Transportation Research Part B 187 (2024) 103026

Contents lists available at ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Automated lane changing control in mixed traffic: An adaptive dynamic programming approach

Sayan Chakraborty [a,*], Leilei Cui [a], Kaan Ozbay [b], Zhong-Ping Jiang [a,b]

[a] Control and Network Lab, Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, New York, NY 11201, USA

[b] C2SMARTER Center, Department of Civil and Urban Engineering, Tandon School of Engineering, New York University, New York, NY 11201, USA

ARTICLE INFO

Keywords:
Autonomous vehicle
Lane changing
Learning-based optimal control
Gain scheduling

ABSTRACT

The majority of the past research dealing with lane-changing controller design of autonomous vehicles (AVs) is based on the assumption of full knowledge of the model dynamics of the AV and the surrounding vehicles. However, in the real world, this is not a very realistic assumption as accurate dynamic models are difficult to obtain. Also, the dynamic model parameters might change over time due to various factors. Thus, there is a need for a learning-based lane change controller design methodology that can learn the optimal control policy in real time using sensor data. In this paper, we have addressed this need by introducing an optimal learning-based control methodology that can solve the real-time lane-changing problem of AVs, where the input-state data of the AV is utilized to generate a near-optimal lane-changing controller by approximate/adaptive dynamic programming (ADP) technique. In the case of this type of complex lane-changing maneuver, the lateral dynamics depend on the longitudinal velocity of the vehicle. If the longitudinal velocity is assumed constant, a linear parameter invariant model can be used. However, assuming constant velocity while performing a lane-changing maneuver is not a realistic assumption. This assumption might increase the risk of accidents, especially in the case of lane abortion when the surrounding vehicles are not cooperative. Thus, in this paper, the dynamics of the AV are assumed to be a linear parameter-varying system. Thus we have two challenges for the lane-changing controller design: parameter-varying, and unknown dynamics. With the help of both gain scheduling and ADP techniques combined, a learning-based control algorithm that can generate a near-optimal lane-changing controller without having to know the accurate dynamic model of the AV is proposed. The inclusion of a gain scheduling approach with ADP makes the controller applicable to non-linear and/or parameter-varying AV dynamics. The stability of the learning-based gain scheduling controller has also been rigorously proved. Moreover, a data-driven lane-changing decision-making algorithm is introduced that can make the AV perform a lane abortion if safety conditions are violated during a lane change. Finally, the proposed learning-based gain scheduling controller design algorithm and the lane-changing decision-making methodology are numerically validated using MATLAB, SUMO simulations, and the NGSIM dataset.

* Corresponding author.
E-mail addresses: sc8804@nyu.edu (S. Chakraborty), l.cui@nyu.edu (L. Cui), kaan.ozbay@nyu.edu (K. Ozbay), zjiang@nyu.edu (Z.-P. Jiang).

https://doi.org/10.1016/j.trb.2024.103026
Received 10 February 2023; Received in revised form 10 May 2024; Accepted 10 July 2024
Available online 1 August 2024

1. Introduction

Inappropriate lane changes are responsible for one-tenth of all accidents in the U.S. (Chovan et al., 1994), due to human drivers' inaccurate estimation and prediction of the surrounding traffic, illegal maneuver, and inefficient driving skills. 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 (Wang et al., 2019b). 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/or 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 Ma et al. (2022), Li (2022), Zhou et al. (2019). However, it remains challenging to precisely control the lateral movement of the AV, especially in the absence of an accurate model (Bevly et al., 2016).

1.1. Model-based techniques

Over the past few years, many decision-making and trajectory-planning methodologies have been proposed in the literature. Nilsson et al. (2016) 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 is determined to perform the maneuver. This methodology requires tuning of parameters for the utility functions that can affect lane change decision-making. Wang et al. (2021a) proposed a real-time dynamic cooperative lane-changing model for connected and autonomous vehicles (CAVs) with possible accelerations of a preceding vehicle. The lane change decision is based on the upper and lower bounds of the acceleration of the preceding and following vehicles in the target lane. Here, the acceleration bounds are derived using a simple kinematic model of the AV that might not be accurate. Luo et al. (2016) and Xu et al. (2019) proposed a constrained optimization-based lane-changing methodology. The objective function is minimized for the longitudinal jerk, lateral jerk, 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 Luo et al. (2016), and requires the knowledge of the dimension of the surrounding vehicles in the formulation given by Xu et al. (2019). A more practical scenario is considered by Wang et al. (2021b, 2020b) 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. Nie et al. (2016) 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 trajectory generation and trajectory tracking techniques have been proposed in the past to maneuver the AV to the desired lane while ensuring safety, see Nilsson et al. (2016)–Xu et al. (2012). In Wang et al. (2020b), the authors have implemented a model predictive control (MPC) based trajectory-tracking controller. Nilsson et al. (2016) used quadratic programming (QP) to compute the trajectories for lateral and longitudinal maneuvers, where a double integrator model is used for the AV dynamics. Wang et al. (2020b, 2021b) used an improved sine function-based trajectory generation and then used MPC to track the longitudinal and lateral trajectories that adopt the two-wheel kinematic vehicle model. Luo et al. (2016) used a quintic polynomial to generate the longitudinal and lateral trajectories considering safety, comfort, and traffic efficiency. Then a trajectory-tracking sliding mode control is proposed to track the trajectory. Nilsson et al. (2014) used a hierarchical, two-level architecture for the trajectory generation and vehicle control of AV. The high-level planner uses QP to generate the trajectory using a low-fidelity point-mass model and linear collision avoidance constraints and MPC is used in the lower level to execute the trajectory that uses a high-fidelity vehicle model. Suh et al. (2018) used a hyperbolic tangent function to generate trajectories and a stochastic MPC with a linear parameter-varying (LPV) vehicle model is used for vehicle control. Zhang et al. (2021) presents a hierarchical multi-layer trajectory planning framework that enables real-time collision avoidance under complex driving conditions. The upper-layer controller generates a reference quintic polynomial trajectory, while the middle-layer controller generates a QP-based trajectory cluster with different time stamps. Zhang et al. (2022) introduces a framework for autonomous emergency avoidance in complex driving situations, using driving primitive transitions and motion control. It employs quintic polynomial-based path planning and combines linear time-varying MPC with direct yaw-moment control for vehicle stability and path tracking. The scheme's efficacy is verified through extensive Hardware-in-Loop tests.

As evident from the literature, most of the works done to solve the lane change problem of AVs are model-based techniques. One major limitation of these model-based approaches is that the performance of the automated lane-changing highly depends on the accuracy of the AVs' model, and the inaccurate model may deteriorate the lane-changing. 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.

1.2. Purely data-driven machine learning methods

In recent years, many researchers have used machine learning (ML) techniques for lane change decision-making and policy learning. Hoel et al. (2019) proposed a framework for decision-making that integrates the principles of planning and learning, utilizing both Monte Carlo tree search and deep reinforcement learning (DRL). A similar approach is used by Hoel et al. (2018), where a deep Q-network agent was trained to handle speed and lane change decisions. Mirchevska et al. (2018) proposed a safe reinforcement learning (RL) method for lane change decision-making. Li et al. (2022a) proposed a transformer network-based lane change decision-making method combined with a DRL architecture for computing safe lane change policies. Xie et al. (2019) proposed a lane-changing model using deep learning, where the lane-change decision-making uses a deep belief network (DBF) and the trajectory generation uses a long-short-term memory (LSTM) network. Xu et al. (2017) utilized the LSTM structure to process current monocular camera views and prior vehicle conditions. Their network was subsequently trained to mimic actual driving behaviors using a vast video dataset. More relevant works for learning-based lane change decision making can be found in Ye et al. (2019), Nageshrao et al. (2019), Ye et al. (2021), and references therein.

Paxton et al. (2017) integrates Monte Carlo tree search with neural networks to generate safe and responsive motion plans. Min et al. (2018, 2019) proposed an autonomous driving framework that leverages traditional driver assistance systems (DAS) combined with DRL. The DRL agent acts as a supervisor to identify the DAS functions, including lane changes, cruise control, and lane maintenance, to optimize average speed and maximize overtakes with minimal lane shifts. The trained DRL agent can directly translate both camera imagery and LIDAR information into an action strategy. Kuderer et al. (2015) proposed an inverse reinforcement learning (IRL) method to learn driving styles from demonstrations. Then, the learned model is used to compute trajectories online during autonomous driving tasks. End-to-end learning for autonomous driving is explored by many researchers in the literature (see Bojarski et al., 2016 Codevilla et al., 2018, Hecker et al., 2018). The main aim of end-to-end learning is to directly optimize the entire driving pipeline from sensor data to control commands as a one machine learning tasks. Folkers et al. (2019) proposed a DRL-based control method for autonomous vehicles. A neural network agent is trained to map its perceived state and produce acceleration and steering commands with the goal of attaining a particular target state, taking into account any identified obstacles. The trained agent is then tested in simulations and applied to a real research vehicle. Wang et al. (2018) proposed a Q-learning-based approach to learn a policy for the autonomous vehicle for lane changing. Li et al. (2022c) proposed DRL algorithms combined with risk assessment functions to find an optimal driving strategy with the minimum expected risk. The proposed algorithms generate a series of actions to minimize the driving risk and prevent the host vehicle from collisions. More recent works can be found in the survey papers (Kiran et al., 2021; Farazi et al., 2021; Zhu and Zhao, 2021).

Machine learning (ML) techniques hold substantial promise in the realm of autonomous driving. While these methodologies are progressing, there remains room for enhancement to further their efficiency. A notable challenge in ML-based approaches is their dependency on extensive and varied datasets, a point underscored by several survey papers (Tampuu et al., 2020, Grigorescu et al., 2020, Kiran et al., 2021). Additionally, convergence may not always be attained during training (Tampuu et al., 2020, Neal et al., 2018, Codevilla et al., 2019). Furthermore, designing a task-specific neural network architecture may require a lot of trial-and-error in ML-based methods owing to their empirical nature (Zhu and Zhao, 2021). Finally, as it can be seen in Wang et al. (2019a), Li et al. (2022b), Tang et al. (2022), converging to an optimal policy for DRL-based methods takes a long period of training. This alone might effect real-time applicability and decision making, specially in a safety-critical scenario. This reliance on empirical methods prompts a careful consideration of these techniques' adaptability in diverse operational contexts.

1.3. Non-model based learning methods

In consideration of the above discussion, this work attempts to address some of the existing aforementioned challenges for ML-based methods. Researchers in the control community have leveraged concepts from reinforcement learning (RL) (Sutton and Barto, 2018) and adaptive/approximate dynamic programming (ADP) (refer to Bertsekas, 2012; Bellman, 1966; Lewis et al., 2012; Powell, 2007) to formulate data-driven adaptive optimal control strategies for tackling the stabilization and tracking challenges inherent in dynamical systems (see Jiang and Jiang, 2012; Vrabie et al., 2009; Vamvoudakis et al., 2020; Chakraborty et al., 2022; Jiang and Jiang, 2014a,b; Lewis and Vrabie, 2009; Gao and Jiang, 2016; Gao et al., 2019, 2015). In this paper, we adopt a similar approach in developing an intelligent and safe lane change maneuver algorithm for AVs in the mixed traffic scenario by considering the structural information of the AV system. The proposed methodology learns an optimal lane-changing policy with comparatively smaller number of data samples and lesser learning times. Also, we provide formal guarantees for stability, convergence and uniqueness of the learned policies. Our learning-based methodology can 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 introduce a lane-changing decision-making algorithm that does not require solving an optimization problem, parameter tuning, and/or vehicle dimension information of nearby vehicles. One major advantage of the ADP-based approach, as opposed to traditional reinforcement learning (Sutton and Barto, 2018), lies in the fact that the closed-loop stability of the dynamical system is established when the learned control policy is implemented. Meanwhile, the stability/robustness of the AV controller characterizes the convergence of the AV's dynamics to a desired equilibrium. Our approach could offer a more robust solution for safety-critical systems like for autonomous driving.

Noting the fact that maintaining a constant velocity during lane change is impractical, and the lateral dynamics of AV depend on the longitudinal velocity, we assume the dynamics of AV is LPV instead of linear time-invariant (LTI). Thus, one cannot directly apply the LTI data-driven controller techniques developed by Jiang and Jiang (2012), Gao and Jiang (2016). In this work, we aim

to extend the results of Jiang and Jiang (2012) for LPV systems. Many authors in the literature have proposed control techniques for LPV systems, some of them are summarized in Rugh and Shamma (2000), Hoffmann and Werner (2014). Among other methods, gain scheduling is more suitable for the control of LPV systems if the time-varying parameter varies slowly. The authors of Shamma (1988), Shamma and Athans (1990) introduced 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 (Bianchi et al., 2004), missile autopilot (Yuan et al., 2016; Shamma and Cloutier, 1993), flight control (Saussié et al., 2011), cloud computing (Saikrishna et al., 2016), Wang et al. (2020a) and more recently for AV control (Zhu et al., 2019; Alcala et al., 2018; Kapsalis et al., 2020; Zhang et al., 2014; Chu et al., 2022). However, these methods are model-based and suffer from similar drawbacks mentioned before. In this work, we propose a learning-based gain scheduling technique.

