C2SMART
CONNECTED CITIES WITH SMART TRANSPORTATION
A USDOT University Transportation Center

New York University
Rutgers University
University of Washington
The University of Texas at El Paso
City College of New York

Automated Lane Change and Robust Safety

March 2023

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Automated Lane Change and Robust Safety		5. Report Date March 2023
		6. Performing Organization Code:
7. Author(s) Zhong-Ping Jiang, Kaan Ozbay, Sayan Chakraborty, Leilei Cui		8. Performing Organization Report No.
9. Performing Organization Name and Address Connected Cities for Smart Mobility towards Accessible and Resilient Transportation Center (C2SMART), 6 Metrotech Center, 4th Floor, NYU Tandon School of Engineering, Brooklyn, NY, 11201, United States		10. Work Unit No. 11. Contract or Grant No. 69A3551747119
12. Sponsoring Agency Name and Address Office of Research, Development, and Technology Federal Highway Administration 6300 Georgetown Pike McLean, VA 22101-2296		13. Type of Report and Period Final Report: 3/1/22- 2/28/23 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Firstly, to guarantee stability and robustness in the face of parametric uncertainties, non-linearities, and modeling errors, we have proposed a data-driven optimal control algorithm to solve the lane-changing problem of AVs which is inspired by reinforcement learning and adaptive dynamic programming. Secondly, we have developed a lane change decision-making algorithm to ensure safe and efficient lane change. Thirdly, the lane change risk index (LCRI) is used to evaluate the AV lane change safety obtained by using the proposed data-driven optimal control algorithm. Fourthly, we have combined the data-driven optimal controller with the lane change decision-making algorithm by using control barrier functions (CBFs). Lastly, we have developed an experimental setup that includes prototypes of AV and highway lanes.

17. Key Words	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161. http://www.ntis.gov		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 43	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page author

Automated Lane Change and Robust Safety

Zhong-Ping Jiang
New York University
0000-0002-4868-9359

Kaan Ozbay
New York University
0000-0001-7909-6532

Sayan Chakraborty
New York University
0000-0002-8638-4652

Leilei Cui
New York University
0000-0001-8031-7638

C2SMART Center is a USDOT Tier 1 University Transportation Center taking on some of today's most pressing urban mobility challenges. Some of the areas C2SMART focus es on include:

Urban Mobility and Connected Citizens

Urban Analytics for Smart Cities

Resilient, Smart, & Secure Infrastructure

Disruptive Technologies and their impacts on transportation systems. Our aim is to develop innovative solutions to accelerate technology transfer from the research phase to the real world.

Unconventional Big Data Applications from field tests and non-traditional sensing technologies for decision-makers to address a wide range of urban mobility problems with the best information available.

Impactful Engagement overcoming institutional barriers to innovation to hear and meet the needs of city and state stakeholders, including government agencies, policy makers, the private sector, non -profit organizations, and entrepreneurs.

Forward-thinking Training and Development dedicated to training the workforce of tomorrow to deal with new mobility problems in ways that are not covered in existing transportation curricula.

Led by New York University's Tandon School of Engineering, **C2SMART** is a consortium of leading research universities, including Rutgers University, University of Washington, the University of Texas at El Paso, and The City College of NY.

Visit c2smart.engineering.nyu.edu to learn more

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Acknowledgements

This project is supported in part by the USDOT Tier-1 C2SMART Center grant, in part by an NYU SOE PhD Fellowship awarded to the student, and in part by the US NSF.

Executive Summary

Lately, the transportation industry has been concentrating heavily on the technologies of connected and autonomous vehicles (CAVs). The lane-changing problem of autonomous vehicles has been the central topic of our project. Accidents can occur due to inappropriate lane changes caused by human drivers' inefficiency in predicting and estimating the surrounding environment. Therefore, our project has aimed to develop effective methodologies to enhance the safety and comfort of passengers traveling in AVs by efficiently changing lanes in mixed traffic conditions.

Firstly, to guarantee stability and robustness in the face of parametric uncertainties, non-linearities, and modeling errors, we have proposed a data-driven optimal control algorithm to solve the lane-changing problem of AVs which is inspired by reinforcement learning and adaptive dynamic programming. The robustness and optimality of the proposed data-driven control algorithm are guaranteed by combining gain-scheduling and Linear Quadratic Regulator (LQR) control respectively. Rigorous theoretical analysis and several SUMO simulations are performed to establish the effectiveness of the proposed control algorithm.

Secondly, we have developed a lane change decision-making algorithm to ensure safe and efficient lane change. The lane change decision-making algorithm includes inequalities that check the safe distance of the AV from the surrounding vehicles. The safe distance from each surrounding vehicle is chosen as a function of their respective velocities which ensures more safety for fast-moving vehicles. The proposed lane change decision-making algorithm can make the AV abort any initiated lane change maneuver at any time if the safety conditions are not met. In such scenarios, the proposed decision-making algorithm makes the AV maneuver back to the original lane.

Thirdly, the lane change risk index (LCRI) is used to evaluate the AV lane change safety obtained by using the proposed data-driven optimal control algorithm. The safety of the proposed technique is compared with Model Predictive Control (MPC) and it was found that the proposed technique provided better safety as compared to MPC. Moreover, it is shown that the proposed data-driven optimal control algorithm is also computationally efficient than MPC.

Fourthly, we have combined the data-driven optimal controller with the lane change decision-making algorithm by using control barrier functions (CBFs). The lane change decision-making algorithm includes inequalities that check the position of the AV from the surrounding vehicles and ensures that the AV is at a safe distance from the surrounding vehicles. These inequalities

have been formed in such a way that they satisfy the requirements to behave as CBFs. Note that in the previous formulation of data-driven optimal control for AV lane change, stability was ensured independent of safety. In the new formulation with CBFs, we try to compute an optimal control action that guarantees stability as well as safety. Such formulation has been seen to improve the performance of safety-critical systems like AVs.

Lastly, we have developed an experimental setup that includes prototypes of AV and highway lanes. The computation device on the AV is a Raspberry Pi microcontroller with is connected to a camera and a GPS device. The controller obtained from the proposed data-driven optimal control algorithm is tested on the AV prototypes. It was found that the AV was successful in changing lanes despite the noise in the measurement and uncertainty in the system parameters.

Table of Contents

List of Figures

Automated Lane Change and Robust Safety

Background and Contribution

Inappropriate lane changes are responsible for one-tenth of all accidents in the U.S. [1], due to human drivers' inaccurate estimation and prediction of the surrounding traffic, illegal maneuver, and inefficient driving skill. Automated lane-changing is regarded as a solution to reduce these human errors. Recently, the rapid development of computing, communication, and sensing technologies advances automated lane-changing and prompts the development of safer and more reliable lane-changing methods. Traditionally, the automated lane-changing task can be decomposed into three modules: decision-making, trajectory planning, and controller design for AV lane change maneuver [2]. The decision-making module determines whether to execute or abort the lane change according to the safety constraints, which are obtained by using the state (position, velocity, acceleration) information of the surrounding vehicles through V2V communication and sensing. The trajectory planning module generates the feasible trajectory for lane-changing, which will be tracked by the control module. There are several challenges to automated lane-changing. Firstly, the complex interactions between the AV and the surrounding vehicles and environment make it hard to guarantee safety during lane-changing. Secondly, the rapid velocity of the vehicles requires that the lane-changing algorithm should quickly respond to the driving conditions in real-time. Thirdly, an accurate dynamic model of the AV and its surrounding environment is hard to get. Fourthly, the lane change maneuvers require both longitudinal and lateral controller design. Many studies in the literature have studied the problem of longitudinal control of AV, some recent studies are done by [3,4,5]. However, it remains challenging to precisely control the lateral movement of the AV, especially in the absence of an accurate model [6].

Over the past few years, many decision-making and trajectory-planning methodologies are proposed in the literature. Authors in [7] proposed a utility function-based lane change and merge technique. The utility function considers the discretionary, anticipatory, and mandatory conditions to judge the desirability of the AV to change lanes. Once lane change is deemed desirable, a safe longitudinal and lateral safety corridor are determined to perform the maneuver. This methodology requires tuning of parameters for the utility functions that can affect lane change decision-making. Authors in [8] proposed a real-time dynamic cooperative lane-changing model for CAVs with possible accelerations of a preceding vehicle. The lane change decision is based on the upper and lower bounds of acceleration of the preceding and following

vehicles in the target lane. The lane change trajectory is then generated using a cubic polynomial. Here, the acceleration bounds are derived using a simple kinematic model of the AV that might not be accurate. Authors in [9] and [10] proposed a constrained optimization-based lane-changing methodology. The objective function is minimized for the longitudinal and lateral jerks, and the total distance of lane change with safety constraints defined as minimum safety spacing (MSS). Computing the MSS model is complex in the formulation given by [9], and requires the knowledge of the dimension of the surrounding vehicles in the formulation given by [10] which might be difficult to estimate for the vehicles that are behind AV. A more practical scenario is considered by [11] and [12] where both human-driven vehicles and CAVs interact for lane change maneuvers where the safety distances are computed using Gipps's safe distance and intelligent driver model. Authors in [13] proposed a cooperative lane-changing methodology where a decentralized cooperative lane-changing decision-making framework for CAV is composed of state prediction, candidate decision generation, and coordination with surrounding vehicles.

Once the lane change decision-making is completed, the next task is to move the AV to the desired position/gap in the desired lane. Many control techniques have been proposed in the past to maneuver the AV to the desired lane while ensuring safety, see [7]-[14]. In [12], the authors have implemented a model predictive control (MPC) based trajectory-tracking controller. Authors in [7] used quadratic programming (QP) to compute the control signals for lateral and longitudinal maneuvers, where a double integrator model is used for the AV dynamics. Authors in [11] and [12] used MPC-based methods for the vehicle control which uses the two-wheel kinematic vehicle model. Here, the two front (or rear) wheels are considered as one wheel. Authors in [9] proposed a trajectory-tracking controller based on sliding mode control. The tracking controller is based on the backstepping approach. Authors in [15] used a hierarchical, two-level architecture for the trajectory generation and vehicle control of AV. The high-level planner uses a simplified point-mass model and linear collision avoidance constraints, whereas, the low-level controller uses a non-linear vehicle model to compute the vehicle control inputs required to execute the planned maneuvers. Both the planners are formulated based on MPC. Authors in [16] formulated a stochastic MPC controller. The MPC controller can predict future states and implement constraints directly into the control algorithm. The proposed algorithm uses a linear parameter-varying (LPV) vehicle model.

As evident from the literature, most of the works done to solve the lane changing problem of AVs used model-based techniques. One major limitation of these model-based approaches is that the

performance of the automated lane-changing algorithm highly depends on the accuracy of the AVs' model, and the inaccurate model may deteriorate the lane-changing performance. Many of the methodologies mentioned above require solving an optimization problem in real-time to generate/track safe trajectories for the AV lane change maneuver which requires high computation effort. Because of these observations, we believe that learning-based optimal control is more desirable for practical implementation, which can continually handle any model uncertainty introduced by the unknown dynamical parameters and simultaneously optimize the performance of the AV lane-changing maneuver by learning from the real-time data. Also, we propose a lane-changing algorithm that does not require parameter tuning, solving an optimization problem, or vehicle dimension information.

This paper adopts ideas from reinforcement learning and adaptive dynamic programming (ADP) ([17]-[20]) to develop an intelligent and safe lane change maneuver algorithm for AVs in the mixed traffic scenario. One major advantage of ADP, as opposed to traditional reinforcement learning ([17]), lies in the fact that the closed-loop stability of the dynamical system is established when the learned control policy is implemented. Noting the facts that maintaining a constant velocity during lane change is impractical, and the lateral dynamics of AV depend on the longitudinal velocity, we assume an linear parameter-varying (LPV) AV dynamics instead of an linear time-invariant (LTI) dynamic AV model. Thus, one cannot directly apply the LTI data-driven controller techniques developed by [21], [18]. In this work, we aim at extending the results of [18] for LPV systems. Many authors in the literature have proposed control techniques for LPV systems, some of them are summarized in [22,23]. Among other methods, gain scheduling is more suitable for the control of LPV systems if the time-varying parameter varies slowly. The authors in [22,25] were the first to introduce the systematic design and analysis of gain scheduled controllers for LPV systems. Gain scheduling has gained popularity as a control technique for complex systems like wind turbines ([26]), missile autopilot ([27, 28]), flight control ([29]), cloud computing ([30]), and more recently for AV control ([31]-[35]). However, these methods are model-based and suffer from the drawbacks mentioned before. In this work, we propose a learning-based gain scheduling technique. This work is an extension of our previous work done by [36]. The difference between the work done by [36] and the present work is that the present work provides rigorous stability analysis of the learning-based gain scheduling controller. Also, the present work studies the safety of the AV during lane change maneuvers when the proposed learning-based gain scheduling controller is used.

The term “safety-critical system” is used to classify those systems for which safety is considered as a major design problem. Control barrier functions (CBFs) have been used to ensure safety in autonomous systems [37]. A barrier function based supervisory control algorithm is proposed in [38], which works in a plug-and-play fashion. When the risk of collision is low, the barrier function is not active; when the risk is high, the barrier function controller will intervene to ensure safety. This method is applied to solve the navigation and pedestrian avoidance problem of a low-speed AV. In [39], a methodology is proposed for control of safety-critical systems, where the safety conditions expressed as control barrier functions are unified with control objectives expressed as control Lyapunov functions (CLFs). Here, given a control objective and an admissible set in the state space, a quadratic program (QP) is formulated that can mediate the tradeoff between obtaining stability and ensuring safety. Here, stability objective is used as a soft constraint while safety objective is used as a hard constraint in QP. The proposed methodology is demonstrated on adaptive cruise control and lane keeping. A similar algorithm of using a CLF-CBF-QP based control methodology is developed for lane change maneuver of autonomous vehicles in [40]. In [41], a Constrained Iterative Linear Quadratic Regulator (CILQR) is proposed for motion planning of autonomous vehicles. By writing the ILQR problem as a CILQR problem, the constraints for obstacle avoidance and reference tracking are incorporated in the objective function. It should be noted that, most of the works done in CBF formulation to ensure safety requires the information of the system models. Thus, these CBF formulations suffers from similar drawbacks of model-based control as mentioned before. In this work, we propose a data-driven formulation for safety of safety-critical systems using CBFs that do not require the knowledge of system model.

The main contributions of this work are summarized as follows:

- Introduced a learning-based optimal control design technique for lane-changing of AVs that
    - Uses only the state and input information.
    - Guarantees algorithmic convergence and vehicle stability.
- Proposed a data-driven lane-changing decision-making algorithm that incorporates the following:
    - Scales safe distances of nearby vehicles with their respective velocities. This ensures more safety for fast-moving vehicles.
    - Lane abortion in a critical scenario of non-cooperative behavior of surrounding vehicles.

- Reduced the computation time by replacing the requirement of trajectory generation with a target point defined in the target lane for AV lane change.
- Proposed a learning-based gain-scheduling controller to handle parameter-varying problems for the dynamic model of the AV. The stability of the learning-based gain-scheduling controller has been rigorously established.
- Demonstrated the applicability of the proposed methodologies in real-time learning and decision-making by SUMO implementation.
- Developed a data-driven formulation to guarantee safety during AV lane change using CBFs.
- Conducted experiments and verified the robustness of the data-driven controller in the presence of measurement noise.

Dynamic Model and Problem Formulation

A. Longitudinal Dynamic Model

The vehicle's longitudinal dynamic model is given as follows:

$$\dot{\boldsymbol{x}}_{lo} = \boldsymbol{A}_{\boldsymbol{lo}}\boldsymbol{x}_{lo} + \boldsymbol{B}_{lo}\boldsymbol{u}_{lo}, \tag{1}$$

where

$$\boldsymbol{A}_{lo} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \boldsymbol{B}_{lo} = \left[0, \frac{1}{m}\right]^T, \boldsymbol{x}_{lo} = [x_1, x_2]^T,$$

with m = mass of the vehicle, and $\boldsymbol{u}_{lo}$ is the force in the acceleration paddle, x_1 = longitudinal position (x_{AV}), and x_2 = longitudinal velocity.

B. Lateral Dynamic Model

The lateral dynamic model is based on the position and orientation error variables as shown in Fig. 1. Let (T_x, T_y) be the coordinates of the target point, ψ be the orientation of the vehicle, e_1 be the error between the distance of the center of gravity of the vehicle and the center line of the target lane, and e_2 be the orientation error of the vehicle with respect to the road.

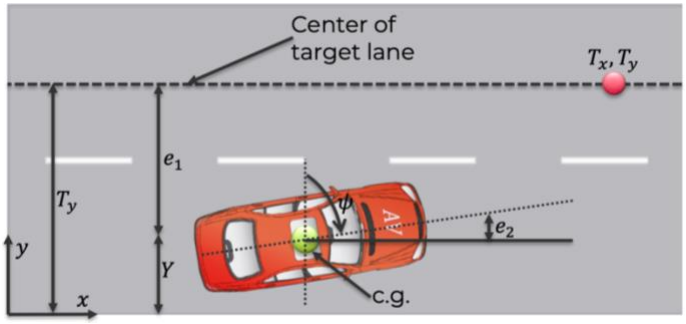


Fig. 1: Defining the errors $e_1(t)$ and $e_2(t)$

Assumption : Vehicles travel on a straight road with radius $r = \infty$.

