. 5, e0268960, 2022 (cit. on p. 29).
- [218] J. Van Den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3475–3482 (cit. on p. 54).
- [219] E. Van Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, and N. Van De Wouw, "Robust model predictive cooperative adaptive cruise control subject to v2v impairments," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8 (cit. on p. 54).
- [220] L. Wang, X. Yang, Z. Chen, and B. Wang, "Application of the improved rapidly exploring random tree algorithm to an insect-like mobile robot in a narrow environment," *Biomimetics*, vol. 8, no. 4, p. 374, 2023 (cit. on p. 47).

- [221] X. Wang, C. Ye, M. Quddus, and A. Morris, "Pedestrian safety in an automated driving environment: calibrating and evaluating the responsibility-sensitive safety model," *Accident Analysis & Prevention*, vol. 192, p. 107 265, 2023 (cit. on p. 62).
- [222] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in automated driving: a survey," *Ieee Access*, vol. 8, pp. 2847–2868, 2019 (cit. on p. 30).
- [223] Z. Wang, Y. Zhuang, Q. Gu, D. Chen, H. Zhang, and W. Liu, "Reinforcement learning based negotiation-aware motion planning of autonomous vehicles," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4532–4537 (cit. on p. 55).
- [224] J. R. Ward, G. Agamennoni, S. Worrall, A. Bender, and E. Nebot, "Extending time to collision for probabilistic reasoning in general traffic scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 51, pp. 66–82, 2015 (cit. on p. 56).
- [225] F. Warg, M. Skoglund, A. Thorsén, R. Johansson, M. Brännström, M. Gyllenhammar, and M. Sanfridson, "The quantitative risk norm-a proposed tailoring of hara for ads," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 86–93 (cit. on p. 61).
- [226] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE international conference on robotics and automation*, 2010, pp. 987–993 (cit. on pp. 31, 53, 70, 117).
- [227] D. Wilkie, J. Van Den Berg, and D. Manocha, "Generalized velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 5573–5578 (cit. on p. 54).
- [228] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: from theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017 (cit. on p. 54).
- [229] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive driving with model predictive path integral control," in *2016 IEEE international conference on robotics and automation (ICRA)*, 2016, pp. 1433–1440 (cit. on p. 54).
- [230] J. Wu, X. Ma, T. Peng, and H. Wang, "An improved timed elastic band (teb) algorithm of autonomous ground vehicle (agv) in complex environment," *Sensors*, vol. 21, no. 24, p. 8312, 2021 (cit. on p. 54).
- [231] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2174–2182 (cit. on p. 60).
- [232] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion planning under uncertainty for on-road autonomous driving," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2507–2512 (cit. on p. 31).

- [233] B. Yang and R. Nevatia, "Multi-target tracking by online learning a crf model of appearance and motion patterns," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 203–217, 2014 (cit. on p. 38).
- [234] M. Yang, H. Yu, and L. Bai, "Simulation-based evaluation of variation in left-turn paths in the coordinated intersection management," *Journal of Advanced Transportation*, vol. 2021, no. 1, p. 6 243 530, 2021 (cit. on p. 57).
- [235] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: a review," *Sensors*, vol. 21, no. 6, p. 2140, 2021 (cit. on p. 30).
- [236] Y. Yoon, C. Kim, J. Lee, and K. Yi, "Interaction-aware probabilistic trajectory prediction of cut-in vehicles using gaussian process for proactive control of autonomous vehicles," *Ieee Access*, vol. 9, pp. 63 440–63 455, 2021 (cit. on p. 59).
- [237] Z. You, K. Shen, T. Huang, Y. Liu, and X. Zhang, "Application of a* algorithm based on extended neighborhood priority search in multi-scenario maps," *Electronics*, vol. 12, no. 4, p. 1004, 2023 (cit. on p. 45).
- [238] W. Yu, Y. Qian, J. Xu, H. Sun, and J. Wang, "Driving decisions for autonomous vehicles in intersection environments: deep reinforcement learning approaches with risk assessment," *World Electric Vehicle Journal*, vol. 14, no. 4, p. 79, 2023 (cit. on p. 60).
- [239] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019 (cit. on p. 60).
- [240] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 222–15 232 (cit. on p. 33).
- [241] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2017 (cit. on p. 59).
- [242] H. Zhao, M. Meng, X. Li, J. Xu, L. Li, and S. Galland, "A survey of autonomous driving frameworks and simulators," *Advanced Engineering Informatics*, vol. 62, p. 102 850, 2024 (cit. on p. 135).
- [243] A. Zhitnikov and V. Indelman, "Simplified continuous high-dimensional belief space planning with adaptive probabilistic belief-dependent constraints," *IEEE Transactions on Robotics*, vol. 40, pp. 1684–1705, 2023 (cit. on p. 92).
- [244] Z. Zhu, L. Adouane, and A. Quilliot, "Flexible multi-unmanned ground vehicles (mugvs) in intersection coordination based on ϵ -constraint probability collectives algorithm," *International Journal of Intelligent Robotics and Applications*, vol. 5, no. 2, pp. 156–175, 2021 (cit. on pp. 56, 58).

- [245] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, et al., “Making bertha drive—an autonomous journey on a historic route,” *IEEE Intelligent transportation systems magazine*, vol. 6, no. 2, pp. 8–20, 2014 (cit. on p. 62).
- [246] J. Ziegler and C. Stiller, “Spatiotemporal state lattices for fast trajectory planning in dynamic on-road driving scenarios,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1879–1884 (cit. on p. 31).
- [247] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, “Chomp: covariant hamiltonian optimization for motion planning,” *The International journal of robotics research*, vol. 32, no. 9-10, pp. 1164–1193, 2013 (cit. on pp. 44, 50).