A preliminary version of this work can be found in our conference paper (Chakraborty et al., 2022). The difference between the preliminary work and the present work is that the present work provides a 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 and compares it with model-based MPC. Furthermore, the present work conducts many simulation studies to analyze the effectiveness of the proposed methodology. In order to keep our study simple, we have considered one AV and four surrounding HDVs. This represents a simplified mixed traffic scenario which has been frequently considered in the transportation literature (see Suo et al., 2024; Jin et al., 2018; Gao et al., 2016). The main contributions of this paper are summarized as follows:

1. Introduced a learning-based optimal control design technique for lane-changing of AVs that:
   - Uses only the state and input information.
   - Guarantees algorithmic convergence, vehicle stability, and is data efficient.
2. A data-driven lane-changing decision-making algorithm that incorporates the following:
   - Compared with the existing literature that requires the generation of trajectory for lane change, our methodology directly tracks a target point defined in the target lane, thus reducing the computational complexity of lane change.
3. 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.
4. Demonstrated the applicability of the proposed methodology in real-time learning and decision-making by SUMO implementation (Lopez et al., 2018) and NGSIM dataset (NGSIM, 2016).

The remainder of the paper is organized as follows. The AV dynamic model and problem formulation are given in Section 2. The lane change decision-making algorithm is discussed in Section 3. The learning-based gain scheduling algorithm is developed in Section 4. The algorithmic details are given in Section 5. Simulation results are given in Section 6. Finally, conclusions are given in Section 7.

Notations: Throughout this paper, $\mathbb{Z}_+$ denotes the set of non-negative integers, $\mathbb{C}_-$, denotes the complex left half-plane, $\|.\|$ represents the Euclidean norm for vectors, and the induced norm of matrices, $\sigma(\mathbf{W})$ is the complex spectrum of $\mathbf{W}$, $\otimes$ indicates the Kronecker product and $\text{vec}(\mathbf{T}) = \left[t_1^T, t_2^T, \dots, t_m^T\right]^T$ with $t_i \in \mathbb{R}^r$ being the columns of $\mathbf{T} \in \mathbb{R}^{r\times m}$. For a symmetric matrix $\mathbf{P} \in \mathbb{R}^{m\times m}$, $\text{vecs}(\mathbf{P}) = [p_{11}, 2p_{12}, \dots, 2p_{1m}, p_{22}, 2p_{23}, \dots, 2p_{(m-1)m}, p_{mm}]^T \in \mathbb{R}^{(1/2)m(m+1)}$, for a column vector $v \in \mathbb{R}^n$, $\text{vecv}(v) = [v_1^2, v_1v_2, \dots, v_1v_n, v_2^2, v_2v_3, \dots, v_{n-1}v_n, v_n^2]^T \in \mathbb{R}^{(1/2)n(n+1)}$. $\mathbf{I}_n(\mathbf{0}_n)$ is the identity (zero) matrix of dimension n. A real analytic function is an infinitely differentiable function such that the Taylor series at any point x_0 in its domain given as $T(x) = \sum_{n=0}^{\infty} \frac{f^n(x_0)}{n!}(x - x_0)^n = f(x_0)+\mathcal{O}(\epsilon)$, where $\mathcal{O}(\epsilon)$ contains the higher order terms in the expansion, converges to $f(x)$ for x in a neighborhood of x_0 pointwise.

2. AV dynamic model and problem formulation

This section explains the AV's states and inputs assumed to design the learning-based controller.

2.1. Longitudinal dynamic model

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

$$\dot{\mathbf{x}}_{lo} = \mathbf{A}_{lo}\mathbf{x}_{lo} + \mathbf{B}_{lo}u_{lo}, \tag{1}$$

where, $\mathbf{A}_{lo} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $\mathbf{B}_{lo} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$, m = mass of the vehicle, and $u_{lo}(t)$ is the driven force. The state vector $\mathbf{x}_{lo} = [x_1(t), x_2(t)]^T$, where $x_1(t) = x_{AV}(t)$ denotes the longitudinal position, and $x_2(t) = V_x(t)$ denotes the longitudinal velocity.

2.2. Lateral dynamic model

Definition 2.1. A point (T_x, T_y) that is placed at a safe distance from the leading vehicle is defined as the target point (TP).

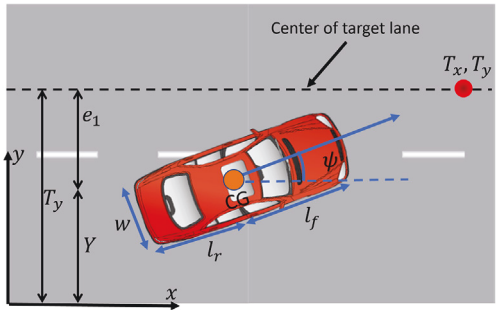


Fig. 1. Position and orientation error of AV.

The lateral dynamic model considered here is based on position and orientation error as shown in Fig. 1. Let (T_x, T_y) be the coordinates of the target point, $\psi(t)$ be the orientation of the vehicle, $e_1(t)$ is the distance of the center of gravity (C.G.) of the vehicle from the center line of the lane, and $e_2(t)$ be the orientation error of the vehicle with respect to the road.

Assumption 2.2. Vehicles travel on a straight road with radius $R = \infty$.

The dynamic model is given as:

$$\dot{\mathbf{x}}_{la} = \mathbf{A}_{la}\mathbf{x}_{la} + \mathbf{B}_{la}u_{la}, \tag{2}$$

where $u_{la}(t)$ denotes the front wheel steering angle, $\mathbf{x}_{la} = [e_1(t), \dot{e}_1(t), e_2(t), \dot{e}_2(t)]^T$,

$$\mathbf{A}_{la} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{af}+2C_{ar}}{mV_x} & \frac{2C_{af}+2C_{ar}}{m} & \frac{-2C_{af}l_f+2C_{ar}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{af}l_f-2C_{ar}l_r}{I_zV_x} & \frac{2C_{af}l_f-2C_{ar}l_r}{I_z} & -\frac{2C_{af}l_f{}^2+2C_{ar}l_r{}^2}{I_zV_x} \end{bmatrix}, \mathbf{B}_{la} = \begin{bmatrix} 0 \\ \frac{2C_{af}}{m} \\ 0 \\ \frac{2C_{af}l_f}{I_z}. \end{bmatrix} \tag{3}$$

In (3), $C_{\alpha f}$ is the cornering stiffness of each front tire, C_{ar} is the cornering stiffness of each rear tire, l_f is the front length of the vehicle from the center of gravity, l_r is the rear length of the vehicle from the center of gravity, m is the mass of the vehicle, I_z is the z moment of inertia, and V_x is the longitudinal velocity of the vehicle. More details on the model can be found in Rajamani (2011). The controllability of system (2) is shown in the following lemma.

Lemma 2.3. *System (2) is controllable if and only if* $V_x^2 \neq \frac{2C_{ar}(l_f+l_r)(ml_fl_r-I_z)}{m^2l_f^2}$.

Proof. The controllability matrix of the system is

$$C = [\mathbf{B}_{la}, \mathbf{A}_{la}\mathbf{B}_{la}, \mathbf{A}_{la}^2\mathbf{B}_{la}, \mathbf{A}_{la}^3\mathbf{B}_{la}].$$

It is checkable by Matlab symbolic toolbox that

$$\det(C) = \frac{64C_{af}^4C_{ar}^2(l_f+l_r)^2[m^2l_f^2V_x^2 - 2C_{ar}(l_f+l_r)(ml_fl_r-I_z)]}{I_z^4m^4V_x^2}.$$

According to Chen (1999, Theorem 6.1), the system is controllable if and only if $\det(C) \neq 0$. Therefore, the lateral dynamics is controllable if and only if

$$V_x^2 \neq \frac{2C_{ar}(l_f+l_r)(ml_fl_r-I_z)}{m^2l_f^2}.$$

Since the vehicle travels on a straight road, the desired orientation of the vehicle is considered as $\psi_{des} = 0$. Then from Fig. 1, the lateral position $(Y(t))$ and yaw angle $(\psi(t))$ can be obtained as:

$$\begin{aligned} Y(t) &= T_y - e_1(t), \\ \psi(t) &= e_2(t). \end{aligned} \tag{4}$$

Remark 2.4. Note that the longitudinal velocity V_x appears in the lateral dynamics of the AV. This makes the lateral model parameter-varying. Thus, an LTI controller design is not sufficient to stabilize the AV lateral dynamics. Thus, we use gain scheduling to guarantee the stability of the AV lateral dynamics. This is discussed in Section 4.

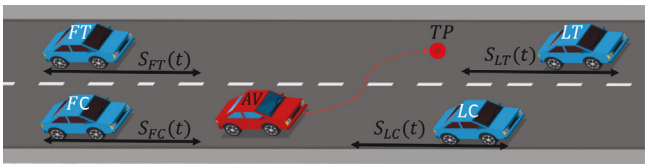


Fig. 2. A typical lane change scenario. The blue vehicles are human-driven vehicles (HDVs) and the red vehicle is AV. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.3. Problem definition

Given that we have the access to the input and state information and the target point (TP) in the target lane, the following problem is addressed in this work:

Problem 2.5. In the absence of the accurate dynamics of the AV and utilizing the collected input-state data, design a learning-based lane-changing algorithm that incorporates the following:

1. a model-free optimal controller for the AV's lateral maneuver such that $e_1(t) \to 0$ and $e_2(t) \to 0$;
2. a model-free optimal longitudinal controller to keep a safe distance from the preceding vehicle;
3. a lane-changing decision-making algorithm that can ensure safe lane-changing during non-cooperative behavior of surrounding vehicles.

3. 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, and

$$S_i(t) = L + hv_i(t) + d + w, \quad i \in \{LT, FT, LC, FC\}, \tag{5}$$

is the safety distance, where h is the headway time, L is the length of vehicle, v_i is the velocity of the ith vehicle, d is the standstill distance, and w is the width of the AV, and t is the time step. The width w is added to $S_i(t)$ to ensure safety when the AV travels a diagonal distance. In this work, the lane-changing decision-making is proposed for a single lane change maneuver. As shown in Fig. 2, five vehicles are involved in a lane change maneuver. The AV performs a maneuver to change the lane and places itself in the target point (TP) with coordinates (T_x, T_y). Let, $x_{AV}(t)$, $x_{LT}(t)$, $x_{FT}(t)$, $x_{LC}(t)$, $x_{FC}(t)$ 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.

$$x_{AV}(t) \le x_{LC}(t) - S_{LC}(t), \tag{6}$$

$$x_{AV}(t) \ge x_{FC}(t) + S_{FC}(t), \tag{7}$$

$$x_{AV}(t) \le x_{LT}(t) - S_{LT}(t), \tag{8}$$

$$x_{AV}(t) \ge x_{FT}(t) + S_{FT}(t). \tag{9}$$

Thus, if the inequalities in (6)–(9) are satisfied, a safe lane change maneuver is possible. The safe distances $S_i(t)$ are evaluated continuously.

The lane change algorithm starts by collecting the position data of the vehicles. Then, the safety inequalities in (6)–(9) are checked. If all the inequalities satisfy and the AV is not in the target lane, the lane change is initiated by setting the target point at a safe distance from the LT. At all times during the lane change maneuver the safety inequalities are checked. If the safety is checked to be true, then the AV continues to change lane until it has reached the target lane. If the safety is checked to be false, the target point is changed to the current lane to abort lane changing and is placed at a safe distance from LC and the process starts over again by checking the inequalities in (6)–(9). During lane change, the x-coordinate of the target point is computed as $T_x(t) = x_{LT}(t) - S_{LT}(t)$ and the y-coordinate (T_y) is set as the coordinate of the center line of the target lane. During lane abortion, the x-coordinate of the target point is computed as $T_x(t) = x_{LC}(t) - S_{LC}(t)$ and the y-coordinate (T_y) is set as the coordinate of the center line of the current lane.

Assumption 3.1. The surrounding vehicles are human-driven vehicles (HDVs) that are non-cooperative with the AV except for lane-change abortion when the vehicles in the current lane are assumed to yield to the AV.

4. Learning-based gain scheduling

4.1. Preliminary results

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 guarantees stability and certain performance objectives; link these controllers together in order to obtain a single controller for the entire range of the system operation (Shahruz and Behtash, 1992).