The dynamic model is given as:

$$\dot{\boldsymbol{x}}_{la} = \boldsymbol{A}_{la}\boldsymbol{x}_{la} + \boldsymbol{B}_{la}\boldsymbol{u}_{la}, \tag{2}$$

where $\boldsymbol{u}_{la}$ is the front wheel steering angle,

$$\boldsymbol{x}_{la} = [e_1(t), \dot{e_1}(t), e_2(t), \dot{e_2}(t)]^T, \boldsymbol{B}_{la} = \left[0, \frac{2C_{\alpha f}}{m}, 0, \frac{2C_{\alpha f}l_f}{I_z}\right]^T,$$

$$\boldsymbol{A}_{la} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f}+2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f}+2C_{\alpha r}}{m} & \frac{-2C_{\alpha f}l_f+2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{-2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_zV_x} & \frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z} & \frac{-2C_{\alpha f}l_f^2+2C_{\alpha r}l_r^2}{I_zV_x} \end{bmatrix}.$$

To keep the development of the model-free learning-based controller simple, in this report, we assume that the longitudinal and lateral dynamics are linear. It can be seen from $\boldsymbol{A}_{la}$ that the longitudinal velocity V_x appears non-linearly. From Fig. 1, the lateral position $Y(t)$ of AV and $\psi(t)$ can be obtained as:

$$Y(t) = T_y - e_1(t),$$
$$\psi(t) = \psi_{des} - e_2(t). \tag{3}$$

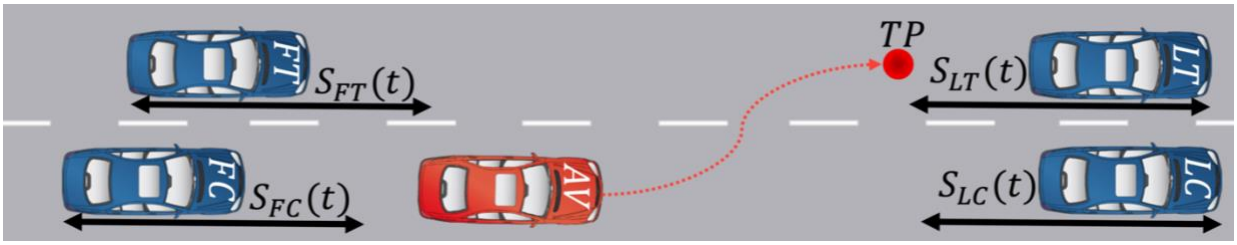


Fig. 2: A typical lane change Scenario

C. Lane Change Decision Making

In Fig. 2, AV denotes the autonomous vehicle, LC denotes the lead vehicle in the current lane, FC denotes the following vehicle in the current lane, LT denotes the lead vehicle in the target lane, FT denotes the following vehicle in the target lane, $S_i(t) = L + hv_i(t), i \in \{LT, FT, LC, FC\}$, h = headway time, L = length of vehicle, and v_i = velocity of the i^{th} vehicle, TP is the target point. In this work, the lane change decision making is introduced for a single lane change maneuver. As shown in Fig. 2, four vehicles are involved in a lane change maneuver. The AV performs a maneuver to change the lane and places itself in the target point. Let $x_{AV}, x_{LT}, x_{FT}, x_{LC}, x_{FC}$ be the longitudinal positions of the vehicles involved in the lane changing process. Then, the following conditions must hold true for a safe lane change:

$$x_{\mathrm{AV}} \leq x_{LC} - S_{LC}(t),$$
$$x_{\mathrm{AV}} \geq x_{FC} + S_{FC}(t),$$
$$x_{\mathrm{AV}} \leq x_{LT} - S_{LT}(t), \tag{4}$$

$$x_{\mathrm{AV}} \geq x_{FT} + S_{FT}(t),$$

The safe distances $S_i(t)$ are evaluated continuously. If the above inequalities are violated at any time instant during the lane changing, the AV maneuvers back to the original lane. This maneuver is done based on the change of leader vehicle. Thus, when the safe conditions violate, the target point TP is chosen at a safe distance from the leader in the original lane. The complete lane change algorithm is presented in the flowchart below:

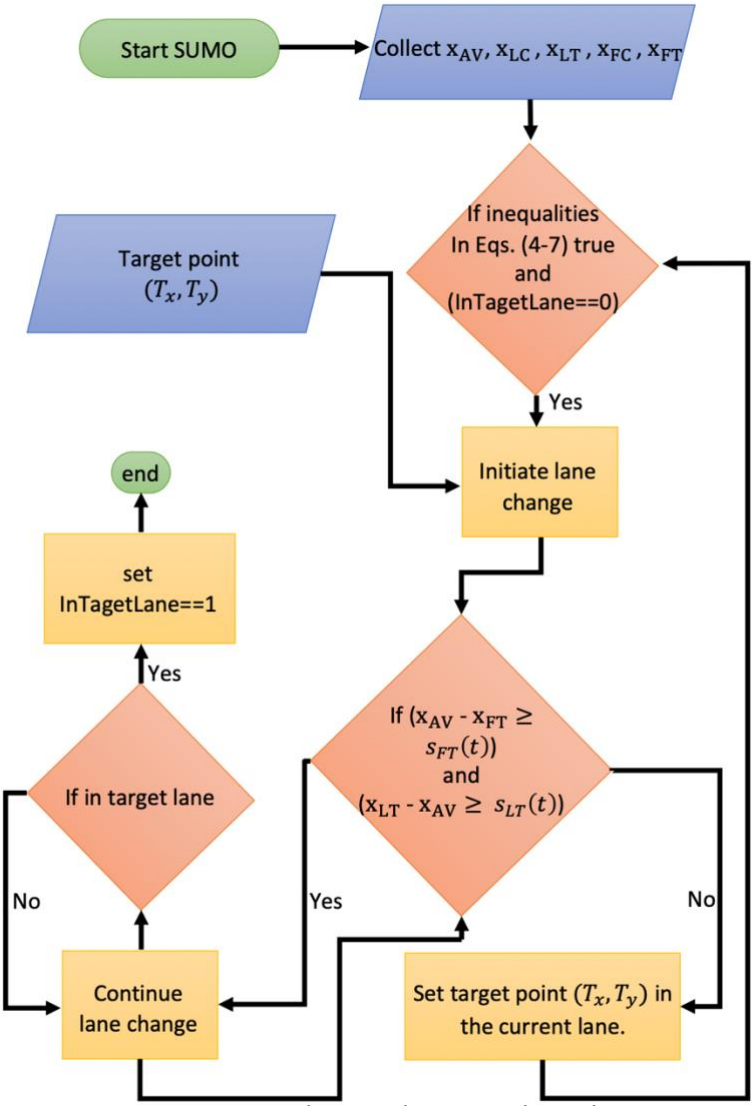


Fig. 3: Lane change decision algorithm.

D. Problem Definition

Given that the dynamics of the AV is unknown and TP in the target lane is defined, the following problem is addressed in this work:

Problem: Design a lane changing algorithm for the AV that incorporates the following:

(i) an optimal model-free controller for the AV 's lateral maneuver such that $e_1 \to 0$, $e_2 \to 0$.
(ii) an optimal model-free controller for the AV's longitudinal maneuver.
(iii) a lane change decision making mechanism based on inequalities in (4).
(iv) an optimal model-free controller for post lane change platooning.
(v) learning and scheduling optimal gains for change in longitudinal velocity.

Learning-based Gain Scheduling Algorithm

A. Model-based Learning

The gain scheduling technique designs controllers as follows: at a good number of operating points obtain the linear time-invariant approximations of the system; design linear time-invariant controllers for each linear time-invariant approximations of the system at the selected operating points that guarantee stability and certain performance objectives; link these controllers together in order to obtain a single controller for the entire range of the system operation. Consider the following LPV system:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\alpha)\boldsymbol{x} + \boldsymbol{B}(\alpha)\boldsymbol{u}, \tag{5}$$

$$\alpha = \alpha(t), \tag{6}$$

where, $\boldsymbol{x}(t) \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u}(t) \in \mathbb{R}^m$ is the input, $\boldsymbol{A}(\alpha) \in \mathbb{R}^{n\times n}$, and $\boldsymbol{B}(\alpha) \in \mathbb{R}^{n\times m}$ are the state and input matrices respectively that are considered unknown. For all $t \geq 0$ the parameter $\alpha = \alpha(t) \in [\alpha_0, \alpha_n] =: I \subset R$. For what follows, we make the following assumptions:

Assumption 1: For all $\alpha \in I$, the matrix $\boldsymbol{B}(\alpha)$ is full column rank.

Assumption 2: The elements of the system matrices $\boldsymbol{A}(\alpha)$, and $\boldsymbol{B}(\alpha)$ are analytic functions of α.

Assumption 3: The parameter α is a continuous and bounded function of time t, differentiable almost everywhere with bounded derivative and is measured for all time $t > 0$.

Assumption 4: All states are available for feedback, and the system in (5) is stabilizable for all $\alpha \in I$.

Without the knowledge of $\boldsymbol{A}(\alpha)$ and $\boldsymbol{B}(\alpha)$, we seek to design a linear optimal control law of the form:

$$\boldsymbol{u} = -\boldsymbol{K}(\alpha)\boldsymbol{x}, \tag{7}$$

where, $\boldsymbol{K}(\alpha) \in \mathbb{R}^{m\times n}$ is the state feedback gain matrix. To design the state feedback control law in (7), we first select finite number of fixed $\alpha_l \in I$. Let $\boldsymbol{K}(\alpha_l)$ and $\boldsymbol{K}(\alpha_{l+1})$, respectively, denote the gain matrices computed at the adjacent points α_l and α_{l+1}. At each $\alpha \in [\alpha_l, \alpha_{l+1}]$, the gain $\boldsymbol{K}(\alpha)$ in (7) is obtained as the linear interpolation between $\boldsymbol{K}(\alpha_l)$ and $\boldsymbol{K}(\alpha_{l+1})$ given as:

$$\boldsymbol{K}(\alpha) = \boldsymbol{K}(\alpha_l) + \frac{\boldsymbol{K}(\alpha_{l+1}) - \boldsymbol{K}(\alpha_l)}{\alpha_{l+1} - \alpha_l}(\alpha - \alpha_l). \tag{8}$$

The gain matrices are computed such that the following are satisfied:

1. For each α_l the state feedback gain matrix $\boldsymbol{K}(\alpha_l)$ is computed such that the closed-loop stability of the frozen system $\boldsymbol{A}_c(\alpha_l) = \boldsymbol{A}(\alpha_l) - \boldsymbol{B}(\alpha_l)\boldsymbol{K}(\alpha_l)$ along with a minimum cost of operating the system is guaranteed.
2. At each $\alpha \in [\alpha_l, \alpha_{l+1}]$, the gain $\boldsymbol{K}(\alpha)$ obtained using (8) guarantees the stability of the closed-loop system.

In order to reduce the state deviations and control effort, we seek to design a linear optimal control law of the form given in (7) for a fixed $\alpha_l \in I$ that can minimize the following cost function:

$$\min_{\boldsymbol{u}} \; J = \int_0^\infty (\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}^T\boldsymbol{R}\boldsymbol{u})\, dt, \tag{9}$$

where, $\boldsymbol{Q} = \boldsymbol{Q}^T \geq \boldsymbol{0}$, $\boldsymbol{R} = \boldsymbol{R}^T > \boldsymbol{0}$, and $\left(\boldsymbol{A}(\alpha_l), \boldsymbol{Q}^{1/2}\right)$ is observable.

If $\boldsymbol{A}(\alpha_l)$, $\boldsymbol{B}(\alpha_l)$ are completely known, the solution to the above-mentioned problem is well known and the optimal gain matrix $\boldsymbol{K}^* \in \mathbb{R}^{m\times n}$ can be found as follows:

$$\boldsymbol{A}(\alpha_l)^T\boldsymbol{P}(\alpha_l) + \boldsymbol{P}(\alpha_l)\boldsymbol{A}(\alpha_l) + \boldsymbol{Q} - \boldsymbol{P}(\alpha_l)\boldsymbol{B}(\alpha_l)\boldsymbol{R}^{-1}\boldsymbol{B}(\alpha_l)^T\boldsymbol{P}(\alpha_l) = \boldsymbol{0},$$

$$\boldsymbol{K}(\alpha_l)^* = \boldsymbol{R}^{-1}\boldsymbol{B}(\alpha_l)^T\boldsymbol{P}(\alpha_l)^*,$$

(10)

(11)

where (10) is called the algebraic Riccati equation and $\boldsymbol{P}(\alpha_l)^* = \boldsymbol{P}(\alpha_l)^{*T} \geq 0$ is the unique solution of (10). Since the Riccati equation is non-linear in $\boldsymbol{P}(\alpha_l)$, it is generally difficult to solve. In the literature, many efficient iterative approaches have been proposed to solve (10). One such approach is given in [42], which is reproduced for completeness:

Theorem 1: If $\boldsymbol{K}_0(\alpha_l)$ is any stabilizing control gain, $\boldsymbol{A}_k(\alpha_l) = \boldsymbol{A}(\alpha_l) - \boldsymbol{B}(\alpha_l)\boldsymbol{K}_k(\alpha_l)$ and $\boldsymbol{P}_k(\alpha_l)$ is the symmetric positive definite solution of the Lyapunov equation:

$$\boldsymbol{A}_k(\alpha_l)^{\boldsymbol{T}}\boldsymbol{P}_k(\alpha_l) + \boldsymbol{P}_k(\alpha_l)\boldsymbol{A}_k(\alpha_l) + \boldsymbol{Q} + \boldsymbol{K}_k^{\boldsymbol{T}}(\alpha_l)\boldsymbol{R}\boldsymbol{K}_k(\alpha_l) = \boldsymbol{0}, \tag{12}$$

$$\boldsymbol{K}_{k+1}(\alpha_l) = \boldsymbol{R}^{-\boldsymbol{1}}\boldsymbol{B}(\alpha_l)^{\boldsymbol{T}}\boldsymbol{P}_k(\alpha_l). \tag{13}$$

Then, the following conditions hold:

- $\boldsymbol{A}(\alpha_l) - \boldsymbol{B}(\alpha_l)\boldsymbol{K}_k(\alpha_l)$ is Hurwitz,
- $\boldsymbol{P}^*(\alpha_l) \leq \boldsymbol{P}_{k+1}(\alpha_l) \leq \boldsymbol{P}_k(\alpha_l)$,
- $\lim_{k\to\infty} \boldsymbol{K}_k(\alpha_l) = \boldsymbol{K}^*(\alpha_l)$, $\lim_{k\to\infty} \boldsymbol{P}_k(\alpha_l) = \boldsymbol{P}^*(\alpha_l)$.

Note that (12) is linear in $\boldsymbol{P}_k(\alpha_l)$. Thus, one can iteratively solve (12) and update $\boldsymbol{K}_k$ to numerically approximate the solution. But this assumes the complete knowledge of the system matrices $\boldsymbol{A}(\alpha_l)$, $\boldsymbol{B}(\alpha_l)$.

B. Model-free Learning

Here, we present an online model-free learning-based controller design strategy that does not assume any knowledge of the system matrices $\boldsymbol{A}(\alpha_l)$, $\boldsymbol{B}(\alpha_l)$. Consider the modified system equation as follows:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_k(\alpha_l)\boldsymbol{x} + \boldsymbol{B}(\alpha_l)(\boldsymbol{K}_k(\alpha_l)\boldsymbol{x} + \boldsymbol{u}), \tag{14}$$

where, $\boldsymbol{A}_k(\alpha_l) = \boldsymbol{A}(\alpha_l) - \boldsymbol{B}(\alpha_l)\boldsymbol{K}_k(\alpha_l)$. Then, using (12), (13), and (14), we have:

$$\boldsymbol{x}(t+\delta t)^T\boldsymbol{P}_k(\alpha_l)\boldsymbol{x}(t+\delta t) - \boldsymbol{x}(t)^T\boldsymbol{P}_k(\alpha_l)\boldsymbol{x}(t)$$

$$= \int_t^{t+\delta t}\left[\boldsymbol{x}^T\left(\boldsymbol{A}_k^T(\alpha_l)\boldsymbol{P}_k(\alpha_l) + \boldsymbol{P}_k(\alpha_l)\boldsymbol{A}_k(\alpha_l)\right)\boldsymbol{x} + 2(\boldsymbol{u} + \boldsymbol{K}_k(\alpha_l)\boldsymbol{x})^T\boldsymbol{B}^{\boldsymbol{T}}(\alpha_l)\boldsymbol{P}_k(\alpha_l)\boldsymbol{x}\right]d\tau, \tag{15}$$

$$= 2\int_t^{t+\delta t}(\boldsymbol{u} + \boldsymbol{K}_k(\alpha_l)\boldsymbol{x})^T\boldsymbol{R}\boldsymbol{K}_{k+1}(\alpha_l)\boldsymbol{x}\,d\tau - \int_t^{t+\delta t}\boldsymbol{x}^T\boldsymbol{Q}_k\boldsymbol{x}\,d\tau,$$

where, $\boldsymbol{Q}_k = \boldsymbol{Q} + \boldsymbol{K}_k^{\boldsymbol{T}}(\alpha_l)\boldsymbol{R}\boldsymbol{K}_k(\alpha_l)$. It must be noted that (15) is independent of the systems matrices $\boldsymbol{A}(\alpha_l), \boldsymbol{B}(\alpha_l)$.

Define the following:

$$\boldsymbol{x}^T\boldsymbol{Q}_k\boldsymbol{x} = (\boldsymbol{x}^T \otimes \boldsymbol{x}^T)\mathrm{vec}(\boldsymbol{Q}_k), \tag{16}$$

$$(\boldsymbol{u} + \boldsymbol{K}_k(\alpha_l)\boldsymbol{x})^T\boldsymbol{R}\boldsymbol{K}_{k+1}(\alpha_l)\boldsymbol{x} = [(\boldsymbol{x}^T \otimes \boldsymbol{x}^T)(\boldsymbol{I}_n \otimes \boldsymbol{K}^{\boldsymbol{T}}{}_k(\alpha_l)\boldsymbol{R}) + (\boldsymbol{x}^T \otimes \boldsymbol{u}^T)(\boldsymbol{I}_n \otimes \boldsymbol{R})]\mathbf{vec}\big(\boldsymbol{K}_{k+1}(\alpha_l)\big). \tag{17}$$

For any positive integer l, define: $\boldsymbol{\Delta_{xx}} \in \mathbb{R}^{l\times(1/2)n(n+1)}$, $\boldsymbol{I_{xx}} \in \mathbb{R}^{l\times n^2}$, $\boldsymbol{I_{xu}} \in \mathbb{R}^{l\times nm}$ as follows for $0 \le t_1 < t_2 < \cdots < t_l$:

$$\boldsymbol{\Delta_{xx}} = [\bar{\boldsymbol{x}}(\boldsymbol{t_1}) - \bar{\boldsymbol{x}}(\boldsymbol{t_0}), \cdots, \bar{\boldsymbol{x}}(\boldsymbol{t_l}) - \bar{\boldsymbol{x}}(\boldsymbol{t_{l-1}})]^T, \tag{18}$$

$$\boldsymbol{I_{xx}} = \left[\int_{t_1}^{t_2} \boldsymbol{x}\otimes\boldsymbol{x}\boldsymbol{d\tau}, \int_{t_2}^{t_3} \boldsymbol{x}\otimes\boldsymbol{x}\boldsymbol{d\tau}, \cdots, \int_{t_{l-1}}^{t_l} \boldsymbol{x}\otimes\boldsymbol{x}\boldsymbol{d\tau}\right]^T, \tag{19}$$

$$\boldsymbol{I_{xu}} = \left[\int_{t_1}^{t_2} \boldsymbol{x}\otimes\boldsymbol{u}\boldsymbol{d\tau}, \int_{t_2}^{t_3} \boldsymbol{x}\otimes\boldsymbol{u}\boldsymbol{d\tau}, \cdots, \int_{\boldsymbol{t_{l-1}}}^{t_l} \boldsymbol{x}\otimes\boldsymbol{u}\boldsymbol{d\tau}\right]^{\boldsymbol{T}}, \tag{20}$$

where $\bar{\boldsymbol{x}} = \left[\boldsymbol{x_1^2}, \boldsymbol{x_1x_2}, \ldots, \boldsymbol{x_1x_n}, \boldsymbol{x_2^2}, \boldsymbol{x_2x_3}, \ldots, \boldsymbol{x_{n-1}x_n}, \boldsymbol{x_n^2}\right]^{\boldsymbol{T}}$. Using (16)-(20), (15) can be written as:

$$\boldsymbol{\Gamma}_k\begin{bmatrix}\widehat{\boldsymbol{p}}\\ \mathbf{vec}\big(\boldsymbol{K}_{k+1}(\alpha_l)\big)\end{bmatrix} = \boldsymbol{\Psi}_k, \tag{19}$$

where,

$$\boldsymbol{\Gamma}_k = \left[\boldsymbol{\Delta_{xx}}, -\boldsymbol{2I_{xx}}\big(\boldsymbol{I_n} \otimes \boldsymbol{K}_k^{\boldsymbol{T}}(\alpha_l)\boldsymbol{R}\big) - \boldsymbol{2I_{xu}}(\boldsymbol{I_n} \otimes \boldsymbol{R})\right], \tag{20}$$

$$\boldsymbol{\Psi}_k = -\boldsymbol{I_{xx}}\mathbf{vec}(\boldsymbol{Q}_k), \tag{21}$$

$$\widehat{\boldsymbol{p}} = \left[p_{11}, 2p_{12}, \ldots, 2p_{1n}, p_{22}, 2p_{23}, \ldots, 2p_{(n-1)n}, p_{nn}\right]^T.$$

Thus, given an initial stabilizing control input, the trajectories of the system can be recorded online in (18)-(20), which can then be recorded in the data matrices (20), (21). The learning-based control algorithm is presented in Fig. (4).

***Assumption 5**:* There exists a sufficiently large integer $l > 0$, such that:

$$\mathrm{rank}([\boldsymbol{I_{xx}}, \boldsymbol{I_{xu}}]) = \frac{n(n+1)}{2} + mn.$$

(22)

Theorem 2[18,19]*:* Under the assumption (22), there is a unique pair of matrices $\boldsymbol{P}_k(\alpha_l), \boldsymbol{K}_{k+1}(\alpha_l)$, with $\boldsymbol{P}_k(\alpha_l) = \boldsymbol{P}_k^{\boldsymbol{T}}(\alpha_l)$, $\forall k \in \mathbb{Z}_+$, such that:

$$\boldsymbol{\Gamma}_k \begin{bmatrix} \widehat{\boldsymbol{p}} \\ \mathbf{vec}\big(\boldsymbol{K}_{k+1}(\alpha_l)\big) \end{bmatrix} = \boldsymbol{\Psi}_k. \tag{23}$$

Theorem 3 [18,19]*:* Given an initial stabilizing gain $\boldsymbol{K}_0(\alpha_l)$ if (22) holds, the sequence $\{\boldsymbol{P}_i(\alpha_l)\}_{i=0}^{\infty}$ and $\{\boldsymbol{K}_i(\alpha_l)\}_0^{\infty}$ obtained by solving (23) converge to the optimal values $\boldsymbol{P}^*(\alpha_l)$ and $\boldsymbol{K}^*(\alpha_l)$, respectively.

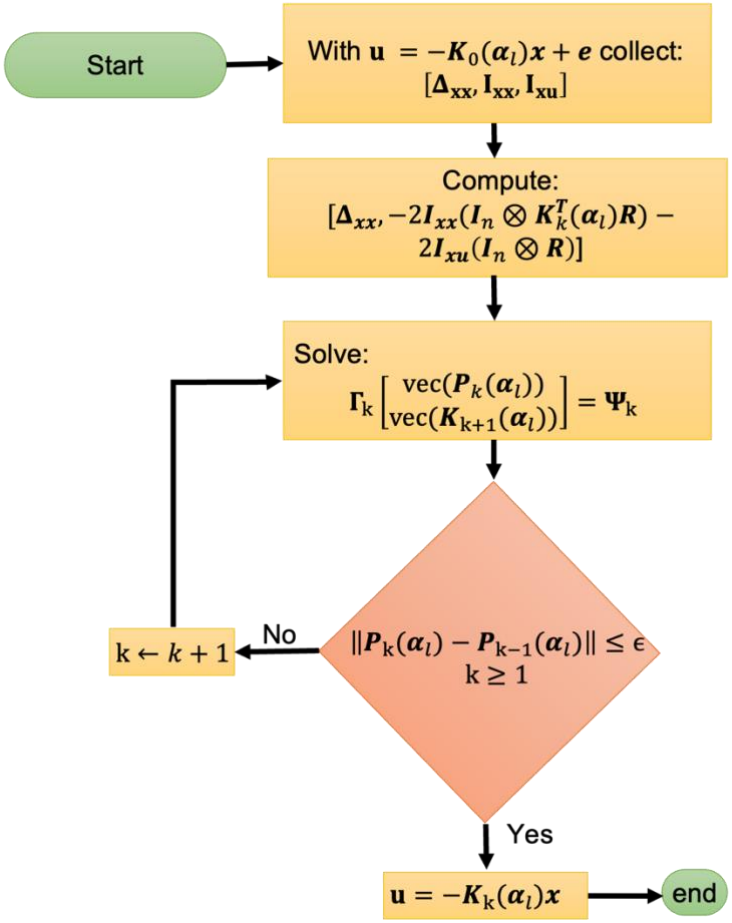


Fig. 4: Learning-based control algorithm.

Thus, we have established a learning-based optimal control framework for fixed $\alpha_l \in I$. The optimality and convergence guarantee of the optimal learning-based controller for a fixed $\alpha_l \in I$ are given by Theorem 2 and Theorem 3. It remains to show that that closed loop system $\boldsymbol{A}_c(\alpha) =$

$\boldsymbol{A}(\alpha) - \boldsymbol{B}(\alpha)\boldsymbol{K}(\alpha)$ is stable for the control gain $\boldsymbol{K}(\alpha)$ obtained using (8) for any fixed $\alpha \in [\alpha_l, \alpha_{l+1}]$. This is discussed in the next subsection.

C. Stability analysis of LPV systems with learning-based gain scheduling controller

Let $\sigma(\boldsymbol{A}_c(\alpha_l))$ be the spectrum of the closed-loop system with the feedback gain $\boldsymbol{K}(\alpha_l)$. Since, we can design an optimal learning-based controller for a fixed $\alpha_l \in I$, we call the spectrum of $\sigma(\boldsymbol{A}_c(\alpha_l))$ optimal, where $\boldsymbol{A}_c(\alpha_l) = \boldsymbol{A}(\alpha_l) - \boldsymbol{B}(\alpha_l)\boldsymbol{K}^*(\alpha_l)$. Suppose $\lambda_o^j \in \sigma(\boldsymbol{A}_c(\alpha_l))$ is an eigenvalue in the optimal spectrum $\sigma(\boldsymbol{A}_c(\alpha_l))$, and $\lambda^j \in \sigma(\boldsymbol{A}_c(\alpha))$ be an eigenvalue in the spectrum $\sigma(\boldsymbol{A}_c(\alpha))$, where, $j = 1,2,\cdots,n$. In this subsection, we discuss how close α_l and α_{l+1} must be such that for each fixed $\alpha \in [\alpha_l, \alpha_{l+1}]$, λ^j is in a small neighbourhood $\mathcal{N}_j(\epsilon)$ of $\sigma(\boldsymbol{A}_c(\alpha_l))$, i.e, $\lambda^j \in \mathcal{N}_j(\epsilon) \coloneqq \{s \in C_- : |s - \lambda_o^j| < \epsilon < 1\}$. The results in this section are based on the theory of eigenvalue perturbation.

Lemma 4: Let $\boldsymbol{S} \in M_n$ be a diagonalizable matrix where M_n is the set of all square matrices. Let, $\boldsymbol{S} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$, where $\boldsymbol{V}$ is nonsingular and $\boldsymbol{\Lambda}$ is diagonal. Let $\boldsymbol{\delta S} \in \boldsymbol{M}_n$ and let $||.||$ be a matrix norm on $\boldsymbol{M}_n$ that is induced by an absolute norm on C^n. If $\lambda_{\bar{S}}$ be an eigenvalue of $\bar{\boldsymbol{S}} = \boldsymbol{S} + \boldsymbol{\delta S}$, there is an eigenvalue of $\lambda_{\boldsymbol{S}}$ of $\boldsymbol{S}$ such that:

$$|\lambda_{\bar{S}} - \lambda_{\boldsymbol{S}}| \leq \kappa(\boldsymbol{V})||\boldsymbol{\delta S}||, \tag{24}$$

where $\kappa(\boldsymbol{V}) = ||\boldsymbol{V}||||\boldsymbol{V}^{-1}||$ is the condition number with respect to the matrix norm $||.||$.

Remark: If $\boldsymbol{S}$ is a symmetric matrix and $\boldsymbol{V}$ is the eigenvector matrix of $\boldsymbol{S}$. If the induced norm is an l_2 norm, we have $\kappa(\boldsymbol{V}) = 1$.

Lemma 5: Let $F(z, w)$ be a function of two variables which is analytic in the neighbourhood of the point (z_0, w_0), and suppose the following conditions hold:

1. $F(z_0, w_0) = 0$, (24)
2. $\frac{\partial F(z,w)}{\partial w}|_{(z_0,w_0)} \neq 0$. (25)

Then there are neighborhoods $\mathcal{N}(z_0)$ and $\mathcal{N}(w_0)$ such that the equation $F(z, w) = 0$ has a unique root $w = w(z)$ in $N(w_0)$ for any given $z \in N(z_0)$. Moreover, the function $w(z)$ is single-valued and analytic in $N(z_0)$, and satisfies the condition $w(z_0) = w_0$.

The result of the following theorem is crucial to show $\lambda^j \in \mathcal{N}_j(\epsilon) \coloneqq \{s \in C_-^{\mathbb{0}} : |s - \lambda_o^j| < \epsilon < 1\}$ for a sufficiently small ϵ.

Theorem 6: Let ϵ denote the length of the gain-scheduling interval. Then for any $\alpha \in [\alpha_l, \alpha_{l+1}]$, we have $\alpha = \alpha_l + \epsilon c$, where $0 \leq c \leq 1$. Then, under Assumptions 2 and 4, there exist a sufficient small ϵ, such that the following relations hold.

$$\boldsymbol{P}^*(\alpha) = \boldsymbol{P}^*(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon), \tag{26}$$

$$\boldsymbol{K}^*(\alpha) = \boldsymbol{K}^*(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon). \tag{27}$$

Proof: Since the elements of $\boldsymbol{A}(\alpha)$, and $\boldsymbol{B}(\alpha)$ are analytic functions of α, for small ϵ, we have $\boldsymbol{A}(\alpha) = \boldsymbol{A}(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon)$, and $\boldsymbol{B}(\alpha) = \boldsymbol{B}(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon)$. Thus, for any α the Lyapunov equation can be written as the following:

$$\boldsymbol{A}_k^T(\alpha)\boldsymbol{P}_k(\alpha) + \boldsymbol{P}_k(\alpha)\boldsymbol{A}_k(\alpha) + \boldsymbol{Q} + \boldsymbol{K}_k^T(\alpha)\boldsymbol{R}\boldsymbol{K}_k(\alpha) = 0, \tag{28}$$

where, $\boldsymbol{A}_k(\alpha) = \boldsymbol{A}(\alpha) - \boldsymbol{B}(\alpha)\boldsymbol{K}_k(\alpha)$. Using (8), one can obtain $\boldsymbol{A}_{\boldsymbol{k}}(\alpha) = \boldsymbol{A}_{\boldsymbol{k}}(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon)$, where $\boldsymbol{A}_{\boldsymbol{k}}(\alpha_l) = \boldsymbol{A}(\alpha_l) - \boldsymbol{B}(\alpha_l)\boldsymbol{K}_{\boldsymbol{k}}(\alpha_l)$. As $\boldsymbol{B}(\alpha)$ is analytic in α, let, $\boldsymbol{K}_{k+1}\big(\epsilon, \boldsymbol{P}_k(\alpha)\big) \coloneqq \boldsymbol{K}_{k+1}(\alpha) = \boldsymbol{R}^{-1}\boldsymbol{B}(\alpha_l)^T\boldsymbol{P}_k(\alpha) + \boldsymbol{\mathcal{O}}(\epsilon)$. Next, define the following function:

$$\begin{aligned}\boldsymbol{F}_k\big(\epsilon, \boldsymbol{P}_k(\alpha)\big) = \big(\boldsymbol{A}_k(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon)\big)^T\boldsymbol{P}_k(\alpha) + \boldsymbol{P}_k(\alpha)\big(\boldsymbol{A}_k(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon)\big) + \boldsymbol{Q} + \\ \boldsymbol{K}_k^T\big(\epsilon, \boldsymbol{P}_{k-1}(\alpha)\big)\boldsymbol{R}\boldsymbol{K}_k\big(\epsilon, \boldsymbol{P}_{k-1}(\alpha)\big).\end{aligned} \tag{29}$$

Let $k = 1$ and note that at the point $\big(\epsilon = 0, \boldsymbol{P}_1(\alpha_l)\big)$, we have $\alpha = \alpha_l$ and the following:

$$\boldsymbol{F}_1\big(0, \boldsymbol{P}_1(\alpha_l)\big) = \boldsymbol{A}_1(\alpha_l)^T\boldsymbol{P}_1(\alpha_l) + \boldsymbol{P}_1(\alpha_l)\boldsymbol{A}_1(\alpha_l) + \boldsymbol{Q} + \boldsymbol{K}_1^T\boldsymbol{R}\boldsymbol{K}_{\boldsymbol{1}}. \tag{30}$$