Consider the following LPV system:

$$\dot{\mathbf{x}} = \mathbf{A}(\alpha)\mathbf{x}(t) + \mathbf{B}(\alpha)\mathbf{u}(t), \tag{10}$$

$$\alpha = \alpha(t), \tag{11}$$

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

Assumption 4.1. For all $\alpha \in I$, the matrix $\mathbf{B}(\alpha)$ is full column rank.

Assumption 4.2. The elements of the system matrices $\mathbf{A}(\alpha)$ and $\mathbf{B}(\alpha)$ are analytic functions of α.

Assumption 4.3. The parameter α is a continuous and bounded function of time, differentiable almost everywhere with bounded derivative, and is measured for all time $t \geq 0$.

Assumption 4.4. All states are available for feedback, and the system in (10) is stabilizable for all $\alpha \in I$.

In this section, we assume no knowledge of the system matrices $\mathbf{A}(\alpha)$, and $\mathbf{B}(\alpha)$, and try to design a feedback control law of the form:

$$\mathbf{u}(t) = -\mathbf{K}(\alpha)\mathbf{x}(t), \tag{12}$$

$$\alpha = \alpha(t), \tag{13}$$

where, $\mathbf{K}(\alpha) \in \mathbb{R}^{m\times n}$ is the state feedback gain matrix. To design the state feedback control law in (12), we first select finite number of fixed $\alpha_l \in I$. Let $\mathbf{K}(\alpha_l)$ and $\mathbf{K}(\alpha_{l+1})$, respectively, denote the gain matrices computed at the adjacent points α_l and α_{l+1} in I. At each $\alpha \in [\alpha_l, \alpha_{l+1}]$, the gain $\mathbf{K}(\alpha)$ in (12) is obtained as the linear interpolation between $\mathbf{K}(\alpha_l)$ and $\mathbf{K}(\alpha_{l+1})$ given as (Shahruz and Behtash (1992)):

$$\mathbf{K}(\alpha) = \mathbf{K}(\alpha_l) + \frac{\mathbf{K}(\alpha_{l+1}) - \mathbf{K}(\alpha_l)}{\alpha_{l+1} - \alpha_l}(\alpha - \alpha_l). \tag{14}$$

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

- For each α_l the state feedback gain matrix $\mathbf{K}(\alpha_l)$ is computed such that the closed-loop stability of the frozen system $\mathbf{A}_c(\alpha_l) = \mathbf{A}(\alpha_l) - \mathbf{B}(\alpha_l)\mathbf{K}(\alpha_l)$ along with a minimum cost of operating the system is guaranteed.
- At each $\alpha \in [\alpha_l, \alpha_{l+1}]$, the gain $\mathbf{K}(\alpha)$ obtained using (14) guarantees the stability of the closed-loop system.

Note that, in this work $\alpha(t) = V_x(t) = x_2(t)$, and $\dot{\alpha}(t) = \dot{V}_x(t) = u_{lo}/m$ which is the longitudinal acceleration. Next, we present a learning-based control methodology to obtain the optimal control gain $\mathbf{K}(\alpha_l)$ for fixed $\alpha_l \in I$. In order to reduce the state deviations and control effort, we seek to design a linear optimal control law of the form given in (12) for a fixed $\alpha_l \in I$ that can minimize the following cost function:

$$\min_{\mathbf{u}} \quad J = \int_0^\infty (\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u})d\tau, \tag{15}$$

where, $\mathbf{Q} = \mathbf{Q}^T \geq 0$, $\mathbf{R} = \mathbf{R}^T > 0$, with $(\mathbf{A}(\alpha_l), \mathbf{Q}^{1/2})$ being observable.

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

$$\mathbf{A}(\alpha_l)^T\mathbf{P}(\alpha_l) + \mathbf{P}(\alpha_l)\mathbf{A}(\alpha_l) + \mathbf{Q} - \mathbf{P}(\alpha_l)\mathbf{B}(\alpha_l)\mathbf{R}^{-1}\mathbf{B}(\alpha_l)^T\mathbf{P}(\alpha_l) = \mathbf{0}, \tag{16}$$

$$\mathbf{K}^*(\alpha_l) = \mathbf{R}^{-1}\mathbf{B}(\alpha_l)^T\mathbf{P}(\alpha_l)^*, \tag{17}$$

where (16) is the well-known algebraic Riccati equation and $\mathbf{P}(\alpha_l)^* = \mathbf{P}(\alpha_l)^{*T} > 0$ is the unique solution of (16). Since, the Riccati equation is non-linear in $\mathbf{P}(\alpha_l)$, it is difficult to solve for large dimensional systems. In the literature, many efficient iterative approaches have been proposed to solve (16). One such approach is given by Kleinman (1968), and is reproduced below for the sake of completeness:

Theorem 4.5. *Let $\mathbf{K}_0(\alpha_l)$ be any stabilizing feedback gain matrix, and $\mathbf{P}_k(\alpha_l)$ is the symmetric positive definite solution of the Lyapunov Eq. (18), then for $k = 0, 1, 2, \ldots$, we have*

- *Policy evaluation step:*

$$(\mathbf{A}(\alpha_l) - \mathbf{B}(\alpha_l)\mathbf{K}_k(\alpha_l))^T\mathbf{P}_k(\alpha_l) + \mathbf{P}_k(\alpha_l)(\mathbf{A}(\alpha_l) - \mathbf{B}(\alpha_l)\mathbf{K}_k(\alpha_l)) + \mathbf{Q} + \mathbf{K}_k(\alpha_l)^T\mathbf{R}\mathbf{K}_k(\alpha_l) = \mathbf{0}, \tag{18}$$

- *Policy update step: Update the gain matrix as*

$$\mathbf{K}_{k+1}(\alpha_l) = \mathbf{R}^{-1}\mathbf{B}(\alpha_l)^T\mathbf{P}_k(\alpha_l), \tag{19}$$

Then, the following properties hold:

1. $\mathbf{A}(\alpha_l) - \mathbf{B}(\alpha_l)\mathbf{K}_k(\alpha_l)$ *is stable,*
2. $\mathbf{P}^*(\alpha_l) \le \mathbf{P}_{k+1}(\alpha_l) \le \mathbf{P}_k(\alpha_l)$,
3. $\lim_{k\to\infty} \mathbf{K}_k(\alpha_l) = \mathbf{K}^*(\alpha_l)$, $\lim_{k\to\infty} \mathbf{P}_k(\alpha_l) = \mathbf{P}^*(\alpha_l)$.

Note that (18) is linear in $\mathbf{P}_k(\alpha_l)$. Thus, one can iteratively solve (18) and update $\mathbf{K}_{k+1}(\alpha_l) = \mathbf{R}^{-1}\mathbf{B}(\alpha_l)^T\mathbf{P}_k(\alpha_l)$ to numerically approximate the solution to (16). But, this assumes the complete knowledge of the system matrices $\mathbf{A}(\alpha_l)$ and $\mathbf{B}(\alpha_l)$.

Now, consider the modified system equation as follows:

$$\dot{\mathbf{x}} = \mathbf{A}_k(\alpha_l)\mathbf{x} + \mathbf{B}(\alpha_l)(\mathbf{K}_k(\alpha_l)\mathbf{x} + \mathbf{u}), \tag{20}$$

where $\mathbf{A}_k(\alpha_l) = \mathbf{A}(\alpha_l) - \mathbf{B}(\alpha_l)\mathbf{K}_k(\alpha_l)$. Then, along the solutions of (20) using (18) and (19), we have:

$$\begin{aligned}
&\mathbf{x}(t+\delta t)^T\mathbf{P}_k(\alpha_l)\mathbf{x}(t+\delta t) - \mathbf{x}(t)^T\mathbf{P}_k(\alpha_l)\mathbf{x}(t)\\
&= \int_t^{t+\delta t}\left[\mathbf{x}^T(\mathbf{A}_k(\alpha_l)^T\mathbf{P}_k(\alpha_l) + \mathbf{P}_k(\alpha_l)\mathbf{A}_k(\alpha_l))\mathbf{x} + 2(\mathbf{u} + \mathbf{K}_k(\alpha_l)\mathbf{x})^T\mathbf{B}(\alpha_l)^T\mathbf{P}_k(\alpha_l)\mathbf{x}\right] d\tau,\\
&= 2\int_t^{t+\delta t}(\mathbf{u} + \mathbf{K}_k(\alpha_l)\mathbf{x})^T\mathbf{R}\mathbf{K}_{k+1}(\alpha_l)\mathbf{x}\, d\tau - \int_t^{t+\delta t}\mathbf{x}^T\mathbf{Q}_k(\alpha_l)\mathbf{x}\, d\tau,
\end{aligned} \tag{21}$$

where, $\mathbf{Q}_k(\alpha_l) = \mathbf{Q} + \mathbf{K}_k(\alpha_l)^T\mathbf{R}\mathbf{K}_k(\alpha_l)$. It must be noted that the last equation of (21) is independent of the system matrices $\mathbf{A}(\alpha_l)$ and $\mathbf{B}(\alpha_l)$.

Lemma 4.6. *Consider the matrices $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ with compatible dimensions. Then the vectorization of the matrix product is given as:*

$$\mathrm{vec}(\mathbf{XYZ}) = \left(\mathbf{Z}^T \otimes \mathbf{X}\right)\mathrm{vec}\left(\mathbf{Y}\right). \tag{22}$$

Using Lemma 4.6, the terms in (21) can be written as follows:

$$\mathbf{x}^T\mathbf{P}_k(\alpha_l)\mathbf{x} = \left(\mathbf{x}^T \otimes \mathbf{x}^T\right)\mathrm{vec}\left(\mathbf{P}_k(\alpha_l)\right), \tag{23}$$

$$\mathbf{x}^T\mathbf{Q}_k(\alpha_l)\mathbf{x} = \left(\mathbf{x}^T \otimes \mathbf{x}^T\right)\mathrm{vec}\left(\mathbf{Q}_k(\alpha_l)\right), \tag{24}$$

$$\left(\mathbf{u} + \mathbf{K}_k(\alpha_l)\mathbf{x}\right)^T\mathbf{R}\mathbf{K}_{k+1}(\alpha_l)\mathbf{x} = \left[\left(\mathbf{x}^T \otimes \mathbf{x}^T\right)\left(\mathbf{I}_n \otimes \mathbf{K}_k(\alpha_l)^T\mathbf{R}\right) + \left(\mathbf{x}^T \otimes \mathbf{u}^T\right)\left(\mathbf{I}_n \otimes \mathbf{R}\right)\right]\mathrm{vec}(\mathbf{K}_{k+1}(\alpha_l)). \tag{25}$$

For any positive integer l, define $\boldsymbol{\Delta}_{xx} \in \mathbb{R}^{l\times\frac{1}{2}n(n+1)}$, $\mathbf{I}_{xx} \in \mathbb{R}^{l\times n^2}$, and $\mathbf{I}_{xu} \in \mathbb{R}^{l\times mn}$ as follows for $0 \le t_0 < t_1 < t_2 < \cdots < t_l$:

$$\boldsymbol{\Delta}_{xx} = \left[\mathrm{vecv}(\mathbf{x}(t_1)) - \mathrm{vecv}(\mathbf{x}(t_0)), \mathrm{vecv}(\mathbf{x}(t_2)) - \mathrm{vecv}(\mathbf{x}(t_1)), \ldots, \mathrm{vecv}(\mathbf{x}(t_l)) - \mathrm{vecv}(\mathbf{x}(t_{l-1}))\right]^T, \tag{26}$$

$$\mathbf{I}_{xx} = \left[\int_{t_0}^{t_1}\mathbf{x}\otimes\mathbf{x}\, d\tau, \int_{t_1}^{t_2}\mathbf{x}\otimes\mathbf{x}\, d\tau, \ldots, \int_{t_{l-1}}^{t_l}\mathbf{x}\otimes\mathbf{x}\, d\tau\right]^T, \tag{27}$$

$$\mathbf{I}_{xu} = \left[\int_{t_0}^{t_1}\mathbf{x}\otimes\mathbf{u}\, d\tau, \int_{t_1}^{t_2}\mathbf{x}\otimes\mathbf{u}\, d\tau, \ldots, \int_{t_{l-1}}^{t_l}\mathbf{x}\otimes\mathbf{u}\, d\tau\right]^T. \tag{28}$$

Using (23)–(28), (21) can be written as follows:

$$\boldsymbol{\Gamma}_k\begin{bmatrix}\mathrm{vecs}(\mathbf{P}_k(\alpha_l))\\ \mathrm{vec}(\mathbf{K}_{k+1}(\alpha_l))\end{bmatrix} = \boldsymbol{\Psi}_k, \tag{29}$$

where, $\boldsymbol{\Gamma}_k \in \mathbb{R}^{l\times(\frac{1}{2}n(n+1)+mn)}$, and $\boldsymbol{\Psi}_k \in \mathbb{R}^l$ are defined as follows:

$$\boldsymbol{\Gamma}_k = \left[\boldsymbol{\Delta}_{xx},\ -2\mathbf{I}_{xx}\left(\mathbf{I}_n \otimes \mathbf{K}_k(\alpha_l)^T\mathbf{R}\right) - 2\mathbf{I}_{xu}\left(\mathbf{I}_n \otimes \mathbf{R}\right)\right], \tag{30}$$

$$\boldsymbol{\Psi}_k = -\mathbf{I}_{xx}\mathrm{vec}(\mathbf{Q}_k(\alpha_l)). \tag{31}$$

Thus, given an initial stabilizing control input $\mathbf{u} = -\mathbf{K}_0(\alpha_l)\mathbf{x}$, the trajectories of the system can be recorded online in $\boldsymbol{\Delta}_{xx}, \mathbf{I}_{xx}, \mathbf{I}_{xu}$, which construct the data matrices $\boldsymbol{\Gamma}_k$ and $\boldsymbol{\Psi}_k$.