Note that $\boldsymbol{K}_1$ is the known initial stabilizing controller gain that is used to start the iteration for the model-free learning. Thus, $F\big(0, \boldsymbol{P}_1(\alpha_l)\big) = 0$ has a unique solution $\boldsymbol{P}_1(\alpha_l)$ as $\boldsymbol{A}_1(\alpha_l)$ is Hurwitz. Also, we have that:

$$\boldsymbol{K}_2\big(0, \boldsymbol{P}_1(\alpha_l)\big) = \boldsymbol{R}^{-1}\boldsymbol{B}(\alpha_l)^T\boldsymbol{P}_{\boldsymbol{1}}(\alpha_l). \tag{31}$$

Now,

$$\boldsymbol{F}_1\big(\epsilon, \boldsymbol{P}_1(\alpha)\big) = \big(\boldsymbol{A}_1(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon)\big)^T\boldsymbol{P}_1(\alpha) + \boldsymbol{P}_1(\alpha)\big(\boldsymbol{A}_1(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon)\big) + \boldsymbol{Q} + \boldsymbol{K}_1^T\boldsymbol{R}\boldsymbol{K}_{\boldsymbol{1}}. \tag{32}$$

Using Lemma 5 for (32) and taking the derivative with respect to $\mathrm{vec}\big(\boldsymbol{P}_1(\alpha)\big)$ at the point $\big(\epsilon = 0, \boldsymbol{P}_1(\alpha_l)\big)$, we have the following:

$$\frac{\partial \mathrm{vec}\left(F_1\big(\epsilon, \boldsymbol{P}_1(\alpha)\big)\right)}{\partial \mathrm{vec}\big(\boldsymbol{P}_1(\alpha)\big)}\Big|_{(0,\boldsymbol{P}_1(\alpha_l))} = \boldsymbol{I} \otimes \boldsymbol{A}_1(\alpha_l)^T + \boldsymbol{A}_1(\alpha_l)^T \otimes \boldsymbol{I}. \tag{33}$$

Since $A_1(\alpha_l)$ Hurwitz, all its eigenvalues have strictly negative real parts. Therefore, $\det(\boldsymbol{I} \otimes \boldsymbol{A}_1(\alpha_l)^T + \boldsymbol{A}_1(\alpha_l)^T \otimes \boldsymbol{I}) \neq 0$. Thus, for a small ϵ by using the implicit function theorem there exists an unique solution for $\boldsymbol{P}_1(\alpha)$ with $F_1\big(\epsilon, \boldsymbol{P}_1(\alpha)\big) = 0$ that is analytic in ϵ. Hence we have the following:

$$\boldsymbol{P}_1(\alpha) = \boldsymbol{P}_1(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon), \tag{34}$$

$$\boldsymbol{K}_2\big(\epsilon, \boldsymbol{P}_1(\alpha)\big) = \boldsymbol{K}_2\big(0, \boldsymbol{P}_1(\alpha_l)\big) + \boldsymbol{\mathcal{O}}(\epsilon), \tag{35}$$

where $\boldsymbol{K}_2\big(0, \boldsymbol{P}_1(\alpha_l)\big) = \boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}}(\alpha_l)\boldsymbol{P}_1(\alpha_l)$. For $k = 2$ at the point $\big(\epsilon = 0, \boldsymbol{P}_2(\alpha_l)\big)$, we have the following:

$$\boldsymbol{F}_2\big(0, \boldsymbol{P}_2(\alpha_l)\big) = \boldsymbol{A}_2(\alpha_l)^T\boldsymbol{P}_2(\alpha_l) + \boldsymbol{P}_2(\alpha_l)\boldsymbol{A}_2(\alpha_l) + \boldsymbol{Q} + \boldsymbol{K}_2\big(0, \boldsymbol{P}_1(\alpha_l)\big)^T\boldsymbol{R}\boldsymbol{K}_2\big(0, \boldsymbol{P}_1(\alpha_l)\big). \tag{36}$$

Then, $\boldsymbol{F}_2\big(0, \boldsymbol{P}_2(\alpha_l)\big) = 0$ has a unique solution $\boldsymbol{P}_2(\alpha_l)$. And thus,

$$\boldsymbol{K}_3\big(0, \boldsymbol{P}_2(\alpha_l)\big) = \boldsymbol{R}^{-1}\boldsymbol{B}(\alpha_l)^T\boldsymbol{P}_2(\alpha_l). \tag{37}$$

Following the similar steps for the $k = 1$ case, by using the implicit function theorem, one can obtain a unique solution for $\boldsymbol{P}_2(\alpha)$ with $F_2\big(\epsilon, \boldsymbol{P}_2(\alpha)\big) = 0$ that is analytic in ϵ. Hence, we have the following:

$$\boldsymbol{P}_2(\alpha) = \boldsymbol{P}_2(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon), \tag{38}$$

$$\boldsymbol{K}_3\big(\epsilon, \boldsymbol{P}_2(\alpha)\big) = \boldsymbol{K}_3\big(0, \boldsymbol{P}_2(\alpha_l)\big) + \boldsymbol{\mathcal{O}}(\epsilon), \tag{39}$$

where $\boldsymbol{K}_3\big(0, \boldsymbol{P}_2(\alpha_l)\big) = \boldsymbol{R}^{-1}\boldsymbol{B}^{\boldsymbol{T}}(\alpha_l)\boldsymbol{P}_2(\alpha_l)$.

From (34) and (38), we have that:

$$\boldsymbol{P}_1(\alpha) - \boldsymbol{P}_2(\alpha) = \boldsymbol{P}_1(\alpha_l) - \boldsymbol{P}_2(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon). \tag{40}$$

By Theorem 1, $\boldsymbol{S}(\alpha_l) = \boldsymbol{P}_1(\alpha_l) - \boldsymbol{P}_2(\alpha_l) \geq 0$. Note that, $\boldsymbol{S}(\alpha_l)$ is a symmetric matrix. Thus, using Lemma 4, we have that $\boldsymbol{P}_1(\alpha) - \boldsymbol{P}_2(\alpha) \geq 0$ for sufficiently small ϵ. Similarly, for $k = 3$, we have $\boldsymbol{P}_2(\alpha) - \boldsymbol{P}_3(\alpha) \geq 0$ for sufficiently small ϵ. Thus, $\boldsymbol{P}_2(\alpha)$ is bounded below and has finite norm and thus $\boldsymbol{A}_2(\alpha) = \boldsymbol{A}(\alpha) - \boldsymbol{B}(\alpha)\boldsymbol{K}_2\big(\epsilon, \boldsymbol{P}_1(\alpha)\big)$ is Hurwitz. Hence, $\boldsymbol{K}_2\big(\epsilon, \boldsymbol{P}_1(\alpha)\big)$ is stabilizing. Repeating the above analysis for $k = 1,2,\cdots$, the statement of the theorem is proved.

Next, using Theorem 6 we show that $\lambda^j \in \mathcal{N}_j(\epsilon) \coloneqq \{s \in C_- : \left|s - \lambda_o^j\right| < \epsilon < 1\}$ for a sufficiently small ϵ. Consider the following Theorem.

Theorem 7: Let the coefficients of the system matrices $\boldsymbol{A}(\alpha)$, and $\boldsymbol{B}(\alpha)$ be analytic functions of the scheduling parameter α, and Assumption 4 hold. Then, using Theorem 6, for sufficiently small ϵ the following holds:

$$\lambda^j = \lambda_o^j + \mathcal{O}(\epsilon), j = 1,2,\cdots,n. \tag{41}$$

Proof:

From Theorem 6, we have:

$$\boldsymbol{K}^*(\alpha_{l+1}) = \boldsymbol{K}^*(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon). \tag{42}$$

Thus, (8) implies that:

$$\boldsymbol{K}(\alpha) = \boldsymbol{K}^*(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon), \tag{43}$$

for all $\alpha \in [\alpha_l, \alpha_{l+1}]$. Also, the elements of $\boldsymbol{A}(\alpha)$, and $\boldsymbol{B}(\alpha)$ are analytic functions of ϵ, for small ϵ, we have, $\boldsymbol{A}(\alpha) = \boldsymbol{A}(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon)$, and $\boldsymbol{B}(\alpha) = \boldsymbol{B}(\alpha_l) + \mathcal{O}(\epsilon)$ for all $\alpha \in [\alpha_l, \alpha_{l+1}]$. Thus,

$$\boldsymbol{A}_c(\alpha) = \boldsymbol{A}(\alpha) - \boldsymbol{B}(\alpha)\boldsymbol{K}(\alpha) = \boldsymbol{A}_c(\alpha_l) + \boldsymbol{\mathcal{O}}(\epsilon). \tag{44}$$

Thus, using the results on the analytic perturbation of eigenvalues, the statement of the theorem holds.

Thus, it can be said that if ϵ is sufficiently small, the set $\sigma\big(\boldsymbol{A}_c(\alpha)\big)$ is in a small neighborhood of $\sigma\big(\boldsymbol{A}_c(\alpha_l)\big)$ for all $\alpha \in [\alpha_l, \alpha_{l+1}]$.

Having answered the question of stability for fixed $\alpha \in [\alpha_l, \alpha_{l+1}]$, the next question arises on the stability of the LPV system (5). From the results on slow-varying systems ([24], [43]), we have that if the rate of change of α is sufficiently small, then the stability of the LPV system (5) under the

control action (7) can be deduced from the stability analysis of the corresponding frozen systems. Readers are advised to check the references [24] and [43] for further details.

The gain scheduling-based learning algorithm is presented in Fig. 5. The algorithm starts by initializing the initial stabilizing controllers $\boldsymbol{K}_0^1, \boldsymbol{K}_0^2, \cdots, \boldsymbol{K}_0^n$ for the finite velocity points $\mathrm{V}_\mathrm{x}^1, \mathrm{V}_\mathrm{x}^2, \cdots, \mathrm{V}_\mathrm{x}^\mathrm{n}$ where we wish to freeze our system and learn the optimal controller gains $\boldsymbol{K}^{1*}, \boldsymbol{K}^{2*}, \cdots, \boldsymbol{K}^{n*}$. Next, we collect the position data $\mathrm{x}_{\mathrm{AV}}, \mathrm{x}_{LC}, \mathrm{x}_{LT}, \mathrm{x}_{FC}, \mathrm{x}_{FT}$ and feed it to the lane change decision module (see Fig. 3). The lane change decision module decides whether to do a lane change or remain in the desired lane based on the safety conditions explained above. In any situation, we collect the actual velocity data $\mathrm{V}_\mathrm{x}^{\mathrm{AV}}$ of the AV. The gain scheduling technique suggests that we need to freeze our system at all the velocity points $\mathrm{V}_\mathrm{x}^1, \mathrm{V}_\mathrm{x}^2, \cdots, \mathrm{V}_\mathrm{x}^\mathrm{n}$, but in a practical scenario this is not possible as when the AV is on the road it would be very challenging to maintain a constant $\mathrm{V}_\mathrm{x}^{\mathrm{AV}}$. Thus, we define a tolerance value ϵ_1 such that when $\mathrm{V}_\mathrm{x}^{\mathrm{AV}}$ is close to one of the $\mathrm{V}_\mathrm{x}^\mathrm{i}$'s and $\left|\mathrm{V}_\mathrm{x}^{\mathrm{AV}} - \mathrm{V}_\mathrm{x}^\mathrm{i}\right| \leq \epsilon_1$ we assume $\mathrm{V}_\mathrm{x}^{\mathrm{AV}} = \mathrm{V}_\mathrm{x}^\mathrm{i}$ and start collecting data for learning $\boldsymbol{K}^i$ for $\mathrm{V}_\mathrm{x}^\mathrm{i}$ and store in the database. It must be noted that the longitudinal velocity of the car might vary and the condition $\left|\mathrm{V}_\mathrm{x}^{\mathrm{AV}} - \mathrm{V}_\mathrm{x}^\mathrm{i}\right| \leq \epsilon_1$ might not be always satisfied when we start collecting data for $\mathrm{V}_\mathrm{x}^\mathrm{i}$. In such scenario, we again start from the very beginning step of collecting the position data $\mathrm{x}_{\mathrm{AV}}, \mathrm{x}_{LC}, \mathrm{x}_{LT}, \mathrm{x}_{FC}, \mathrm{x}_{FT}$ and repeat all the steps as explained above until we have collected enough data. Once we collect enough data, say m samples for a particular velocity $\mathrm{V}_\mathrm{x}^\mathrm{i}$, we pass the data to the learning module (see Fig. 4). The learning module then returns the learned controller gain $\boldsymbol{K}^{i*}$ and it is stored in the database. The flag learned $\mathrm{V}_\mathrm{x}^\mathrm{i}$ is used to avoiding repeated learning for the same $\mathrm{V}_\mathrm{x}^\mathrm{i}$. Once, a gain $\boldsymbol{K}^{i*}$ is learned for a $\mathrm{V}_\mathrm{x}^\mathrm{i}$, we change $\boldsymbol{K}_0^1$ with $\boldsymbol{K}^{i*}$ and use the new controller for the AV maneuvers. Once $\boldsymbol{K}^{i*}$ and $\boldsymbol{K}^{i+1*}$ for two given adjacent points are learned, we define the interval $[\mathrm{V}_\mathrm{x}^\mathrm{i}, \mathrm{V}_\mathrm{x}^{\mathrm{i}+1})$ and use the interpolated gain given in (26) to obtain the control signal whenever $\mathrm{V}_\mathrm{x}^\mathrm{i}$ lies the interval $[\mathrm{V}_\mathrm{x}^\mathrm{i}, \mathrm{V}_\mathrm{x}^{\mathrm{i}+1})$.

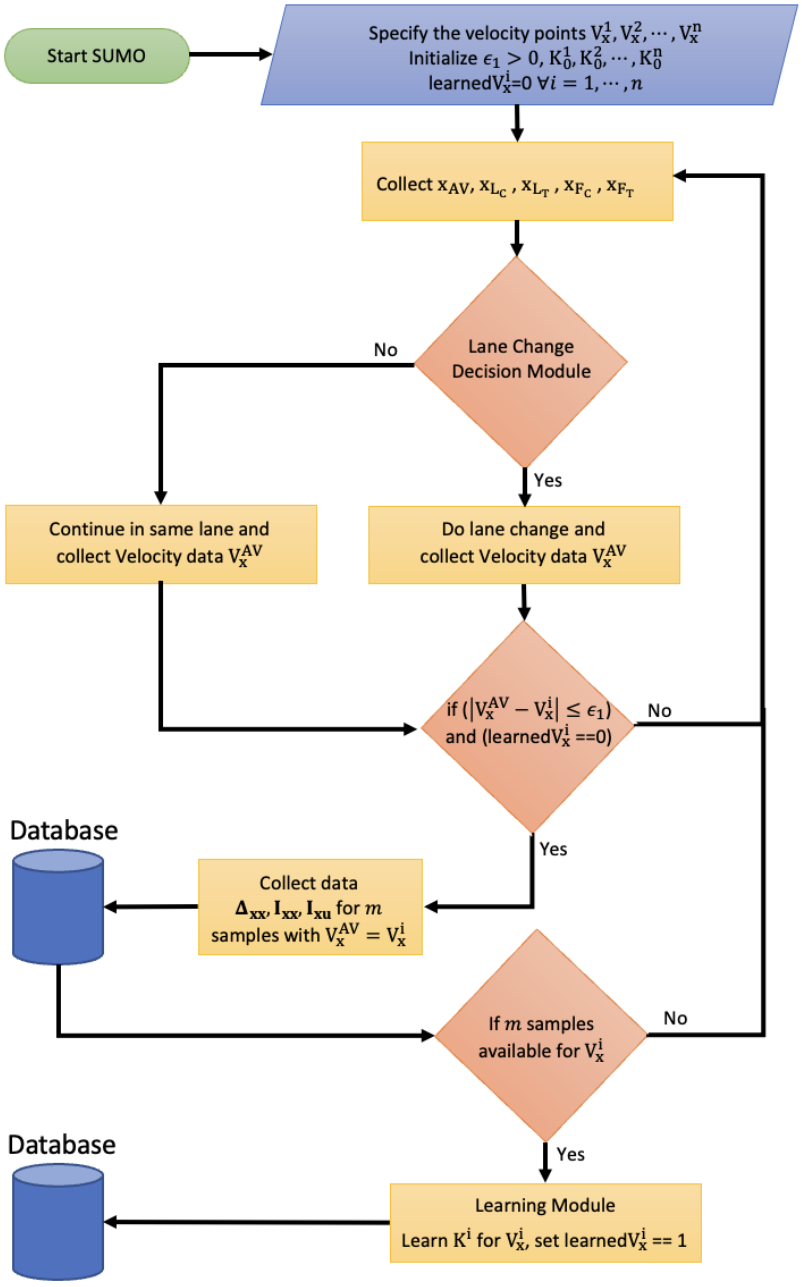


Fig. 5: Learning-based gain scheduling

A Data-Driven Approach to Safety-Critical Systems Using Control Barrier Function

In this section, we develop a data-driven formulation for the safety of safety-critical systems using control barrier functions. Consider the following definitions:

Definition 1: A continuous function $\beta: [0, \infty) \to [0, \infty)$ is a κ-function if the following hold:

- It is strictly increasing;
- $\beta(0) = 0$.

Definition 2: A continuous function $\beta: [0, \infty) \to [0, \infty)$ is a κ_∞-function if it is κ-function and the following holds:

- $\lim_{s \to \infty} \beta(s) = \infty$.

Definition 3: A continuous function $\beta: [-\infty, \infty) \to [-\infty, \infty)$ is an extended κ_∞-function if

- It is strictly increasing;
- $\beta(0) = 0$;
- $\lim_{s \to \infty} \beta(s) = \infty$, $\lim_{s \to -\infty} \beta(s) = -\infty$.

Consider an affine-in-the-control system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}, \tag{44}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$ are the state and control input, respectively.

The control objective is to keep the state of the above system in a closed admissible set $\mathcal{C} \subset \mathbb{R}^n$ defined as:

$$\begin{aligned} \mathcal{C} &= \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\}, \\ \partial\mathcal{C} &= \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) = 0\}, \\ Int(\mathcal{C}) &= \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) > 0\}, \end{aligned} \tag{45}$$

where $h: \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function. In addition, we assume that $\mathcal{C}$ is nonempty, that is, $Int(\mathcal{C}) \neq \emptyset$. For the system (44) with control input $\boldsymbol{u}$, a function $h(\boldsymbol{x})$ is a *control barrier function* (CBF) with respect to the admissible set $\mathcal{C}$ if there exists an extended κ_∞-function β such that for the control system (44):

$$\sup_{\mathrm{u} \in \mathbb{R}^m} \left[L_{\boldsymbol{f}} h(\boldsymbol{x}) + L_{\boldsymbol{g}} h(\boldsymbol{x}) \boldsymbol{u} \right] \geq -\beta\big(h(\boldsymbol{x})\big), \tag{46}$$

where

$$L_f h(\boldsymbol{x}) = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} f(\boldsymbol{x}),$$
$$L_g h(\boldsymbol{x}) = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} g(\boldsymbol{x}). \tag{47}$$

For the safety of AV along the longitudinal direction consider the following barrier functions:

$$h_1 \coloneqq -x_{\mathrm{AV}} + x_{LC} - S_{LC}(t) \geq 0,$$
$$h_2 \coloneqq x_{\mathrm{AV}} - x_{FC} - S_{FC}(t) \geq 0,$$
$$h_3 \coloneqq -x_{\mathrm{AV}} + x_{LT} - S_{LT}(t) \geq 0,$$
$$h_4 \coloneqq x_{\mathrm{AV}} - x_{FT} - S_{FT}(t) \geq 0,$$

where x_{AV} is the longitudinal position of the AV, $x_i, i \in \{LC, FC, LT, LC\}$ are the longitudinal position of the surrounding vehicles, $S_i(t) = L + hv_i(t)$ is the safety distance, L is the length of the vehicles, h is the headway time and $v_i(t)$ is the velocity of the i^{th} vehicle. Here AV is the autonomous vehicle, LC is the lead vehicle in the current lane, FC is the following vehicle in the current lane, LT is the lead vehicle in the target lane, FT is the following vehicle in the target lane.

Combining the longitudinal and lateral dynamics leads to:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{B}\,\boldsymbol{u}, \tag{48}$$

where $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A_{la}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A_{lo}} \end{bmatrix} \in \boldsymbol{R^{n_1 \times n_1}}$, $\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b_{la}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{b_{lo}} \end{bmatrix} \in \boldsymbol{R^{n_1 \times m_1}}$, $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x_{la}} \\ \boldsymbol{x_{lo}} \end{bmatrix} \in \boldsymbol{R^{n_1}}$, $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u_{la}} \\ \boldsymbol{u_{lo}} \end{bmatrix} \in \boldsymbol{R^{m_1}}$, $\boldsymbol{x_{la}} = [e_1(t), \dot{e_1}(t), e_2(t), \dot{e_2}(t)]^T$, $\boldsymbol{x_{lo}} = [x_{AV}, \dot{x}_{AV}]^T$.

Equation (46) for h_1 to h_4 can be obtained as follows:

$$\boldsymbol{C_1 A x} + \boldsymbol{C_1 B u} \geq -h_1(\boldsymbol{x}),$$
$$\boldsymbol{C_2 A x} + \boldsymbol{C_2 B u} \geq -h_2(\boldsymbol{x}),$$
$$\boldsymbol{C_3 A x} + \boldsymbol{C_3 B u} \geq -h_3(\boldsymbol{x}),$$
$$\boldsymbol{C_4 A x} + \boldsymbol{C_4 B u} \geq -h_4(\boldsymbol{x}), \tag{49}$$

where, $\boldsymbol{C}_1 = \boldsymbol{C}_3 = [0,0,0,0,-1,0]$, $\boldsymbol{C}_2 = \boldsymbol{C}_4 = [0,0,0,0,1,0]$.

Note that the optimal data-driven controller for (48) can be computed by following the algorithm given in Fig. 5. Let $\boldsymbol{K}^i$ be the controller gain for the i^{th} scheduling point. Then, the safe controller can be obtained by solving the following optimization problem:

$$\begin{aligned} \boldsymbol{u_i}(\boldsymbol{x}) &= \text{argmin}_{u\in\mathbb{R}^{m_1}} 0.5||\boldsymbol{u_i} - \boldsymbol{K}^i\boldsymbol{x}||^2 \\ &\text{subject to } (49). \end{aligned} \tag{50}$$

However, (49) uses the knowledge of $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices. The data-driven solution is given next that does not use the information of the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices. Note that the derivative of any barrier function h_j given above can be written as:

$$\dot{h}_j(\boldsymbol{x}) = \boldsymbol{C}_j\boldsymbol{A}\boldsymbol{x} + \boldsymbol{C}_j\boldsymbol{B}\boldsymbol{u}, j = 1,\cdots,4. \tag{51}$$

Then along the solutions of (51), it follows that

$$h_j\big(\boldsymbol{x}(t+\delta t)\big) - h_j\big(\boldsymbol{x}(t)\big) = \int_t^{t+\delta t} \boldsymbol{C}_j\boldsymbol{A}\boldsymbol{x}\, d\tau + \int_t^{t+\delta t} \boldsymbol{C}_j\boldsymbol{B}\boldsymbol{u}\, d\tau. \tag{52}$$

For any positive integer l, define $\boldsymbol{\Lambda^j} \in \mathbb{R}^{l_1\times 1}$, $\boldsymbol{J^j}_{\boldsymbol{x}} \in \mathbb{R}^{l\times n_1}$, $\boldsymbol{J^j}_{\boldsymbol{u}} \in \mathbb{R}^{l\times m_1}$ as follows for $0 \le t_1 < t_2 < \cdots < t_l$:

$$\begin{aligned} \boldsymbol{\Lambda^j} &= \big[h_j\big(\boldsymbol{x}(t_2)\big) - h_j\big(\boldsymbol{x}(t_1)\big),\ldots,h_j\big(\boldsymbol{x}(t_l)\big) - h_j\big(\boldsymbol{x}(t_{l-1})\big)\big]^{\boldsymbol{T}}, \\ \boldsymbol{J^j}_{\boldsymbol{x}} &= \left[\int_{\boldsymbol{t_1}}^{\boldsymbol{t_2}} (\boldsymbol{x}\otimes 1)\, d\tau, \int_{\boldsymbol{t_2}}^{\boldsymbol{t_3}} (\boldsymbol{x}\otimes 1)\, d\tau,\cdots,\int_{\boldsymbol{t_{l-1}}}^{\boldsymbol{t_l}} (\boldsymbol{x}\otimes 1)\, d\tau\right]^{\boldsymbol{T}}, \\ \boldsymbol{J^j}_{\boldsymbol{u}} &= \left[\int_{\boldsymbol{t_1}}^{\boldsymbol{t_2}} (\boldsymbol{u}\otimes 1)\, d\tau, \int_{\boldsymbol{t_2}}^{\boldsymbol{t_3}} (\boldsymbol{u}\otimes 1)\, d\tau,\cdots,\int_{\boldsymbol{t_{l-1}}}^{\boldsymbol{t_l}} (\boldsymbol{u}\otimes 1)\, d\tau\right].^{\boldsymbol{T}} \end{aligned} \tag{53}$$

Using (52) and (53), we have:

$$\tilde{\boldsymbol{\Gamma}}_{\boldsymbol{j}} \begin{bmatrix} \text{vec}\big(\boldsymbol{C}_j\boldsymbol{A}\big) \\ \text{vec}\big(\boldsymbol{C}_j\boldsymbol{B}\big) \end{bmatrix} = \boldsymbol{\Lambda}^{\mathbf{j}}, \tag{54}$$

where $\tilde{\boldsymbol{\Gamma}}_{\boldsymbol{j}} = [\boldsymbol{J^j}_{\boldsymbol{x}} \quad \boldsymbol{J^j}_{\boldsymbol{u}}]$.

***Assumption 6**:* There exists a sufficiently large integer $l > 0$, such that

$$\operatorname{rank}\left(\tilde{\boldsymbol{\Gamma}}_{\boldsymbol{j}}\right) = \mathrm{n}_1 + \mathrm{m}_1. \tag{55}$$

Under Assumption 6, $\boldsymbol{C}_j\boldsymbol{A}$ and $\boldsymbol{C}_j\boldsymbol{B}$ can be computed as follows:

$$\begin{bmatrix} \operatorname{vec}\left(\boldsymbol{C}_j\boldsymbol{A}\right) \\ \operatorname{vec}\left(\boldsymbol{C}_j\boldsymbol{B}\right) \end{bmatrix} = \tilde{\boldsymbol{\Gamma}}_{\boldsymbol{j}}{}^{+}\boldsymbol{\Lambda}^{\mathrm{j}}, \tag{56}$$

where $\tilde{\boldsymbol{\Gamma}}_{\boldsymbol{j}}{}^{+}$ is the pseudo inverse of $\tilde{\boldsymbol{\Gamma}}_{\boldsymbol{j}}$.

Lane Change Risk Index

In this section, we perform a comparative study between the proposed gain scheduling controller and the MPC for the lane-changing risks when both controllers are used for a lane-changing maneuver in a non-cooperative scenario. To evaluate the safety, we use the lane change risk index (LCRI) proposed in the [44]. The authors in [44] uses stopping sight distance (SSD) and stopping distance index (SDI) to compute two risk indicators: risk exposure level (REL) and risk severity level (RSL). SDI is a discrete measure used to determine whether a given car-following event is safe by comparing SSDs for the preceding vehicle and the following vehicle. The REL indicates how long a subject vehicle is exposed to a hazardous situation that could potentially lead to a crash while making a lane change. Meanwhile, RSL represents the severity of the crash that would occur if a subject vehicle does not make the appropriate evasive maneuver. Then, a fault tree analysis (FTA), which is a well-known technique for risk analysis, is adopted to integrate the REL and the RSL. As a result, a new index to estimate the probability of failing to make a safe lane change, which is referred to as the lane change risk index (LCRI), is proposed.

The SSD is computed as:

$$SSD_i(t) = \frac{V_i(t)^2}{254 \times (f \pm g)} + t_r \times V_i(t) \times 0.278, \tag{57}$$

where $V_i(t)$ is the vehicle speed in kph, f is the coefficient of friction, typically for a poor, wet pavement, g is the grade decimal, t_r is the perception-reaction time (2.5s), $i \in \{LT, FT, AV, LC, FC\}$. Once, the SSDs are computed, one can compute the SDIs as follows:

$$SDI_{i,j}(t) = \begin{cases} \text{safe, if } S_{i,j}(t) + SSD_i(t) - SSD_j(t) - l_i > 0, \\ \text{unsafe, otherwise,} \end{cases} \quad (58)$$

where, $SDI_{i,j}(t)$ is the stopping distance index for the front vehicle i and the following vehicle j, $S_{i,j}(t)$ is the front spacing between the front vehicle i and the following vehicle j, $SSD_i(t)$ is the stopping distance index for the front vehicle, $SSD_j(t)$ is the stopping distance index for the following vehicle, l_i is the length of the front vehicle. Note that for the group of vehicles $\{AV, LT, LC\}, i \in \{LT, LC\}, j = AV$, and for the group of vehicles $\{AV, FT, FC\}, i = AV, j \in \{LF, LC\}$.

Then, using SDI, REL and RSL are computed as follows:

$$REL_{i,j} = \frac{ULCD}{TLCD}, \quad (59)$$

where $ULCD$ is the unsafe lane change distance, $TLCD$ is the total lane change distance.

$$RSL_{i,j} = \frac{\max\left(-SDI_{i,j}(t)\right)}{SDI_{cri}}, \quad (60)$$

where SDI_{cri} is obtained when a crash occurs while the subject vehicle is traveling at the highest speed. Next, using fault tree analysis one can obtain the crash probabilities as ([44]):

$$\phi_k = REL_{i,j} \times RSL_{i,j}, \quad (61)$$

where, $k = 1,2,3,4$. Next, the probability of lane change failure for the AV can be obtained as:

$$\phi_{AV} = 1 - \prod_{k=1}^{4}(1 - \phi_k). \quad (62)$$

Simulation and Experimental Results

A. Simulation results

We obtain each $\boldsymbol{K}(\alpha_i)$ by means of the learning-based control technique discussed above which guarantees the stability for each of the fixed α_i's. To guarantee the stability of the overall system, we need that $V_x(t)$ is slowly varying. Since, $\dot{V}_x(t) = u_{lo}/m$, and $u_{lo} = -\boldsymbol{K_{lo}}\boldsymbol{x_{lo}}$ one needs to design $\boldsymbol{K_{lo}}$ such that that vehicle acceleration has a small magnitude. We have generated the results by implementing the above techniques of learning-based gain scheduling and lane change decision-making in SUMO simulation. The SUMO simulation time started at t_0 = 0s and terminated at t_f = 80s. Data from SUMO environment is collected at every 0.01s. We have

generated the results by assuming the distance between FT and LT remains constant for a cooperative scenario and varying the headway time (h) from 0.5s to 1s.

This section discusses the effectiveness of the proposed gain scheduling-based learning-based controller and the lane change decision-making algorithm by implementing them in SUMO. We assume that the AV learns in a cooperative scenario where the neighboring vehicles of the AV are cooperative with the AV while the AV starts changing the lane. By keeping the distance between FT and LT constant, we have tested the proposed methodology by varying h from 0.5s to 1s. We have observed that the AV could change the lane for all the considered h. We use the following weight matrices for the lateral controller design $\boldsymbol{Q} = \text{diag}([20,50,2000,3000]) = \text{diag}([q_1, q_2, q_3, q_4])$, and $\boldsymbol{R}$ = 1. The desired orientation of the vehicle $\psi_{des} = 90°$. For the purpose of learning with an initial control gain $\boldsymbol{K}_0$, we apply the control input $\boldsymbol{u} = -\boldsymbol{K}_0\boldsymbol{x}_{\boldsymbol{la}} + e$, where e is noise which is obtained using the summations of sinusoidal signals with randomly distributed frequencies. Note that the noise e is deterministic. The choice of the $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices is done considering the passenger and driver comfort, and low fuel use. The diagonal entries q_1, q_2 will penalize $\left(e_1(t), \dot{e}_1(t)\right)$ of AV, and q_3, q_4 will penalize $\left(e_2(t), \dot{e}_2(t)\right)$ which will ensure passenger and driver comfort. Increasing q_1, q_2 will make the controller more aggressive and might increase fuel consumption. Choosing $\boldsymbol{R} = 1$, we have found that the control input to the AV, i.e., the steering angle of the AV, can be computed such that the driver comfort is assured.