Assumption 4.7. There exists a sufficiently large integer $l_0 > 0$, such that for all $l \geq l_0$, the following holds:

$$\text{rank}([\mathbf{I}_{xx},\ \mathbf{I}_{xu}]) = \frac{n(n+1)}{2} + mn \tag{32}$$

Remark 4.8. Assumption 4.7 introduced above is inspired from the persistent excitation (PE) condition in adaptive control, which is necessary for the convergence of Algorithm 1.

Remark 4.9. It must be noted that, one only needs to collect data for learning the optimal controller till Assumption 4.7 is satisfied.

Theorem 4.10. *Under Assumption 4.7, there is a unique pair of matrices* $\left(\mathbf{P}_k(\alpha_l), \mathbf{K}_{k+1}(\alpha_l)\right)$, *with* $\mathbf{P}_k(\alpha_l) = \mathbf{P}_k(\alpha_l)^T\ \forall k \in \mathbb{Z}_+$, *that solves (29).*

Proof. see Jiang and Jiang (2017, 2012).

Theorem 4.11. *Given an initial stabilizing gain* $\mathbf{K}_0(\alpha_l) \in \mathbb{R}^{m\times n}$, *and under Assumption 4.7, the sequences* $\{\mathbf{P}_k(\alpha_l)\}_{k=0}^{\infty}$ *and* $\{\mathbf{K}_k(\alpha_l)\}_{k=0}^{\infty}$ *obtained by solving (29) converge to the optimal values* $\mathbf{P}^*(\alpha_l)$ *and* $\mathbf{K}^*(\alpha_l)$, *respectively.*

Proof. see Jiang and Jiang (2017, 2012).

Algorithm 1.

1. *With a stabilizing control policy* $\mathbf{K}_0(\alpha_l)$, *employ* $\mathbf{u} = -\mathbf{K}_0(\alpha_l)\mathbf{x} + \mathbf{e}$ *as the input on the time interval* $[t_0, t_l]$, *where* $\mathbf{e}$ *is the exploration signal. Compute* $\mathbf{\Delta}_{xx}, \mathbf{I}_{xx}, \mathbf{I}_{xu}$ *until Assumption 4.7 is satisfied. Let* $k = 0$.
2. *Solve* $\mathbf{P}_k(\alpha_l)$ *and* $\mathbf{K}_{k+1}(\alpha_l)$ *from (29).*
3. *Let* $k \leftarrow k+1$, *and repeat Step 2 until* $\|\mathbf{P}_k(\alpha_l) - \mathbf{P}_{k-1}(\alpha_l)\| \leq \epsilon_0$ *for* $k \geq 1$, *where the constant* $\epsilon_0 > 0$ *is a predefined small constant.*
4. *Use* $\mathbf{u} = -\mathbf{K}_k(\alpha_l)\mathbf{x}$ *as the approximated optimal control policy.*

Remark 4.12. In Algorithm 1, the exploration signal $\mathbf{e}$ is added to the initial stabilizing controller for data collection, such that the collected data set is persistently exciting to satisfy the full-rank condition in Assumption 4.7.

Hereafter, we denote the approximated optimal control gain $\mathbf{K}_k(\alpha_l)$ as $\hat{\mathbf{K}}^*(\alpha_l)$. We have established a learning-based optimal control framework for fixed $\alpha_l \in I$. The optimality and convergence guarantees of the optimal learning-based controller for a fixed $\alpha_l \in I$ are given by Theorems 4.10 and 4.11. It remains to show that the closed-loop system $\mathbf{A}_c(\alpha) = \mathbf{A}(\alpha) - \mathbf{B}(\alpha)\mathbf{K}(\alpha)$ is stable for the control gain $\mathbf{K}(\alpha)$ obtained using (14) for any fixed $\alpha \in [\alpha_l, \alpha_{l+1}]$. This is discussed in the next subsection.

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

Since, we can design an optimal learning-based controller for a fixed $\alpha_l \in I$, we call the spectrum $\sigma(\mathbf{A}_c(\alpha_l))$ optimal, where $\mathbf{A}_c(\alpha_l) = \mathbf{A}(\alpha_l) - \mathbf{B}(\alpha_l)\hat{\mathbf{K}}^*(\alpha_l)$. Suppose $\lambda_o^j \in \sigma(\mathbf{A}_c(\alpha_l))$ is an eigenvalue in the optimal spectrum $\sigma(\mathbf{A}_c(\alpha_l))$, and let $\lambda^j \in \sigma(\mathbf{A}_c(\alpha))$ be an eigenvalue in the spectrum $\sigma(\mathbf{A}_c(\alpha))$, where, $j = 1, 2, \ldots, n$. 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 neighborhood $\mathcal{N}_j(\epsilon)$ of $\sigma(\mathbf{A}_c(\alpha_l))$, i.e, $\lambda^j \in \mathcal{N}_j(\epsilon) := \{s \in \mathbb{C}_- : |s - \lambda_o^j| < \epsilon < 1\}$, $j = 1, \ldots, n$. The results in this section are based on the theory of eigenvalue perturbation (Lancaster and Tismenetsky, 1985; Horn and Johnson, 2012), and implicit function theorem (Markushevich, 2005).

Theorem 4.13 (*Implicit Function Theorem (Markushevich, 2005, Chapter 3).*) *Let* $F(z, w)$ *be a function of two variables which is analytic in a neighborhood of the point* (z_0, w_0), *and suppose the following conditions hold:*

- $F(z_0, w_0) = 0$,
- $\left.\frac{\partial F(z,w)}{\partial w}\right|_{(z_0,w_0)} \neq 0$.

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* $\mathcal{N}(w_0)$ *for any given* $z \in \mathcal{N}(z_0)$. *Moreover, the function* $w(z)$ *is single-valued and analytic on* $\mathcal{N}(z_0)$, *and satisfies the condition* $w(z_0) = w_0$.

The result of Theorem 4.14 is crucial to show $\lambda^j \in \mathcal{N}_j(\epsilon) := \{s \in \mathbb{C}_- : |s - \lambda_o^j| < \epsilon < 1\}$ for a sufficiently small ϵ.

Theorem 4.14. *Let* ϵ *denote the length of the gain-scheduling interval. Then for any* $\alpha \in [\alpha_l, \alpha_{l+1}]$, *we have* $\alpha = \alpha_l + c\epsilon$, *where* $0 \leq c \leq 1$. *Then, under Assumptions 4.2 and 4.4, there exists a sufficient small* ϵ, *such that the following relations hold.*

$$\hat{\mathbf{P}}^*(\alpha) = \hat{\mathbf{P}}^*(\alpha_l) + \mathcal{O}(\epsilon), \tag{33}$$

and

$$\hat{\mathbf{K}}^*(\alpha) = \hat{\mathbf{K}}^*(\alpha_l) + \mathcal{O}(\epsilon), \tag{34}$$

where $\hat{\mathbf{P}}^*(\alpha)$ *and* $\hat{\mathbf{P}}^*(\alpha_l)$ *are the approximate optimal solutions of the Riccati equation computed at* α *and* α_l*, respectively.* $\hat{\mathbf{K}}^*(\alpha)$ *and* $\hat{\mathbf{K}}^*(\alpha_l)$ *are the approximate optimal control gains computed at* α *and* α_l*, respectively.*

Proof. See Appendix A.

Next, using Theorem 4.14 we show that $\lambda^j \in \mathcal{N}_j(\epsilon) := \{s \in \mathbb{C}_- : |s - \lambda_o^j| < \epsilon < 1\}$ for a sufficiently small ϵ. Consider the following Theorem.

Theorem 4.15. *Under Assumptions 4.2 and 4.4, for sufficiently small* ϵ *the following holds:*

$$\lambda^j = \lambda_o^j + \mathcal{O}(\epsilon), j = 1, 2, \dots, n. \tag{35}$$

Proof. see Appendix B.

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

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 (10). From the results on slow-varying systems (Shamma, 1988, Vidyasagar, 2002), we have that if the rate of change of α is sufficiently small, then the stability of the LPV system (10) under the control action (12) can be deduced from the stability of the corresponding frozen systems. We try to obtain an upper bound on $\bar{\alpha} := \sup_{t\geq 0} |\dot{\alpha}(t)|$, such that the stability of closed-loop frozen systems implies the stability of (10) under the control law (12). Let J be the set of all frozen points α_i such that $\alpha_i < \alpha_j, i < j$. Lemma 4.16 gives a bound on the rate of change of the closed-loop LPV system, such that the stability of the closed-loop LPV system can be deduced from that of the corresponding frozen systems.

Lemma 4.16 (*Shamma (1988)*)**.** *Consider the following closed-loop LPV system:*

$$\dot{\mathbf{x}} = \mathbf{A}_c(\alpha(t))\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0 \tag{36}$$

From Theorem 4.15, we have that at a fixed time $\tau \geq 0$, $\sigma(\mathbf{A}_c(\alpha(\tau))) \subset \mathbb{C}_-$*. Thus, there exist constants* $\zeta \geq 1$, $\gamma \geq 0$, $\eta \in (0, \gamma]$ *such that for all* $t \geq 0$ *and* $\tau \geq 0$ *the followings holds:*

$$\|e^{\mathbf{A}_c(\tau)t}\|_2 \leq \zeta e^{-\gamma t}, \tag{37}$$

$$\|\dot{\mathbf{A}}_c(\alpha(t))\|_2 \leq \frac{(\gamma - \eta)^2}{4\zeta \ln(\zeta)}. \tag{38}$$

Then for all $t \geq 0$ *and any* $\mathbf{x}_0 \in \mathbb{R}^n$*,*

$$\|\mathbf{x}(t)\|_2 \leq \zeta e^{-\eta t}\|\mathbf{x}_0\|_2. \tag{39}$$

Next, using Lemma 4.16, an upper bound on the rate of change of α was obtained in Shahruz and Behtash (1992) which is reproduced in Theorem 4.17.

Theorem 4.17. *Consider the LPV system given in (10), and the control law (12). Let for any point* $\alpha \notin J$*, the control gain* $\mathbf{K}(\alpha)$ *be computed using (14). Then, under Assumptions 4.2 and 4.3, if*

$$\bar{\alpha} := \sup_{t\geq 0} |\dot{\alpha}(t)| \leq \frac{(\gamma - \eta)^2}{4\beta\zeta \ln\zeta}, \tag{40}$$

where,

$$\beta := \sqrt{n}\left(\max_{\alpha\in I} \left\| \frac{\partial \mathbf{A}(\alpha)}{\partial \alpha} \right\|_\infty + \max_{\alpha\in I} \left\| \frac{\partial \mathbf{B}(\alpha)}{\partial \alpha} \right\|_\infty \max_{1\leq i\leq m} \max_{\alpha_l \in J} \sum_{j=1}^{n} |k_{ij}(\alpha_l)| + \right.$$
$$\left. \max_{\alpha\in I} \|\mathbf{B}(\alpha)\|_\infty \max_{1\leq i\leq m} \max_{\alpha_l \in J} \sum_{j=1}^{n} \left| \frac{k_{ij}(\alpha_{l+1}) - k_{ij}(\alpha_l)}{\alpha_{l+1} - \alpha_l} \right| \right),$$

$\eta \in (0, \gamma]$ *and* k_{ij} *is the* ij*th component of* $\hat{\mathbf{K}}^*(\alpha_i)$*, then the system (10) with the control law (12) is uniformly exponentially stable.*

5. Algorithmic details

In this section, we outline the details of the algorithmic implementation of the proposed methodology for the lane changing problem of AV.