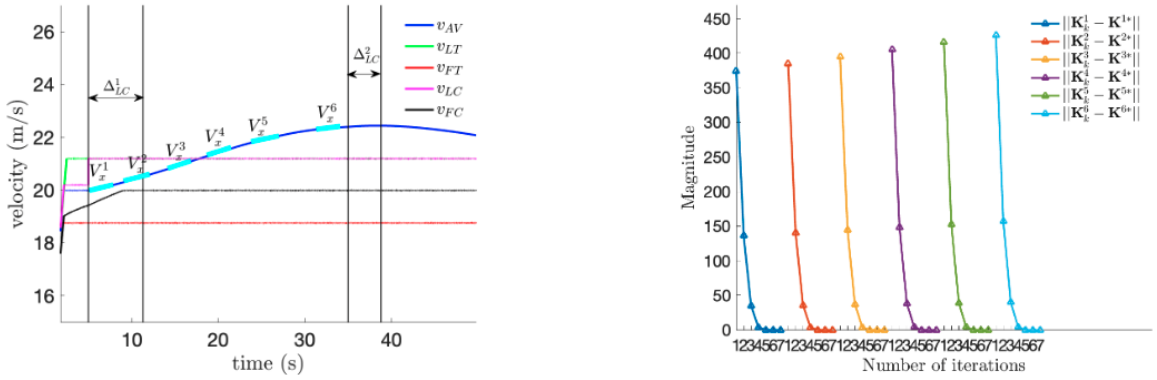


Figure 6: (a) Velocities of the vehicles (b) Convergence of gains

We learn optimal controllers when the AV longitudinal velocity changes by half a unit, i.e., $\epsilon = 0.5$. As the AV longitudinal velocity changes we learn optimal controllers for V_x^1 = 20m/s, V_x^2 = 20.5m/s, and V_x^3 = 21m/s, V_x^4 = 21.5m/s, V_x^5 = 22m/s, and V_x^6 = 22.5m/s (see Fig. 6a). We use the

algorithm presented in Fig. 5 to perform gain scheduling-based learning using these initial stabilizing control gains:

$$\boldsymbol{K}_0^1 = [0.5345, 0.0233, 88.5456, 92.4413],$$

$$\boldsymbol{K}_0^2 = [0.5345, 0.0246, 88.8799, 92.4431],$$

$$\boldsymbol{K}_0^3 = [0.5345, 0.0258, 89.2143, 92.4448],$$

$$\boldsymbol{K}_0^4 = [0.5345, 0.0270, 89.5487, 92.4463],$$

$$\boldsymbol{K}_0^5 = [0.5346, 0.0282, 89.8831, 92.4478],$$

$$\boldsymbol{K}_0^6 = [0.5345, 0.0293, 90.2176, 92.4491].$$

The tolerance ϵ_1 is set as 0.2. To demonstrate the learning process and application of the learned gains we perform the lane-changing two times.

Fig. 6a shows the velocity of the vehicles that are obtained from the SUMO environment, where Δ_{LC}^1 and Δ_{LC}^2 are the pre-learning and post-learning lane-changing times. The light blue strips in Fig. 6a indicate the intervals where $\left|V_x^{\mathrm{AV}} - V_x^i\right| \leq \epsilon_1$. Each of these intervals comprises of 50 data points. Thus, with a sampling rate of 0.01s, we collect data for 0.5s for learning for every V_x^i. After the optimal controller gains were obtained for every V_x^i, the initial gains are replaced with the updated gains and the AV maneuver is performed using the updated gains and using the interpolation formula presented in (8). It is evident for Fig. 6a that $\Delta_{LC}^2 \approx 4\mathrm{s} < \Delta_{LC}^1 \approx 8\mathrm{s}$, where Δ_{LC}^1 is the lane change duration. Thus, a significant improvement is seen in the lane-changing time with the update optimal gain scheduled controller. In this work, the lane change start time is defined as the time when the AV decides to do a lane change maneuver and the lane change end time is defined as the time when the AV's back bumper crosses the lane marking.

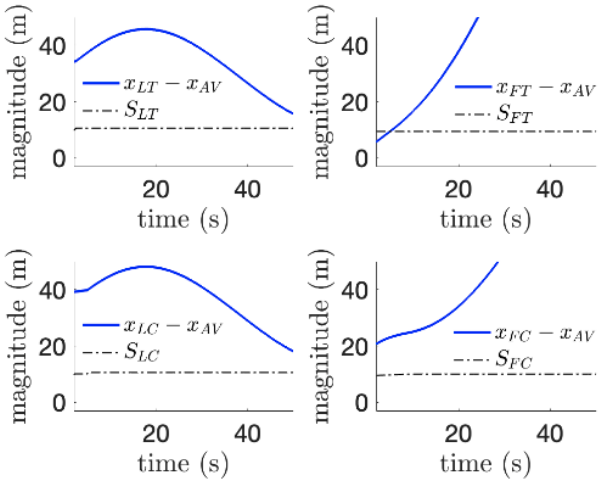


Figure 7: Distance of AV from surrounding vehicles.

Fig. 6b shows the convergence of the optimal gains. The ϵ_0 in Fig. 4 is set as 10^{-4}. It is clearly seen from Fig. 6b that the gains converge to the optimal gains with just 7 iterations. Thus, it can be said that for the proposed algorithm, 50 samples or in other words 0.5s data is enough for the learning algorithm to converge. The converged gains are:

$$\boldsymbol{K}^{1*} = [4.4721, 1.4437, 149.0064, 53.6656],$$

$$\boldsymbol{K}^{2*} = [4.4721, 1.4633, 151.5636, 53.6489],$$

$$\boldsymbol{K}^{3*} = [4.4721, 1.4831, 154.1037, 53.6320],$$

$$\boldsymbol{K}^{4*} = [4.4721, 1.5030, 156.6272, 53.6149],$$

$$\boldsymbol{K}^{5*} = [4.4721, 1.5231, 159.1316, 53.5974],$$

$$\boldsymbol{K}^{6*} = [4.4721, 1.5434, 161.6178, 53.5797].$$

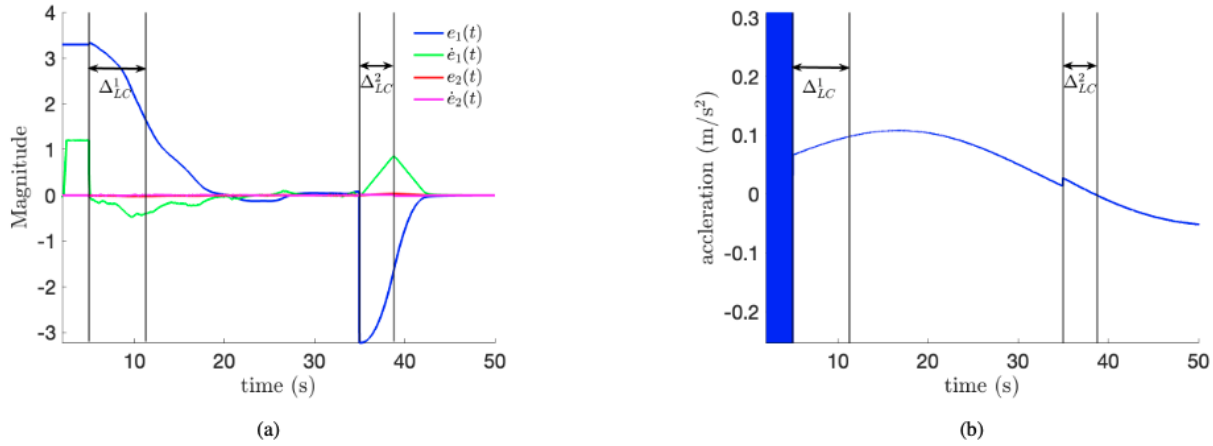


Figure 8: (a) Error states (b) Acceleration of AV.

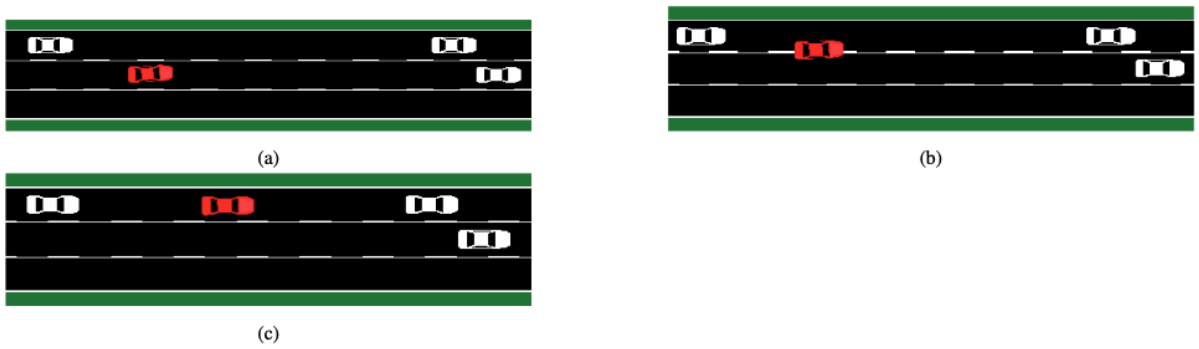


Figure 9: SUMO screenshots for non-cooperative scenario with h=0.5s: (a) $t = 46.5s$, (b) $t = 47.8s$, (c) $t = 53.8s$

Fig. 7 shows the safe distance of the AV with the surrounding vehicle. It can be observed at $t = 0s$, the AV was not at a safe distance from FT, thus the AV does not start a lane change maneuver. At $t \approx 4s$, the safety conditions for lane-changing satisfy for all the surrounding vehicles and the AV starts the first lane change maneuver that is completed in approximately 8s. For the second lane change maneuver, the vehicles are already at safe distance, thus the AV can safely start the lane change maneuver.

Fig. 8a shows the states of the lateral system. The states converge to zero with the application of the controllers obtained using the proposed methodology. It was mentioned above that the gain scheduled controller can guarantee overall system stability if the feedback law $\boldsymbol{K}_{lo}$ for the longitudinal motion can be obtained such that vehicle acceleration has a small magnitude. Here,

we have obtained the $\boldsymbol{K}_{lo}^{*} = [4.4721, 110.3821]$ with $\boldsymbol{Q}_{lo} = \text{diag}([1, 1])$, $\boldsymbol{R}_{lo} = 0.05$ using historical data. The choice of $\boldsymbol{Q}_{lo}$ and $\boldsymbol{R}_{lo}$ must be such that the acceleration has a lower magnitude. Fig. 8b shows the longitudinal acceleration profile of the AV. It can be seen that the acceleration magnitude is low.

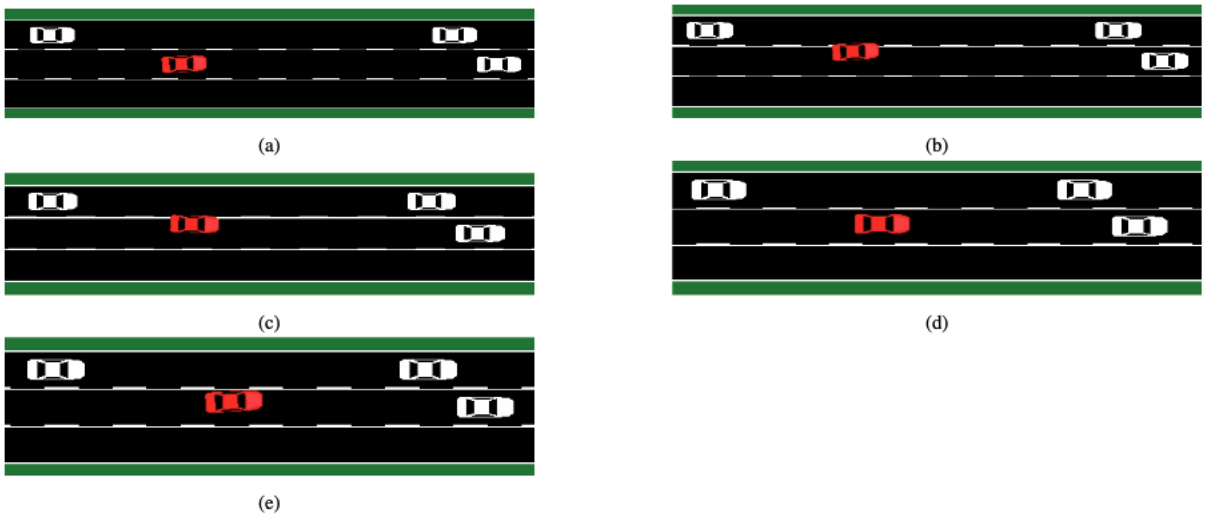


Figure 10: SUMO screenshots for non-cooperative scenario with h=0.65s: (a) $t = 49.1s$, (b) $t = 50.1s$, (c) $t = 50.9s$ (d) $t = 53.5s$, (e) $t = 54.5s$, (f) $t = 68.6s$

In a non-cooperative scenario, the FT is not cooperative with the AV while the AV starts changing the lane. By running the SUMO simulations for different values of h, it was observed that the AV was able to change lanes for all $h \leq 0.65s$. The scenarios are presented as screenshots for $h = 0.5s$ in Fig. 9, for $h = 0.65s$ in Fig. 10, for $h = 0.7s$ in Fig. 11. It can be seen that the AV was able to change the lane for $h = 0.5s$ and $h = 0.65s$. For demonstrating the lane abortion scenario clearly. The case of $h = 0.65s$ is elaborately explained next. Fig. 12a shows lane abortion, and the velocities of the vehicles are shown in Fig. 12b. The lane change starts at 49.1s. From Fig. 12a, it can be seen that the FT starts accelerating more than the AV. At around 50.9s, FT comes close to the AV and thus to maintain safety, the AV starts aborting the lane change and maneuvers back to the current lane at 53.5s. Again, at 54.5s when the safety conditions are satisfied, the AV starts maneuvering to the target lane. It must be noted that the plots in Fig. 12a are normalized for the sake of clarity in understanding. For the cases where $h \geq 0.7s$, the AV could not change the lane as the safety conditions did not satisfy for the simulation duration (see Fig. 11).

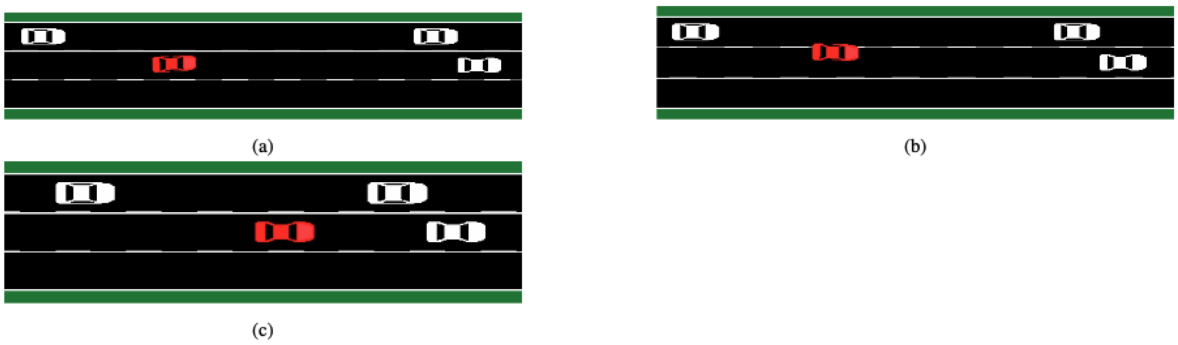
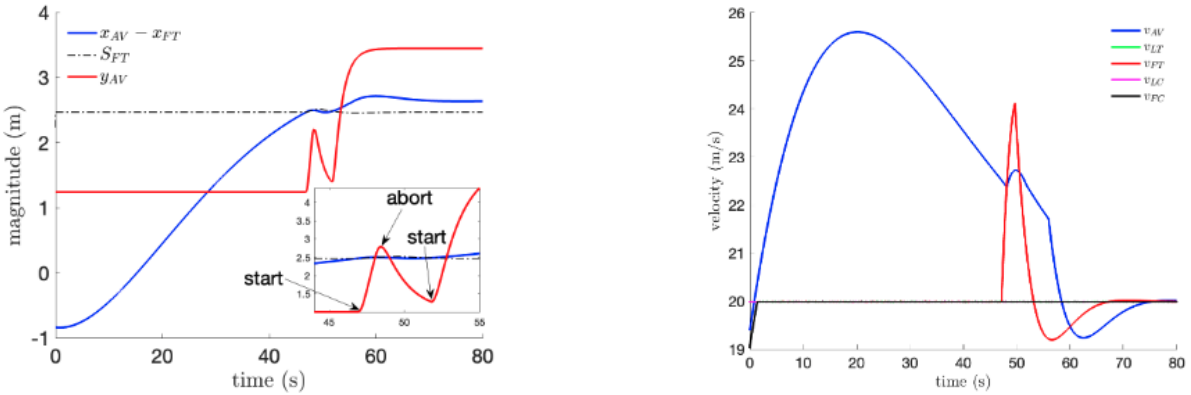


Figure 11: SUMO screenshots for non-cooperative scenario with h=0.7s: (a) $t = 49.5s$, (b) $t = 50.7s$, (c) $t = 50.9s$ (d) $t = 63.5s$



(a) (b)

Figure 12: (a) Lane abortion of AV, (b) Velocities during lane abortion

To evaluate if lane change could be possible with a constant gain instead of gain scheduling, we did SUMO simulations where the AV was made to change lane using constant gains. The results are given in Table 1, Fig. 13a and Fig. 13b. The headway time is 0.5s, and the lane change velocity is in the range of 21.5m/s to 23m/s. We do five simulations, where we make the AV change lane using controller gains ($\boldsymbol{K}_{V_x}$) trained for $V_x \in \{12m/s, 15m/s, 17m/s, 20m/s, 23m/s\}$. Fig. 13a shows the trajectories obtained using gain scheduling and constant gains. It can be seen that when the AV changes lane with $\boldsymbol{K}_{12}$, there is a small overshoot in the AV trajectory. This overshoot decreases as we make the AV change lane with the controller gain that is trained close to the actual lane change velocity. Also, as we make the AV change lane with the controller gain that is trained close to the actual lane change velocity, the AV trajectories converge to the trajectory obtained using gain scheduling (GS). A similar observation is seen with the convergence of AV lateral states (see Fig. 13b for e_1). Table 1 shows the cost obtained using the proposed gain-scheduling controller for lane-changing and the constant gain controllers for lane-changing. The cost is computed using $J = \int_{t_0}^{t_f} (\boldsymbol{x}^T \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R} \boldsymbol{u}) dt$. It can be seen that the

proposed gain-scheduling controller gives the least cost. This suggests that the gain-scheduling controller is optimal when compared to the constant gain controllers.

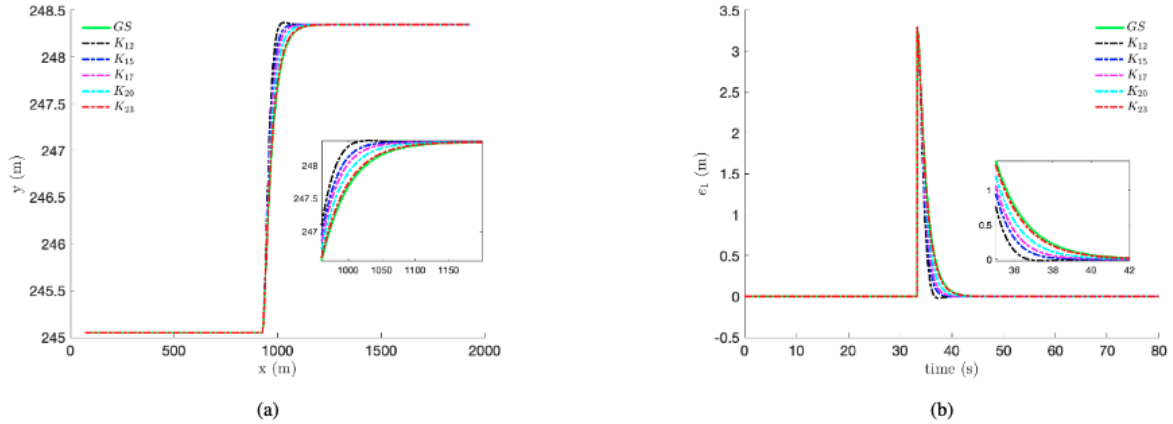


Figure 13: (a) AV trajectories obtained using gain scheduling and constant gains, (b) error state e_1 obtained using gain scheduling and constant gains.

Technique	Cost (J)
Gain scheduling	4.1510e+04
K_{12}	4.6470e+04
K_{15}	4.4074e+04
K_{17}	4.3033e+04
K_{20}	4.2063e+04
K_{23}	4.1543e+04

Table 1: Comparison with constant gain

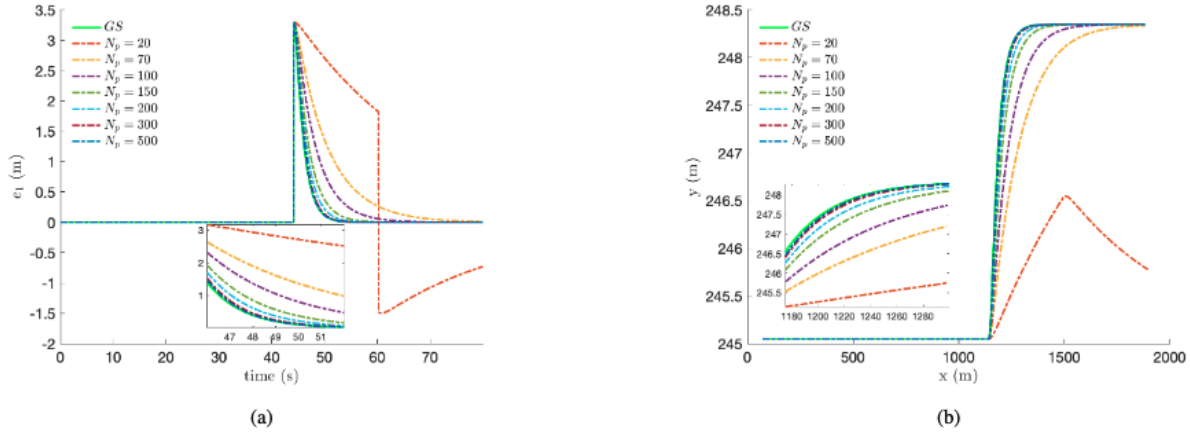


Figure 14: (a) AV trajectories obtained using gain scheduling and MPC, (b) error state e_1 obtained using gain scheduling and MPC.

MPC is also an optimal control technique. MPC tries to find the optimal control input at each time step by minimizing the cost function $J = \sum_{i=0}^{N_p-1}(\boldsymbol{x_i}^T \boldsymbol{Q} \boldsymbol{x_i} + \boldsymbol{u_i}^T \boldsymbol{R} \boldsymbol{u_i}) + \boldsymbol{x_{N_p}}^T \boldsymbol{Q}_{N_p} \boldsymbol{x_{N_p}}$, where $\boldsymbol{x_{N_p}}^T \boldsymbol{Q}_{N_p} \boldsymbol{x_{N_p}}$ is the terminal cost, and N_p is the prediction horizon. Designing a proper $\boldsymbol{Q}_{N_p}$ is essential for stability of the MPC controller. Also, one needs to properly select the prediction horizon N_p to attain a balance between accuracy and computation cost. In the literature, an MPC controller is used to track a trajectory generated by the trajectory planning module for a lane change. Here, we test if the MPC can be used as a lane-changing controller when a lane-change reference trajectory is not available. We have implemented the model-based MPC controller for lane change for $N_p \in \{20,70,100,150,200,300,500\}$ and compared the results with the proposed learning-based gain scheduling technique in a non-cooperative scenario with h=0.5s. The plots obtained from SUMO simulations are given in Fig. 14a and Fig. 14b. It can be seen that the MPC controller does not achieve satisfactory performance with small prediction horizon. As, the prediction horizon is increased, the performance is similar to the proposed learning-based gain scheduling technique. It was observed that for $N_p < 150$ the AV could not perform a successful lane change in a non-cooperative scenario. The SUMO simulation screenshots are given in Fig. 15 and Fig. 16. It can be seen that when $N_p = 100$, the AV could not change the lane as the MPC controller could not produce adequate control input (steering wheel angle). Whereas, when $N_p = 300$, the AV could successfully change the lane and the performance is like the proposed learning-based gain scheduling controller.

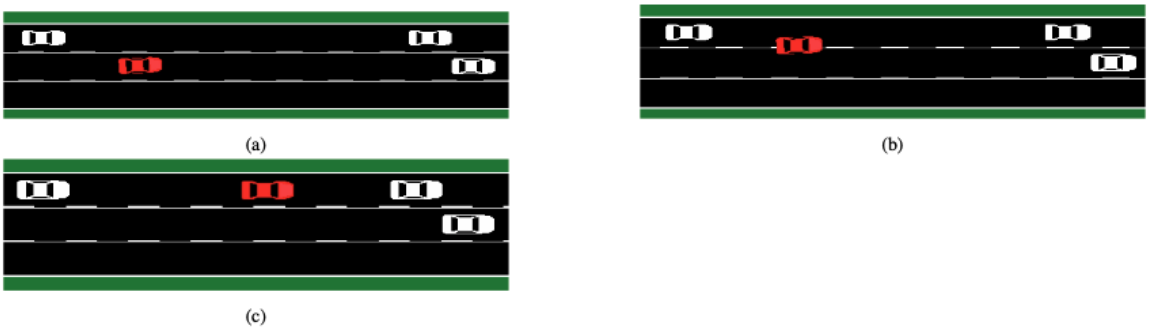


Figure 15: SUMO screenshots obtained for MPC controller in a non-cooperative scenario with h=0.5s, N_p=300: (a) $t = 46.8s$, (b) $t = 48.4s$, (c) $t = 53.5s$.

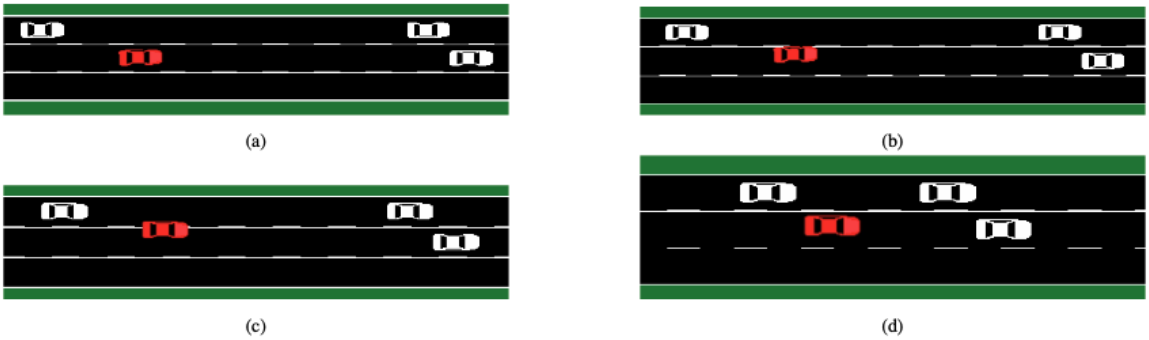


Figure 16: SUMO screenshots obtained for MPC controller in a non-cooperative scenario with h=0.5s, N_p=100: (a) $t = 47s$, (b) $t = 47.7s$, (c) $t = 49.5s$ (d) $t = 54s$.

It must be noted that, increasing the prediction horizon increases the computation time of MPC. The MPC used in this work is model-based and thus there is no learning time and thus to compare with the learning-based gain scheduling technique, we calculate the computation time of the MPC controller shown in Table 2. In Table 2, the computation start time in the time when the AV decides to do a lane change maneuver, and the computation end time is the time when the AV reaches the mid-point of the target lane. Note that the learning-based gain scheduling technique uses exploration noise. Hence, the learning time might vary each time the gain scheduling algorithm is executed. Thus, we execute the algorithm given in Fig. 5, 100 times to get a distribution of the learning times for different control gains. The histogram for learning times for each $\boldsymbol{K}$ is given in Fig. 17 and the histogram of total learning times is given in Fig. 19. Comparing the total computation time of MPC in Table 2 and the total learning times for the proposed technique in Fig. 19 it can be said that the proposed learning-based gain scheduling technique is computationally efficient when compared to MPC.

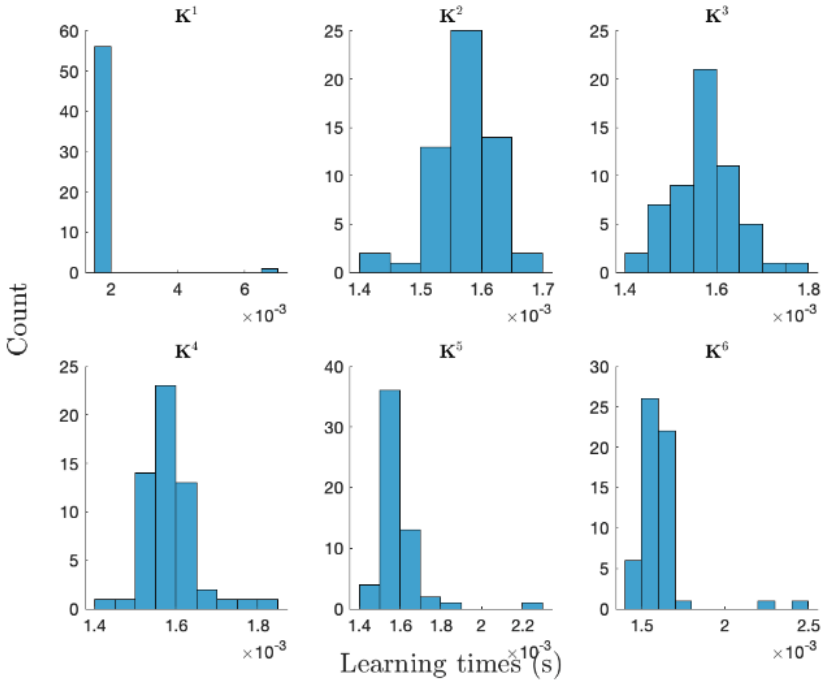


Figure 17: Histogram of learning times for each $\boldsymbol{K}$

N_p	Computation start, end time	Time steps	Avg. comp. time/step	Tot. comp. time
150	39.86s, 48.76s	900	0.0011s	0.99s
200	39.86s, 47.15s	730	0.0013s	0.95s
300	39.86s, 46.12s	630	0.002s	1.26s
500	39.86s, 45.82s	600	0.0034s	2s

Table 2: Computation time of MPC

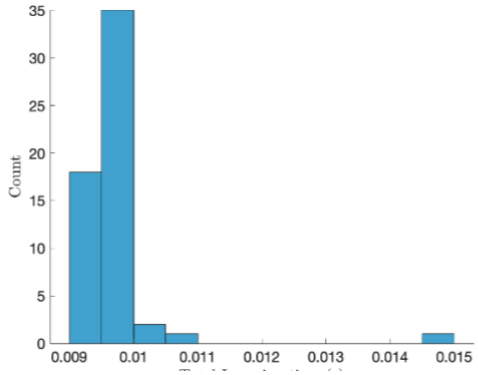


Figure 19: Histogram of total learning times in 100 runs obtained by adding the learning times of each $\boldsymbol{K}$ of the proposed gain scheduling algorithm in one run

In this work, the SDI for the AV and FT for the proposed gain scheduling controller and MPC are shown in Fig. 20a and Fig. 20b, respectively. As we want to determine the safety of AV from the non-cooperative vehicle FT, we have not considered the SDI of other vehicles. In this work, SDI_{cri} is set to 40m considering the spacing between two interacting vehicles is 0m and that the speed of the following vehicle is 100 kph.

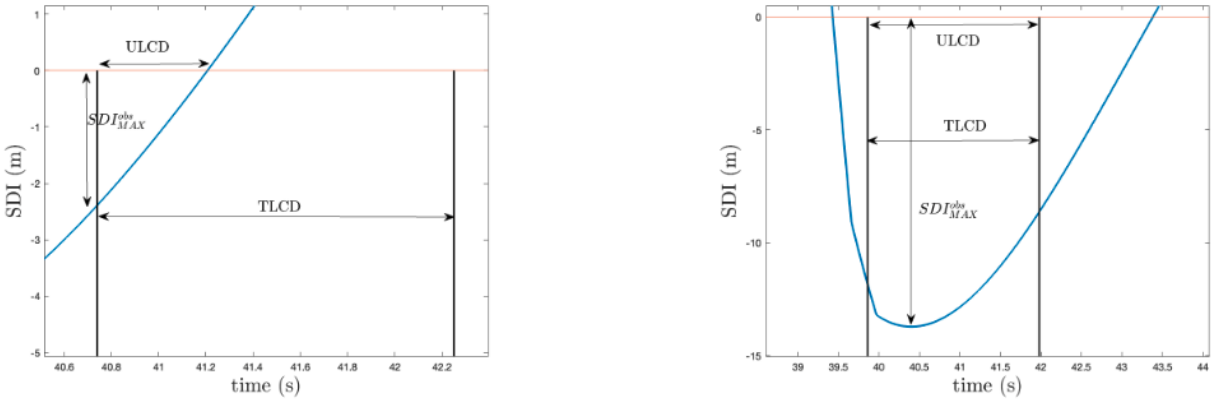


Figure 20: (a) SDI for GS obtained for AV and FT, (b) SDI for MPC obtained for AV and FT.

After computation, we have obtained the LCRI for gain scheduling ϕ_{AV}=0.05, and for MPC ϕ_{AV}=0.5 with 150 steps as prediction horizon. Also, note that the proposed learning-based gain scheduling technique needs only 50 samples to learn the scheduling gains. In a similar manner, we have computed LCRI for MPC for $N_p = 200,300,500$ and observed that as the prediction horizon is increased, the LCRI for MPC is improved. But note that increasing N_p increases the computation time, whereas the proposed methodology provides better safety in a non-cooperative scenario with lower computation costs. Thus, it can be said that the proposed gain-scheduling technique is safer and computationally efficient.

B. Experimental setup development and validation

Here we explain the AV prototypes we are building. We have tested the robustness of the data-driven controller by implementing it on the Raspberry Pi.

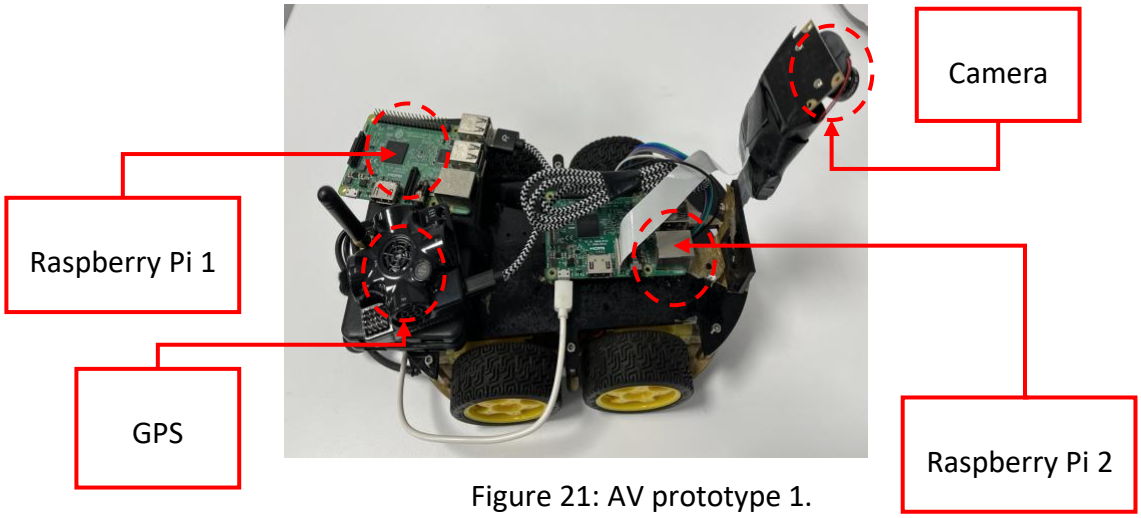


Figure 21: AV prototype 1.