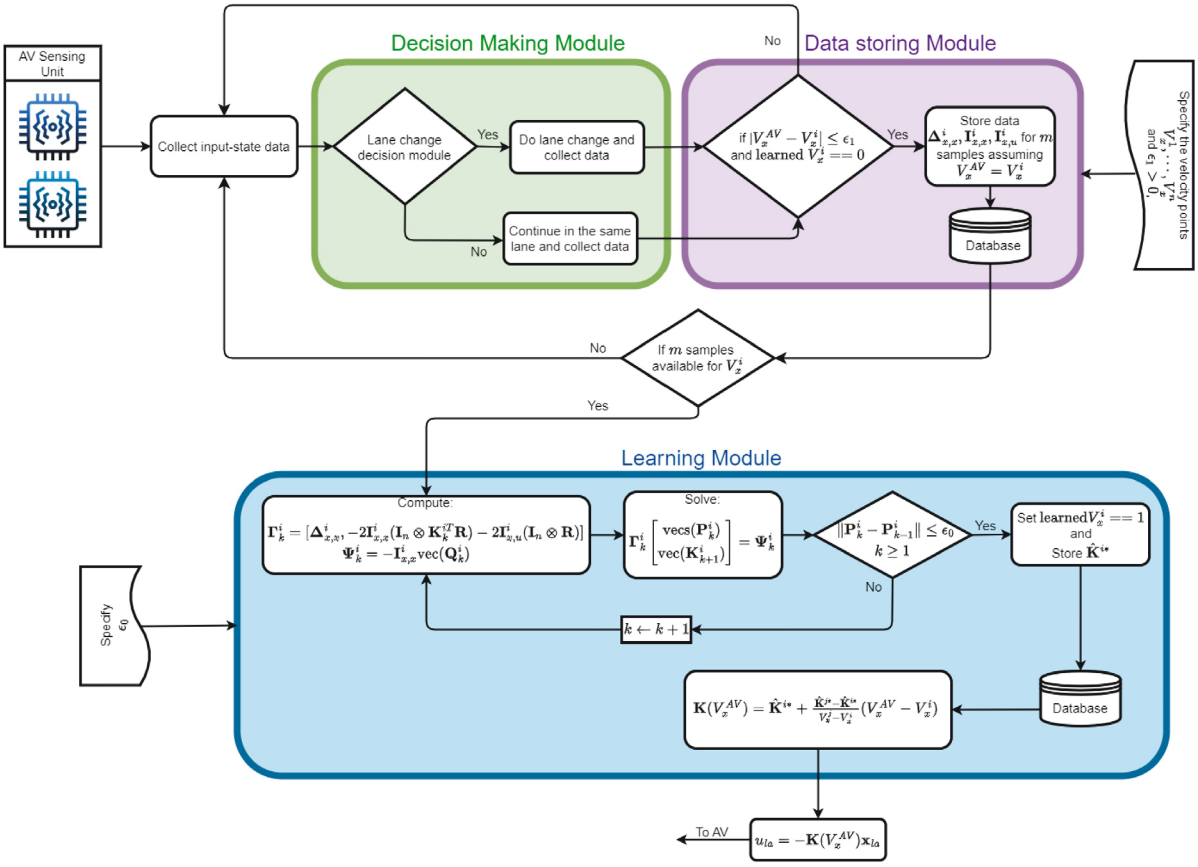


Fig. 3. Proposed learning-based gain scheduling algorithm.

5.1. Algorithmic details for learning optimal controller for lateral dynamics

The algorithm illustrated in Fig. 3 contains three principal modules: (1) Decision Making Module, (2) Data Storing Module, and (3) Learning Module. We collect input-state data from the AV sensing unit.

- **Decision Making Module:** The collected input-state data is passed to the lane change decision making module. Using the collected data, this module conducts a safety check by evaluating the safety inequalities in (6)–(9). Based on the safety check, a lane change decision is made. If the safety conditions are satisfied and the AV is not in the target lane, the lane change is initiated by setting the target point at a safe distance from the LT. In an unsafe scenario, the target point is changed to the current lane to abort lane changing and is placed at a safe distance from LC. This module is evaluated at all times to ensure safe operation of the AV. For more details please see Section 3.
- **Data Storing Module:** As it is difficult for the AV to maintain a constant velocity on the road, we define a tolerance value ϵ_1 such that when the longitudinal velocity of the AV (i.e. V_x^{AV}) is close to one of the operating points V_x^i, i.e. when $|V_x^{AV} - V_x^i| \le \epsilon_1$, we assume $V_x^{AV} = V_x^i$ and store the collected data for learning the optimal controller for V_x^i in the database. As the longitudinal velocity of AV can vary, we store data for a particular V_x^i only when $|V_x^{AV} - V_x^i| \le \epsilon_1$.
- **Learning Module:** Once we collect enough data, say m samples for a particular velocity V_x^i, we pass the collected data from the database to the learning module. For convenience we denote $\mathbf{K}(\alpha_i)$ as $\mathbf{K}^i$. As discussed in Section 4, for each operating point V_x^i, an initial stabilizing gain $\mathbf{K}_0^i$ along with an exploration signal $\mathbf{e}$ was used to collect data for learning. The flag learnedV_x^i is used to avoid repeated learning for the same V_x^i. Once, a learned gain $\hat{\mathbf{K}}^{i*}$ is obtained for a V_x^i, we change $\mathbf{K}_0^i$ with $\hat{\mathbf{K}}^{i*}$ and use the interpolated gain given in (14) to obtain the control signal whenever $V_x^{AV} \in [V_x^i,\ V_x^j)$. Each step of the learning process is clearly depicted in the learning module in Fig. 3. Details on the theoretical analysis of this module can be found in Section 4.

5.2. Algorithmic details for learning optimal controller for longitudinal dynamics

The longitudinal dynamics is non-parameter varying and there is no need for gain scheduling. In this case, the data storing and learning module in Fig. 3 coincide with the work done by Jiang and Jiang (2012) and the decision making module remains as in Fig. 3.

Table 1
Sensitivity of converged gains due to change in ϵ_1.

	$\epsilon_1 = 0.08$	$\epsilon_1 = 0.1$	$\epsilon_1 = 0.3$	$\epsilon_1 = 0.5$	$\epsilon_1 = 0.7$	$\epsilon_1 = 1$
$\hat{e}_1$	0.4031	0.4031	0.4031	⋆	⋆	⋆
$\hat{e}_2$	1.0061	0.3887	1.5722	3.9523	2.1630	⋆
$\hat{e}_3$	0.1068	0.0168	2.7288	2.4247	3.9705	6.4967
$\hat{e}_4$	0.0796	0.2790	1.1321	2.1374	2.9564	4.9480
$\hat{e}_5$	0.3984	0.0260	1.0809	2.4079	5.0798	4.6419
$\hat{e}_6$	0.0443	0.1550	1.0343	2.7822	3.1806	4.8999

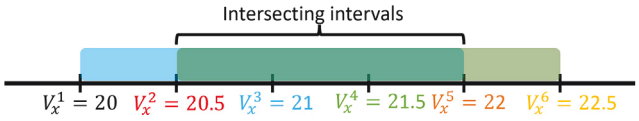


Fig. 4. Intersecting intervals for the scheduling points $V_x^3 = 21$ and $V_x^4 = 21.5$.

6. Results and discussions

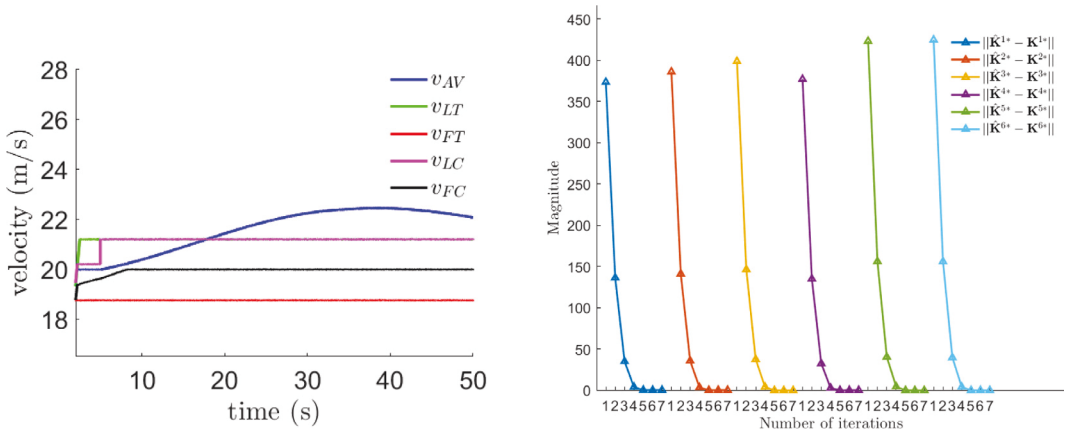
We obtain each $\mathbf{K}(\alpha_i)$ by means of the proposed learning-based control technique which guarantees the stability for each of the fixed α_i's. To guarantee the stability of the overall system, we need that $V_x(t)$ is slowly varying (Shahruz and Behtash, 1992). Since, $\dot{V}_x(t) = u_{lo}/m$, and $u_{lo} = -\mathbf{K}_{lo}\mathbf{x}_{lo}$, one needs to design $\mathbf{K}_{lo}$ such that vehicle acceleration has a small magnitude. We have generated the results by implementing the proposed technique of learning-based gain scheduling and lane change decision-making in the software packages MATLAB (MathWorks Inc., 2021) and Simulation of Urban MObility (SUMO). SUMO is an open source, portable, microscopic and continuous multi-modal traffic simulation package designed to handle large networks (Lopez et al., 2018). The SUMO simulation time started at $t_0 = 0$ s and terminated at $t_f = 80$ s. Data from SUMO environment is collected at every 0.01 s. In simulation, the standstill distance d is 2 m; the length of the vehicle L is 2.5 m; the width of the vehicle w is 1 m. 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 0.5 s to 1 s. For non-cooperative scenario we have varied the distance between FT and LT and studied the lane change maneuvers by varying the headway time (h) from 0.5 s to 1 s. We use the following weight matrices for the lateral controller design $\mathbf{Q} = \text{diag}([20, 50, 2000, 3000]) = \text{diag}([q_1, q_2, q_3, q_4])$, and $\mathbf{R} = 1$. We learn optimal controllers when the AV longitudinal velocity changes by half an unit, i.e., $\epsilon = 0.5$.

6.1. Sensitivity analysis for ϵ_1

In reality it is difficult for the AV to maintain a constant velocity on the road. Thus, in order to collect data for learning an optimal control gain for a scheduling point V_x^i, we use the parameter ϵ_1 to define the data collection interval for a scheduling point V_x^i. Whenever the condition $|V_x^{AV} - V_x^i| \leq \epsilon_1$ is satisfied, we start collecting data to learn the optimal controller gain for a scheduling point V_x^i. Thus, the parameter ϵ_1 plays a crucial role in determining the convergence of the learning algorithm for a scheduling point V_x^i. In this section, we conduct empirical sensitivity analysis of ϵ_1 in terms of convergence of the learning algorithm. We have selected multiple values of ϵ_1, i.e. $\epsilon_1 = [0.03, 0.05, 0.08, 0.1, 0.3, 0.5, 0.7, 1]$, and for each value of ϵ_1 we learn the approximate optimal control gain $\hat{\mathbf{K}}^{i*}$ for each scheduling point V_x^i. Then, we compute the error $\hat{e}_i = \|\hat{\mathbf{K}}^{i*} - \mathbf{K}^{i*}\|$, where $\mathbf{K}^{i*}$ is the optimal control gain. We have used 100 data samples to learn the optimal controller and observed that for $\epsilon_1 < 0.08$ the interval $|V_x^{AV} - V_x^i| \leq \epsilon_1$ is inadequate for collecting sufficient data to learn the optimal controller. Also, it was observed that for values of $\epsilon_1 \geq 0.3$, the intervals $|V_x^{AV} - V_x^i| \leq \epsilon_1$ intersect (see Fig. 4). When the actual velocity of the vehicle V_x^{AV} falls in this intersection, one can choose to learn the control gain for any of the scheduling point who shares this interval. However, while checking sequentially, only the next scheduling point is chosen. This leads to missed learning for some scheduling points. Also, as the intervals are made larger by choosing large values of ϵ_1, the actual velocity of the vehicle may lie away from the scheduling point. In this case, the collected data may not represent the dynamics of the vehicle for the scheduling point, which leads to noise in the collected data and the learned gains may be far away from the optimal gains. This is evident from Table 1 for all $\epsilon_1 \geq 0.3$, which shows the errors $\hat{e}_i$ computed for different values of ϵ_1, where the missing values are denoted as ⋆. Based on this analysis, we propose a thumb rule for selecting ϵ_1 given as $\epsilon_1 \leq \epsilon/2$. In this work, we choose $\epsilon_1 = 0.1$.