Figure 21 shows the AV prototype 1. In this prototype, we have used two Raspberry Pi microcontrollers. Raspberry Pi 1 is connected to the GPS device which is used to collect the location data of the AV and Raspberry Pi 2 is connected to the camera and the AV motor control system. The position data from the Raspberry Pi 1 is sent to a local server which sends the data to the Raspberry Pi 2. The data-driven controller (see Fig. 4) and lane changing algorithm (see Fig. 3) are implemented on the Raspberry Pi 2. The GPS provides the (x, y) coordinates of the AV. Also, same GPS devices are mounted on the surrounding vehicles. Thus, we can acquire the position data of all the vehicles in the environment. The flow chart of signal transmission for AV prototype 1 is shown in Fig. 22 and the AV prototype 1 placed between lane markings is shown in Fig. 23.

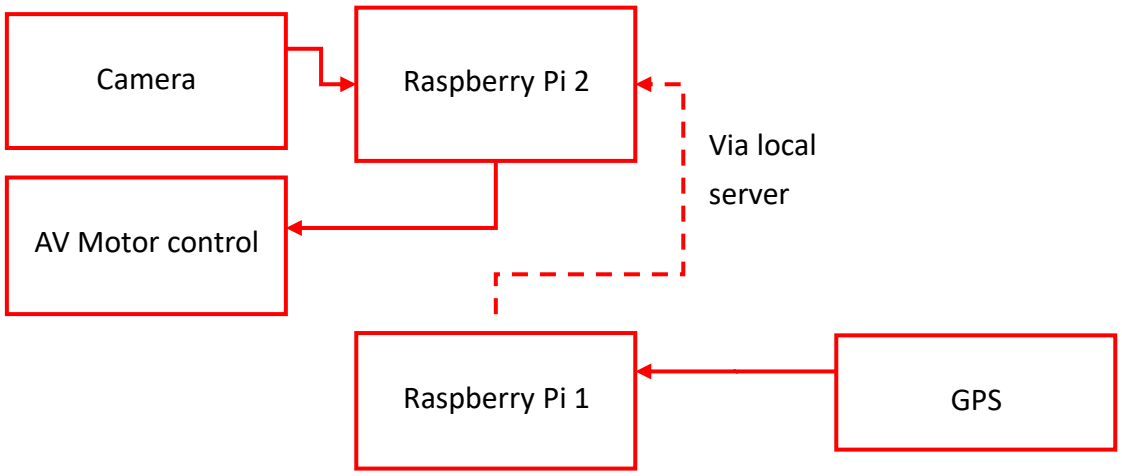


Figure 22: Signal transmission in AV prototype 1.

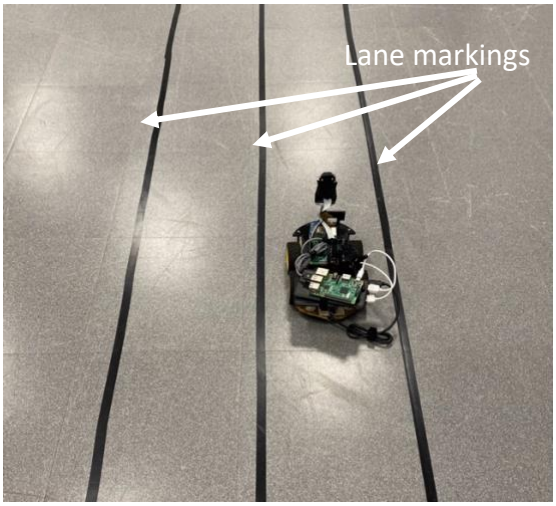


Figure 23: AV prototype 1 and lane markings.

To compute the optimal control signal for longitudinal and lateral motion, and evaluate safety distance, we need the information of the following states:

$$\boldsymbol{x_{la}} = [e_1(t), \dot{e_1}(t), e_2(t), \dot{e_2}(t)]^T,$$

$$\boldsymbol{x_{lo}} = [x_1(t), \dot{x}_1(t)]^T,$$

where x_1 is longitudinal position, and $\dot{x}_1(t) = x_2$ = longitudinal velocity, e_1 is the error between the distance of the center of gravity of the vehicle and the center line of the target lane, and e_2 be the orientation error of the vehicle with respect to the road. Thus, using the position data obtained from the GPS device, one can obtain $x_1, e_1, \text{and } e_2$. The derivates of these states can be computed using Euler approximation.

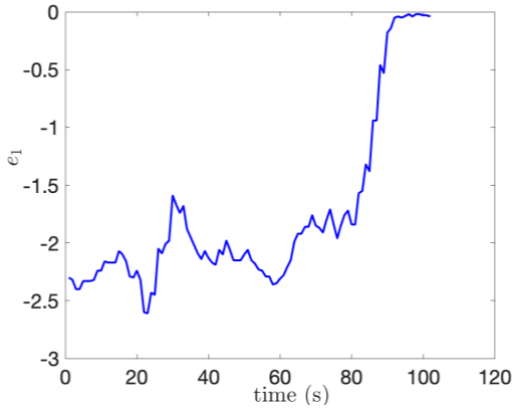


Figure 24: e_1 tends to a neighborhood of 0 despite measurement noise.

Figure 24 shows the result obtained by implementing the data-driven controller in Raspberry Pi 2. It can be seen that, with the optimal control law, the state e_1 converges to a small neighborhood of the origin. This implies that the lane change has been successful, and the data-driven controller is robust to measurement noise. The video of lane change has been uploaded as a separate file with this report. Interested readers can play the video to observe the lane change.

Conclusions and contributions

In this work, we have introduced an optimal data-driven control algorithm to solve the lane changing problem of AVs, where we make use of the online information of the state and input to solve the algebraic Riccati equation iteratively by using approximate/adaptive dynamic programming (ADP). In this work, we have assumed that the state and control input are received from a linear system. In order to make the proposed methodology applicable to non-linear and/or parameter varying systems, we have proposed a gain scheduling-based data-driven control technique to learn optimal gains. Also, we have developed a lane change decision making algorithm to ensure safe and efficient lane change. Safety is assured during lane changing by maintaining a safe distance from the surrounding vehicles. The proposed lane changing algorithm can make the AV perform a lane abortion if safety conditions are violated during lane change. The optimal data-driven gain-scheduling control algorithm and the lane change decision making algorithm has been validated by means of SUMO and MATLAB based computer simulations.

As compared to existing methodologies in the literature, our proposed method is completely data driven. We do not use or assume any information of the system parameters. We only assume the knowledge of the state vector and the control input, and derive a model-free optimal controller with guaranteed stability. It must be noted that many methodologies in the literature of lane changing does not guarantee optimal control of their AV. Many of the techniques that are proposed in the literature require to solve an optimization problem at every time step whereas our proposed methodology only requires to learn at specific time intervals with a smaller number of data points when the longitudinal velocity changes. Also, due to the fast convergence of the proposed methodology, it is suitable for real-time applications. Although we assume that we receive data from a linear model, the gain-scheduling based data-driven controller design adds to the versatility of the methodology that makes the proposed methodology applicable to non-linear systems and/or parameter varying systems as well. Also, with the obtained optimal gain, the lane changing time is seen to be considerably improved. The safety of lane change maneuver was evaluated using Lane Change Risk Index (LCRI) for the proposed learning-based gain scheduling controller and MPC. Also, the computation time of MPC and learning time of the proposed controller were compared. It was found that the proposed controller is safer and computationally efficient than MPC. Also, in this work we have proposed a data-driven approach to use Control Barrier Function (CBFs) to improve safety of safety-critical systems. Furthermore, the data-driven controller is implemented on a Raspberry Pi and it was found that the data-driven controller is robust to measurement noise.

References

[1] J. Chovan, L. Tijerina, G. Alexander, D. Hendricks, Examination of lane change crashes and potential IVHS countermeasures. final report, Tech. rep. (1994).

[2] Z. Wang, X. Shi, X. Li, Review of lane-changing maneuvers of connected and automated vehicles: models, algorithms and traffic impact analyses, Journal of the Indian Institute of Science 99 (4) (2019) 589–599.

[3] K. Ma, H. Wang, Z. Zuo, Y. Hou, X. Li, R. Jiang, String stability of automated vehicles based on experimental analysis of feedback delay and parasitic lag, Transportation research part C: emerging technologies 145 (2022) 103927.

[4] X. Li, Trade-off between safety, mobility and stability in automated ve- hicle following control: An analytical method, Transportation Research Part B: Methodological 166 (2022) 1–18.

[5] Y. Zhou, M. Wang, S. Ahn, Distributed model predictive control ap- proach for cooperative car-following with guaranteed local and stringstability, Transportation research part B: methodological 128 (2019) 69– 86.

[6] D. Bevly, X. Cao, M. Gordon, G. Ozbilgin, D. Kari, B. Nelson, J. Woodruff, M. Barth, C. Murray, A. Kurt, et al., Lane change and merge maneuvers for connected and automated vehicles: A survey, IEEE Transactions on Intelligent Vehicles 1 (1) (2016) 105–120.

[7] J. Nilsson, J. Silvlin, M. Brannstrom, E. Coelingh, J. Fredriksson, If, when, and how to perform lane change maneuvers on highways, IEEE Intelligent Transportation Systems Magazine 8 (4) (2016) 68–78.

[8] Z. Wang, X. Zhao, Z. Chen, X. Li, A dynamic cooperative lane-changing model for connected and autonomous vehicles with possible accelerations of a preceding vehicle, Expert Systems with Applications 173 (2021) 114675.

[9] Y. Luo, Y. Xiang, K. Cao, K. Li, A dynamic automated lane change maneuver based on vehicle-to-vehicle communication, Transportation Research Part C: Emerging Technologies 62 (2016) 87–102.

[10] M. Xu, Y. Luo, G. Yang, W. Kong, K. Li, Dynamic cooperative auto- mated lane-change maneuver based on minimum safety spacing model, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019, pp. 1537–1544.

[11] Z. Wang, X. Zhao, Z. Xu, X. Li, X. Qu, Modeling and field experiments on autonomous vehicle lane changing with surrounding human-driven vehicles, Computer-Aided Civil and Infrastructure Engineering 36 (7) (2021) 877–889.

[12] Z. Wang, X. Zhao, Z. Xu, X. Li, X. Qu, Modeling and field experiments on lane changing of an autonomous vehicle in mixed traffic, Computer- aided Civil and Infrastructure Engineering (2020).

[13] J. Nie, J. Zhang, W. Ding, X. Wan, X. Chen, B. Ran, Decentralized cooperative lane-changing decision-making for connected autonomous vehicles, IEEE access 4 (2016) 9413–9420.

[14] G. Xu, L. Liu, Y. Ou, Z. Song, Dynamic modeling of driver control strategy of lane-change behavior and trajectory planning for collision prediction, IEEE Transactions on Intelligent Transportation Systems 13 (3) (2012) 1138–1155.

[15] J. Nilsson, Y. Gao, A. Carvalho, F. Borrelli, Manoeuvre generation and control for automated highway driving, IFAC Proceedings Volumes 47 (3) (2014) 6301–6306.

[16] J. Suh, H. Chae, K. Yi, Stochastic model-predictive control for lane change decision of automated driving vehicles, IEEE Transactions on Vehicular Technology 67 (6) (2018) 4771–4782.

[17] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[18] Y. Jiang, Z.-P. Jiang, Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics, Au- tomatica 48 (10) (2012) 2699–2704.

[19] Y. Jiang, Z.-P. Jiang, Robust adaptive dynamic programming, John Wiley & Sons, 2017.

[20] W. Gao, Z.-P. Jiang, K. Ozbay, Data-driven adaptive optimal control of connected vehicles, IEEE Transactions on Intelligent Transportation Systems 18 (5) (2016) 1122–1133.

[21] W. Gao, Z.-P. Jiang, Adaptive dynamic programming and adaptive op- timal output regulation of linear systems, IEEE Transactions on Auto- matic Control 61 (12) (2016) 4164–4169.

[22] W. J. Rugh, J. S. Shamma, Research on gain scheduling, Automatica 36 (10) (2000) 1401–1425.

[23] C. Hoffmann, H. Werner, A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations, IEEE Transactions on Control Systems Technology 23 (2) (2014) 416–433.

[24] J. S. Shamma, Analysis and design of gain scheduled control systems, Ph.D. thesis, Massachusetts Institute of Technology (1988).

[25] J. S. Shamma, M. Athans, Analysis of gain scheduled control for non- linear plants, IEEE Transactions on Automatic Control 35 (8) (1990) 898–907.

[26] F. D. Bianchi, R. Mantz, C. Christiansen, Control of variable-speed wind turbines by lpv gain scheduling, Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology 7 (1) (2004) 1–8.

[27] C. Yuan, Y. Liu, F. Wu, C. Duan, Hybrid switched gain-scheduling control for missile autopilot design, Journal of Guidance, Control, and Dynamics 39 (10) (2016) 2352–2363.

[28] J. S. Shamma, J. R. Cloutier, Gain-scheduled missile autopilot design using linear parameter varying transformations, Journal of guidance, Control, and dynamics 16 (2) (1993) 256–263.

[29] D. Saussi´e, L. Saydy, O. Akhrif, C. B´erard, Gain scheduling with guardian maps for longitudinal flight control, Journal of guidance, con- trol, and dynamics 34 (4) (2011) 1045–1059.

[30] P. S. Saikrishna, R. Pasumarthy, N. P. Bhatt, Identification and mul- tivariable gain-scheduling control for cloud computing systems, IEEE Transactions on Control Systems Technology 25 (3) (2016) 792–807.

[31] S. Zhu, S. Y. Gelbal, B. Aksun-Guvenc, L. Guvenc, Parameter-space based robust gain-scheduling design of automated vehicle lateral control, IEEE Transactions on Vehicular Technology 68 (10) (2019) 9660–9671.

[32] E. Alcala, V. Puig, J. Quevedo, T. Escobet, Gain-scheduling lpv control for autonomous vehicles including friction force estimation and compen- sation mechanism, IET Control Theory & Applications 12 (12) (2018) 1683–1693.

[33] D. Kapsalis, O. Sename, V. Milan´es, J. J. Martinez, Gain-scheduled steering control for autonomous vehicles, IET Control Theory & Appli- cations 14 (20) (2020) 3451–3460.

[34] H. Zhang, X. Zhang, J. Wang, Robust gain-scheduling energy-to-peak control of vehicle lateral dynamics stabilisation, Vehicle System Dynam- ics 52 (3) (2014) 309–340.

[35] S. Chu, Z. Xie, P. K. Wong, P. Li, W. Li, J. Zhao, Observer-based gain scheduling path following control for autonomous electric vehicles subject to time delay, Vehicle System Dynamics 60 (5) (2022) 1602–1626.

[36] S. Chakraborty, L. Cui, K. Ozbay, Z.-P. Jiang, Automated lane changing control in mixed traffic: An adaptive dynamic programming approach, in: 2022 IEEE 25th International Conference on Intelligent Transporta- tion Systems (ITSC), IEEE, 2022, pp. 1823–1828.

[37] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," 2019 18th IEEE European Control Conference (ECC). IEEE, pp. 3420-3431, 2019.

[38] Y. Chen, H. Peng, and J. Grizzle, "Obstacle avoidance for low-speed autonomous vehicles with barrier function," IEEE Trans on Control Systems Technology, vol. 26, no. 1, pp. 194-206, 2017.

[39] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," IEEE Transactions on Automatic Control, vol. 62, no. 8, pp. 3861-3876, 2016.

[40] S. He, J. Zeng, B. Zhang, and K. Sreenath, "Rule-Based Safety-Critical Control Design using Control Barrier Functions with Application to Autonomous Lane Change," arXiv preprint arXiv:2103.12382 ,2021.

[41] Y. Pan, Q. Lin, H. Shah, and J. M. Dolan, "Safe planning for self-driving via adaptive constrained ILQR," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2377-2383, 2020.

[42] Kleinman, D., 1968. On an iterative technique for riccati equation computations. IEEE Transactions on Automatic Control 13, 114–115.

[43] Vidyasagar, M., 2002. Nonlinear systems analysis. SIAM.

[44] Park, H., Oh, C., Moon, J., Kim, S., 2018. Development of a lane change risk index using vehicle trajectory data. Accident Analysis & Prevention 110, 1–8.