6.2. Learning-based controller

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 0.5 s to 1 s. We have observed that the AV could change the lane for all the considered h. The desired orientation of the vehicle is $\psi_{des} = 0°$. For the purpose of learning with an



(a) (b)

Fig. 5. (a) Velocities of the vehicles, (b) Convergence of $\mathbf{K}^i$ to $\mathbf{K}^{i*}$, $i = 1, 2, 3, 4, 5, 6$.

initial control gain $\mathbf{K}_0$, we apply the control input $\mathbf{u} = -\mathbf{K}_0\mathbf{x}_{la} + \mathbf{e}$, where $\mathbf{e}$ is the exploration signal which is obtained using the summations of sinusoidal signals with randomly distributed frequencies. Note that the noise $\mathbf{e}$ is deterministic (Jiang and Jiang, 2017). The choice of the $\mathbf{Q}$ and $\mathbf{R}$ matrices is done considering the passenger and driver comfort, and low fuel use. The diagonal entries q_1, q_2 will penalize $(e_1(t), \dot{e}_1(t))$ of AV, and q_3, q_4 will penalize $(e_2(t), \dot{e}_2(t))$ which will ensure passenger and driver comfort. Increasing q_1, q_2 will make the controller more aggressive, which might increase fuel consumption. Choosing $\mathbf{R} = 1$, 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.

As the AV longitudinal velocity changes, we learn optimal controllers for $V_x^1 = 20$ m/s, $V_x^2 = 20.5$ m/s, $V_x^3 = 21$ m/s, $V_x^4 = 21.5$ m/s, $V_x^5 = 22$ m/s, and $V_x^6 = 22.5$ m/s. We use the algorithm presented in Fig. 3 to perform gain scheduling-based learning using these initial stabilizing control gains: $\mathbf{K}_0^1 = [0.535,\ 0.023,\ 88.546,\ 92.441]$, $\mathbf{K}_0^2 = [0.535,\ 0.025,\ 88.879,\ 92.443]$, $\mathbf{K}_0^3 = [0.535,\ 0.026,\ 89.214,\ 92.445]$, $\mathbf{K}_0^4 = [0.535,\ 0.027,\ 89.549,\ 92.446]$, $\mathbf{K}_0^5 = [0.535,\ 0.028,\ 89.883,\ 92.448]$, $\mathbf{K}_0^6 = [0.535,\ 0.029,\ 90.218,\ 92.449]$. The tolerance ϵ_1 is set as 0.1. To demonstrate the learning process and application of the learned gains we perform the lane-changing two times.

Fig. 5(a) shows the velocity of the vehicles that are obtained from the SUMO environment. Each of the intervals $|V_x^{AV} - V_x^i| \leq \epsilon_1, i = 1, \ldots, 6$, comprises of 100 data points. Thus, with a sampling rate of 0.01 s, we collect data for 1 s for learning for every V_x^i. After the learned optimal controller gains are obtained for every V_x^i, the initial gains are replaced with the learned gains and the control policy for lane change maneuver is computed using the learned gains and the interpolation formula presented in (14). It was found that the lane changing time reduced with the application of the learned optimal gains for the lane change maneuver, where the initial lane change time is ≈ 4.5 s and with the application of the learned optimal gains, the lane change time is obtained as ≈ 3 s. 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.

Fig. 5(b) shows the convergence of the optimal gains. The ϵ_0 in Algorithm 1 is set as 10^{-4}. It is clearly seen from Fig. 5(b) that the gains converge to the optimal gains with just 7 iterations. Thus, it can be said that for the proposed algorithm, 100 samples or in other words 1 s data is enough for the learning algorithm to converge. The converged gains are: $\hat{\mathbf{K}}^{1*} = [4.472,\ 1.404,\ 149.405,\ 53.70705747]$, $\hat{\mathbf{K}}^{2*} = [4.476,\ 1.481,\ 151.177,\ 53.626]$, $\hat{\mathbf{K}}^{3*} = [4.478,\ 1.490,\ 154.118,\ 53.627]$, $\hat{\mathbf{K}}^{4*} = [4.423,\ 1.498,\ 156.357,\ 53.618]$, $\hat{\mathbf{K}}^{5*} = [4.489,\ 1.517,\ 159.150,\ 53.601]$, $\hat{\mathbf{K}}^{6*} = [4.472,\ 1.533,\ 161.463,\ 53.588]$. The optimal gains are: $\mathbf{K}^{1*} = [4.472\ 1.444\ 149.006\ 53.665]$, $\mathbf{K}^{2*} = [4.472,\ 1.463,\ 151.564,\ 53.649]$, $\mathbf{K}^{3*} = [4.472,\ 1.483,\ 154.105,\ 53.632]$, $\mathbf{K}^{4*} = [4.472,\ 1.503,\ 156.627,\ 53.615]$, $\mathbf{K}^{5*} = [4.472,\ 1.523,\ 159.132,\ 53.597]$, $\mathbf{K}^{6*} = [4.472,\ 1.543,\ 161.618,\ 53.579]$.

Fig. 6 shows the safe distance of the AV from the surrounding vehicles. It can be observed at t=0 s, the AV was not at a safe distance from FT, thus the AV does not start a lane change maneuver. At $t \approx 5$ s, the safety conditions for lane-changing (6)–(9) satisfy for all the surrounding vehicles and the AV starts the first lane change maneuver. For the second lane change maneuver, the vehicles are already at safe distance, thus the AV can safely start the lane change maneuver.

Fig. 7(a) shows the states of the lateral system. It can be seen that the states converge to zero with the application of the learned optimal controllers obtained using the proposed methodology. It was mentioned above that the gain scheduled controller can guarantee overall system stability if the feedback law $\mathbf{K}_{lo}$ for the longitudinal motion can be obtained such that vehicle acceleration has a small magnitude. Here, we have obtained the $\hat{\mathbf{K}}_{lo}^* = [4.4721, 110.3821]$ using $\mathbf{Q}_{lo} = \text{diag}([1,\ 1])$, $\mathbf{R}_{lo} = 0.05$. The choice of $\mathbf{Q}_{lo}$ and $\mathbf{R}_{lo}$ must be such that the acceleration has a small magnitude. Since, the AV longitudinal model is non-parameter varying, $\hat{\mathbf{K}}_{lo}^*$ can be learned using the techniques discussed by Jiang and Jiang (2012). Fig. 7(b) shows the longitudinal acceleration profile of the AV. It can be seen that the acceleration magnitude is low.

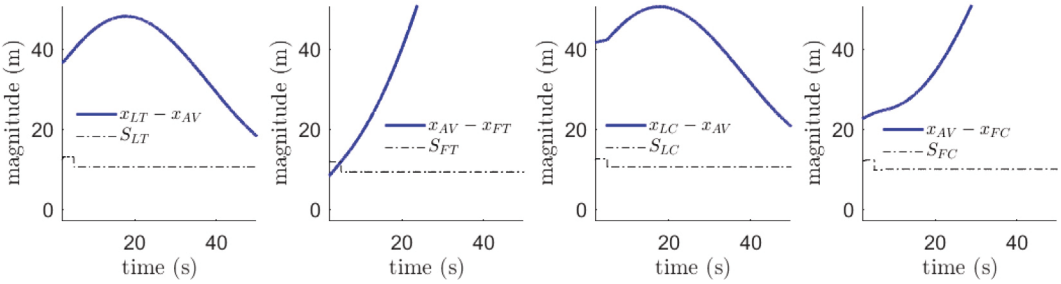


Fig. 6. Distance of AV from surrounding vehicles.

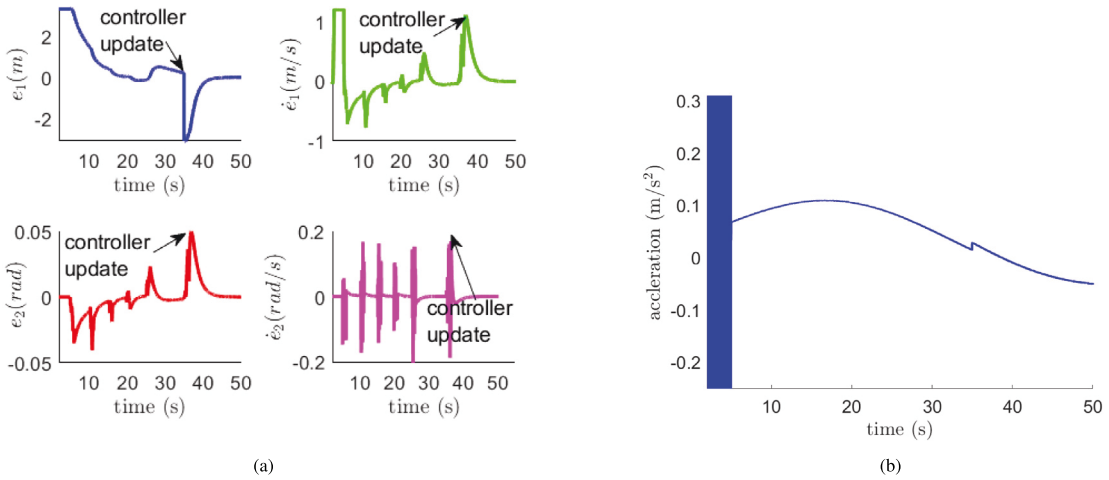


Fig. 7. (a) Error states, (b) Acceleration of AV.

6.3. Non-cooperative scenario

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 lane for all $h \leq 0.6$ s. The scenarios are presented as screenshots for $h = 0.5$ s in Fig. 8, for $h = 0.6$ s in Fig. 9, for $h = 0.65$ s in Fig. 10. It can be seen that the AV was able to change the lane for $h = 0.5$ s and $h = 0.6$ s.

For demonstrating the lane abortion scenario clearly, the case of $h = 0.6$ s is elaborately explained next. Fig. 11(a) shows lane abortion, and the velocities of the vehicles are shown in Fig. 11(b). The lane change starts at 48.16 s. From Fig. 11(b), it can be seen that the FT starts accelerating more than the AV. At around 50.03 s, 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 53.3 s. Again, at 54.2 s when the safety conditions are satisfied, the AV starts maneuvering to the target lane. It must be noted that the plots in Fig. 11(a) are normalized for the sake of clarity in understanding. For the cases where $h \geq 0.65$ s, the AV could not change the lane as the safety conditions did not satisfy for the simulation duration (see Fig. 10).

6.4. Comparison with fixed-gain controller

In order 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 2, Figs. 12(a) and 12(b). The headway time is 0.5 s, and the lane change velocity is in the range of 21.5 m/s to 23 m/s. We have performed five simulations, where we made the AV change lane using controller gains ($\mathbf{K}_{V_x}$) learned for each $V_x \in \{12\text{ m/s}, 15\text{ m/s}, 17\text{ m/s}, 20\text{ m/s}, 23\text{ m/s}\}$. Fig. 12(a) shows the trajectories obtained using gain scheduling and constant gains. It can be seen that when the AV changes lane with $\mathbf{K}_{12}$, 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. 12(b)). Table 2 shows the cost obtained using the proposed gain scheduling controller and the constant gain controllers. The cost is computed using $J = \int_{t_0}^{t_f} (\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u})dt$. It can

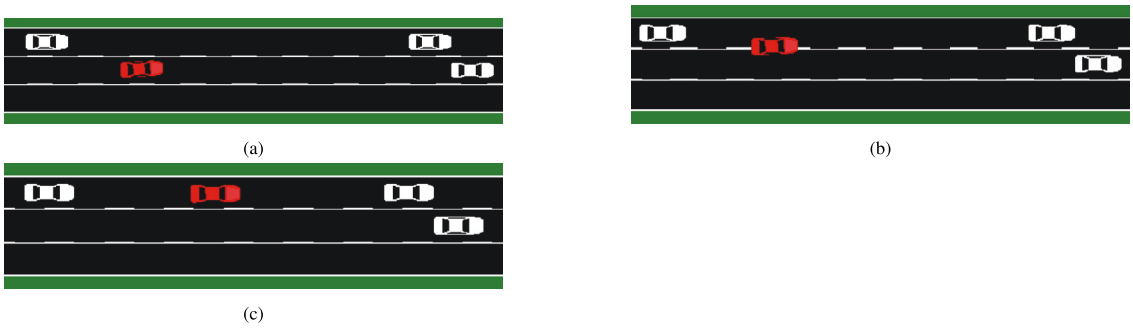


Fig. 8. SUMO screenshots for non-cooperative scenario with h=0.5 s: (a) t = 46.5 s, (b) t = 47.8 s, (c) t = 53.8 s.

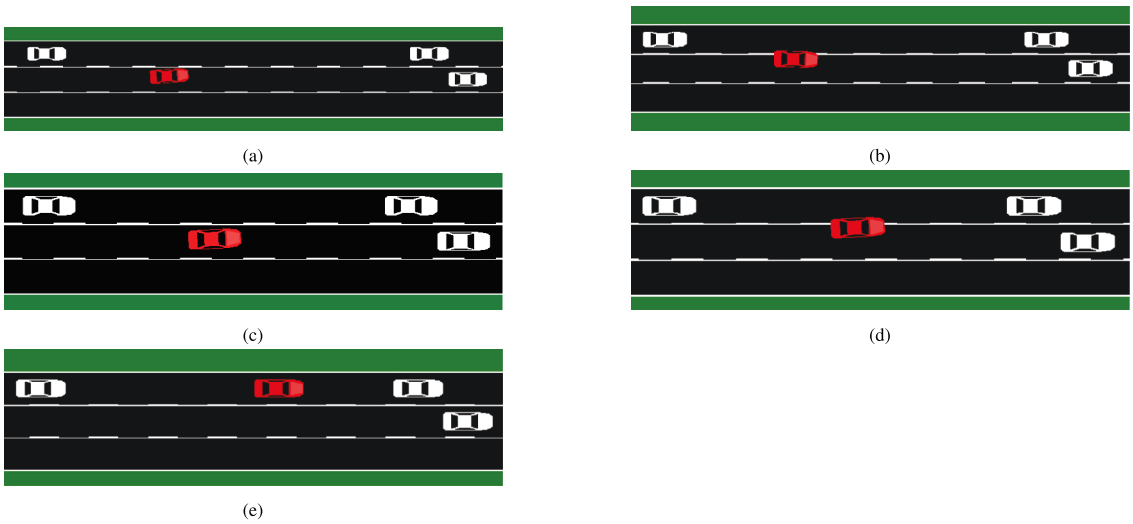


Fig. 9. SUMO screenshots for non-cooperative scenario with h=0.6 s: (a) t = 48.16 s, (b) t = 50.03 s, (c) t = 53.3 s (d) t = 54.2 s, (e) t = 70.1 s.

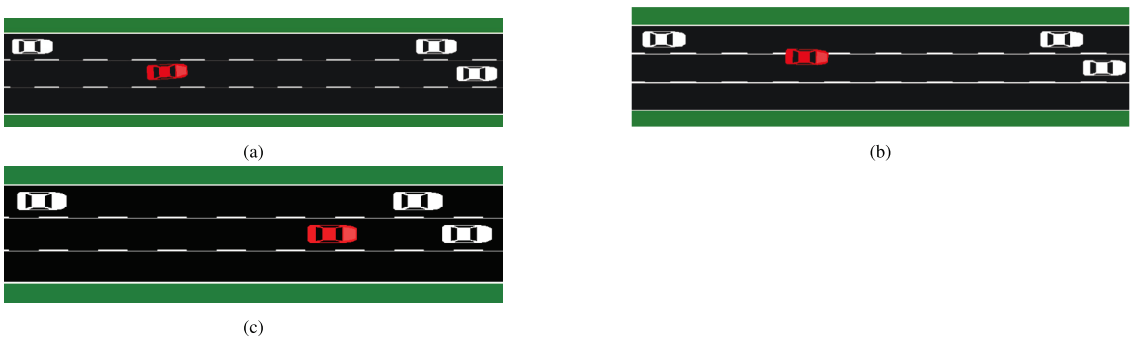
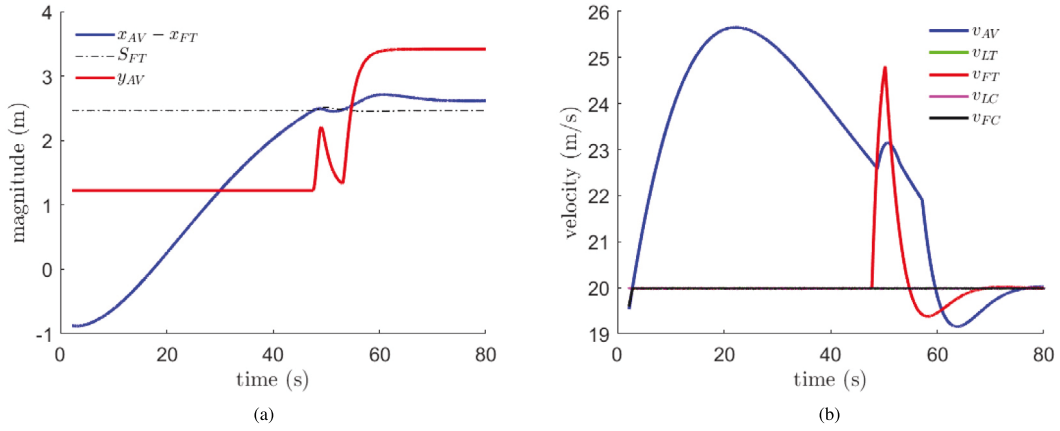


Fig. 10. SUMO screenshots for non-cooperative scenario with h=0.65 s: (a) t = 48.7 s, (b) t = 50 s, (c) t = 67.06 s.

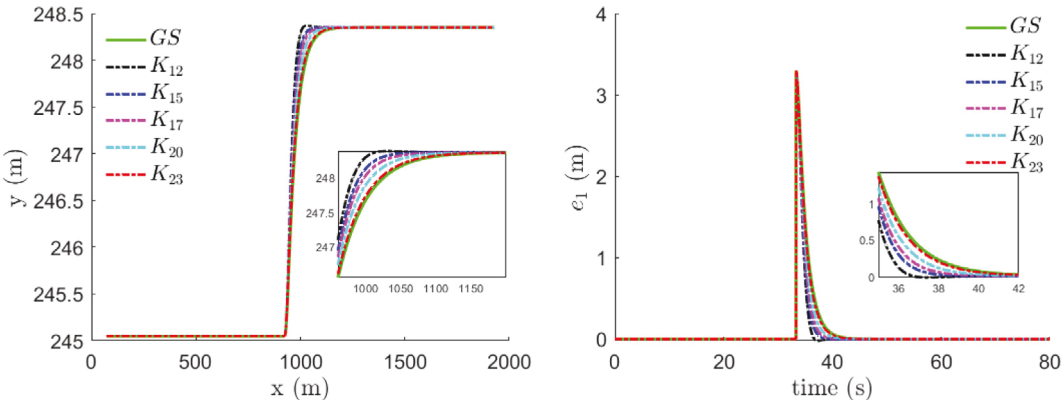
be seen that the proposed gain scheduling controller gives the least cost. This suggests that the gain scheduling controller is optimal when compared to the constant gain controllers.

Remark 6.1. As we schedule the controller gains that are learned for the velocities close to the actual AV velocity, the performance is seen to be improved. This observation suggests the importance of gain scheduling. The interpolation formula in (14) is used to achieve smooth transition from the present controller gain to the next. Instead, one can directly switch the controller gains with respect to the scheduling variable. But switching, might cause system instability and thus gain scheduling using the interpolation formula in (14) is more preferable.



(a) (b)

Fig. 11. (a) Lane abortion of AV, (b) Velocities during lane abortion.



(a) (b)

Fig. 12. (a) AV trajectories obtained using gain scheduling and constant gains, (b) error state e_1 obtained using gain scheduling and constant gains.

Table 2
Comparison with constant gain.

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

6.5. Comparison with model-based MPC

Model predictive control (MPC) is also an optimal control technique. MPC tries to find the optimal control input at each time step by minimizing the cost function $J = \sum_{i=0}^{N_p}(\mathbf{x}_i^T\mathbf{Q}\mathbf{x}_i + \mathbf{u}_i^T\mathbf{R}\mathbf{u}_i) + \mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N$, where $\mathbf{x}_N^T\mathbf{Q}_N\mathbf{x}_N$ is the terminal cost, and N_p is the prediction horizon. Designing a proper $\mathbf{Q}_N$ 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, MPC controller is used to track a trajectory generated by the trajectory planning module for 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=0.5 s. The plots obtained from SUMO simulations are given in Figs. 13(a) and 13(b). It can be seen that the MPC controller does not achieve satisfactory performance with small prediction horizon. As the prediction horizon is

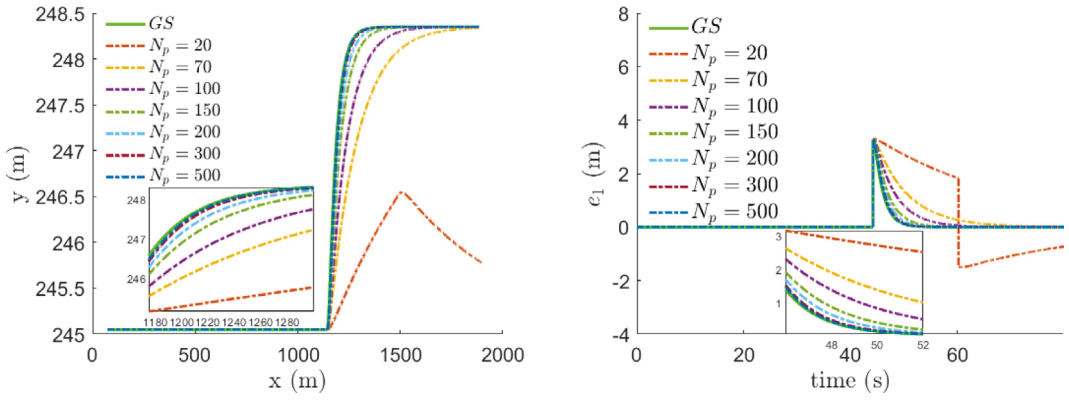


Fig. 13. (a) *AV* trajectories obtained using gain scheduling and MPC, (b) error state e_1 obtained using gain scheduling and MPC.

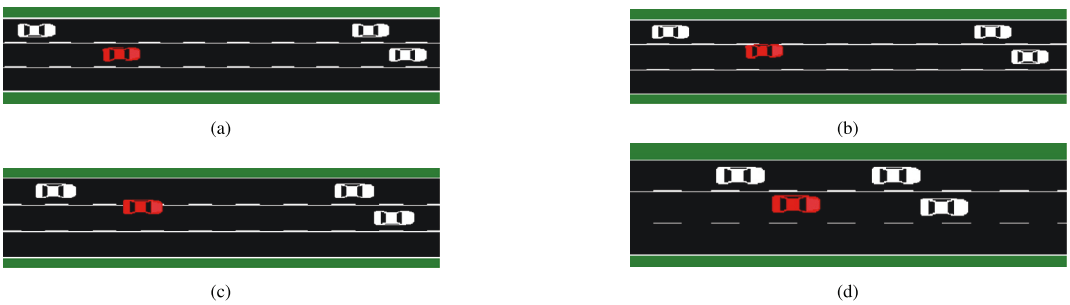


Fig. 14. SUMO screenshots obtained for MPC controller in a non-cooperative scenario with h=0.5 s, N_p=100: (a) t = 47 s, (b) t = 47 s, (c) t = 49.5 s (d) t = 54 s.

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

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 Figs. 14 and 15. 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 angle). Whereas, when $N_p = 300$, the *AV* could successfully change the lane and the performance is similar to the proposed learning-based gain scheduling controller.

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 as shown in Table 3. In Table 3, the computation start time is 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 algorithm given in Fig. 3 is executed. Thus, we execute the algorithm given in Fig. 3, 100 times to get a distribution of the learning times for different control gains. The histogram for learning times for each $\hat{\mathbf{K}}^{i*}$ is given in Fig. 16 and the histogram of total learning times is given in Fig. 17. Comparing the total computation time of MPC in Table 3 and the total learning time for

Table 3
Computation time of MPC.

N_p	Computation start, end time	Time steps	Average computation time per step	Total computation time
150	39.86 s, 48.76 s	900	0.0011 s	0.99 s
200	39.86 s, 47.15 s	730	0.0013 s	0.95 s
300	39.86 s, 46.12 s	630	0.002 s	1.26 s
500	39.86 s, 45.82 s	600	0.0034 s	2 s

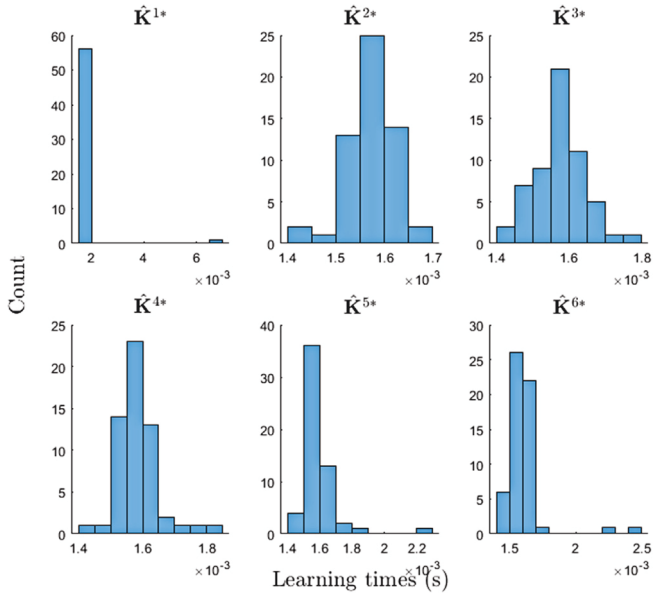


Fig. 16. Histogram of learning time for each $\hat{\mathbf{K}}^{i*}$.

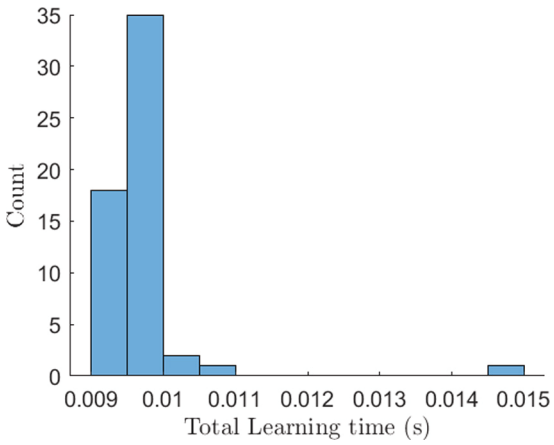


Fig. 17. Histogram of total learning time in 100 runs obtained by adding the learning time for each $\hat{\mathbf{K}}^{i*}$ of the proposed gain scheduling algorithm.

the proposed technique in Fig. 17, it can be said that the proposed learning-based gain scheduling technique is computationally efficient when compared to MPC.

Remark 6.2. It must be noted that if the ϵ is reduced further, we need to learn more number of controller gains due to increased number of scheduling points. This would increase the total learning time. However, note that the lane change performance of the AV is found to be satisfactory by learning controllers whenever the AV longitudinal velocity changes by half an unit, i.e., $\epsilon = 0.5$. Also, it must be noted that once the control gains are learned, we do not need to re-learn them unless the weight matrices $\mathbf{Q}$ and $\mathbf{R}$ are changed. Thus, the computation burden is for one-time learning for the proposed technique. However, in case of MPC whenever

the AV changes lane an optimization problem need to be solved to compute the steering angle at each time step of the lane change duration.

Remark 6.3. If $\mathbf{K}_{lo}^{*}$ is very conservative such that the AV response in tracking the x-coordinate of the target point (T_x) is sluggish, one can change $\mathbf{K}_{lo}^{*}$ to a more aggressive gain for target tracking. This must only be done when the states of the lateral dynamics are negligible. In other words, when the AV has already reached T_y, and there is no dependence on the lateral dynamics, one can switch to an aggressive $\mathbf{K}_{lo}^{*}$ for better tracking of T_x.

Remark 6.4. This paper proposes an integrated learning-based method to the AV lane change problem. Any prior information of the system parameters is not assumed. 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 either do not guarantee optimal control of the AVs or require to solve an optimization problem at every time step. The proposed methodology in this work learns only at specific time intervals with a smaller number of data points whenever the AV longitudinal velocity changes by half an unit, i.e., $\epsilon = 0.5$. Also, due to the fast convergence, our proposed methodology is suitable for real-time applications. Although, we assume that we receive data from a linear model, the gain scheduling based learning-based controller design adds to the versatility of the proposed methodology that makes it applicable to non-linear systems and/or parameter-varying systems.

6.6. Evaluation of safety of lane change maneuver

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 by the Park et al. (2018). Park et al. (2018) 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:

$$SSD_i(t) = \frac{V_i(t)^2}{254 \times (f \pm g)} + t_r \times V_i(t) \times 0.278, \tag{41}$$

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 (2.5 s), $i \in \{LT, FT, AV, LC, FC\}$. Once, the SSDs are computed, one can compute the SDIs as follows:

$$SDI_{i,j}(t) = \begin{cases} \text{safe}, & if\, S_{i,j}(t) + SSD_i(t) - SSD_j(t) - l_i > 0 \\ \text{unsafe}, & \text{otherwise}, \end{cases} \tag{42}$$

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 sight distance for the front vehicle, $SSD_j(t)$ is the stopping sight distance 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:

$$REL_{i,j} = \frac{ULCD}{TLCD}, \tag{43}$$

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

$$RSL_{i,j} = \frac{\max(-SDI_{i,j}(t))}{SDI_{cri}}, \tag{44}$$

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 Park et al. (2018):

$$\phi_k = REL_{i,j} \times RSL_{i,j}, \tag{45}$$

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

$$\phi_{AV} = 1 - \prod_{k=1}^{4}(1 - \phi_k). \tag{46}$$

The SDI for the AV and FT for the proposed gain scheduling controller and MPC are shown in Fig. 18(a) and Fig. 18(b) 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

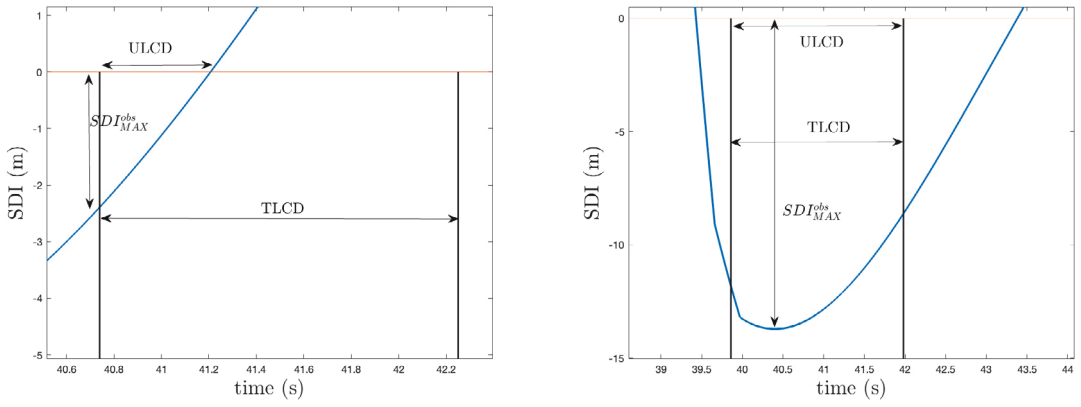


Fig. 18. (a) SDI for gain scheduling obtained for AV and FT, (b) SDI for MPC obtained for AV and FT.

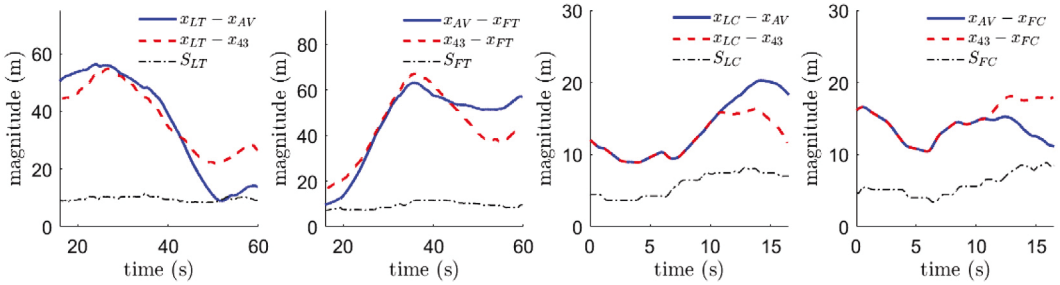


Fig. 19. Comparison of safe distances maintained by AV and vehicle 43 from surrounding vehicles.

this work, SDI_{cri} is set to 40 m considering the spacing between two interacting vehicles is 0 m and that the speed of the following vehicle is 100 kph.

After computation, we have obtained the LCRI for the case of gain scheduling as ϕ_{AV}=0.05, and for the case of MPC as ϕ_{AV}=0.5 with 150 steps as prediction horizon. Also, note that the proposed learning-based gain scheduling technique needs only 100 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. However, note that increasing N_p increases the computation time. In comparison, the proposed methodology provides better safety in a non-cooperative scenario with lower computation cost, which is essential in safety-critical scenarios. Thus, it can be said that the proposed gain scheduling technique is safer and also computationally efficient.

6.7. Performance evaluation using NGSIM data

This section presents the simulation results using SUMO and the vehicle trajectories data obtained from NGSIM (NGSIM, 2016) dataset, which is a program funded by the U.S. Federal Highway Administration. We have tested our method for the US101 highway dataset. The US101 dataset consists a total of 45 min of data divided into fifteen minutes periods: 7:50am–8:05am, 8:05am–8:20am, and 8:20am–8:35am. We have collected the trajectories of vehicles 30, 43, 47, 50, 54 from the 8:20am–8:35am dataset of US101 dataset, where vehicle 43 performs a lane change from lane 1 to lane 2. From the dataset, we can define 50 as FC, 47 as LC, 54 as FT, 30 as LT. To compare the proposed controller, we replace vehicle 43 with AV equipped with the proposed controller. The proposed algorithm is used to learn optimal controllers for V_x^1 = 7.1 m/s, V_x^2 = 7.9 m/s, V_x^3 = 8.7 m/s, V_x^4 = 9.5 m/s, V_x^5 = 10.3 m/s, and V_x^6 = 11.1 m/s. These scheduling points are selected based on the lane changing velocities of vehicle 43. For SUMO simulation, we let the AV start with the same initial condition as vehicle 43. The distances of AV and vehicle 43 from the surrounding vehicles are shown in Fig. 19. It can be seen that both AV and vehicle 43 can maintain safe distances from the surrounding vehicles. Figs. 20(a) and 20(b) compares the trajectories and the velocity profiles of the AV and vehicle 43, respectively. The results show a similar trend in trajectory and velocity profiles with comparatively smoother profiles generated by the AV. Thus, the proposed algorithm may lead to better passenger comfort and less fuel usage. It is desired to have a smoother lateral acceleration profile and minimize lateral jerk during lane changing (see Luo et al., 2016). The lateral acceleration and jerk profiles for AV and vehicle 43 are shown in Figs. 20(c) and 20(d), respectively. It can be seen that the AV produced smoother lateral accelerations and lesser lateral jerks as compared to vehicle 43. Therefore, the AV equipped with the proposed controller may lead to a comfortable and safe transportation experience.

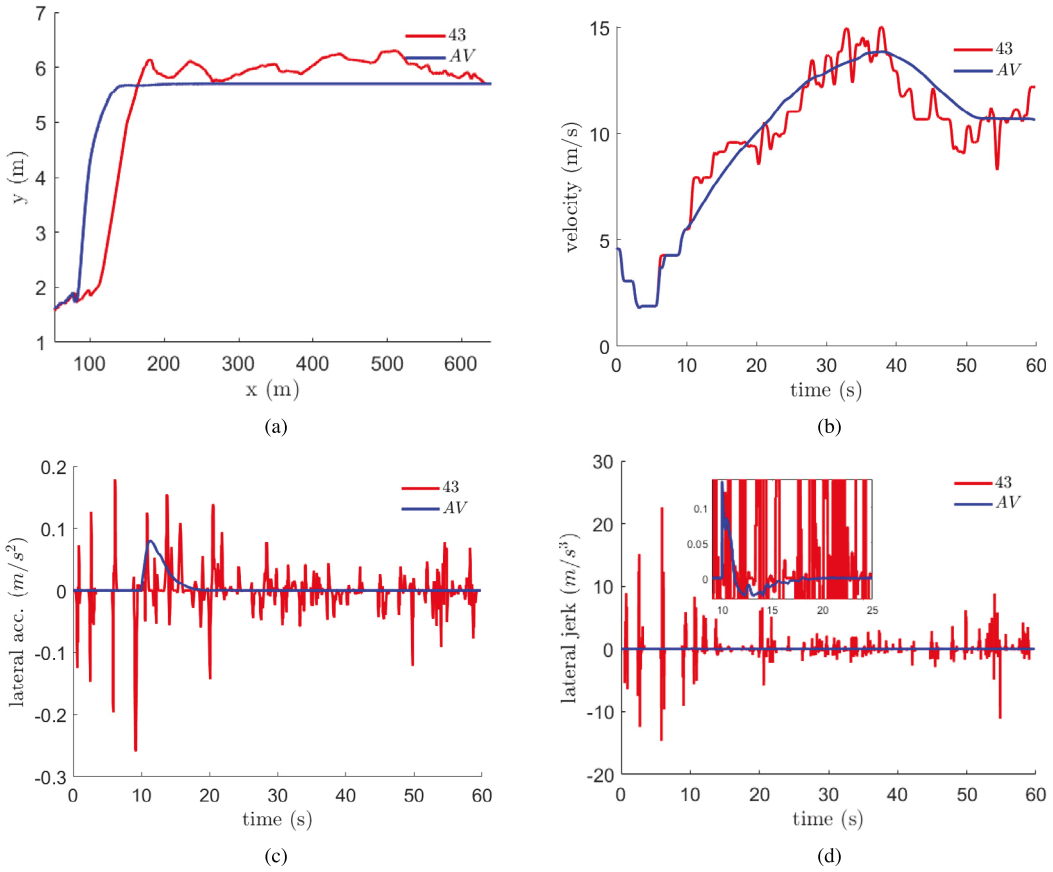


Fig. 20. Performance evaluation of the proposed methodology using NGSIM Data: (a) Trajectories of AV and vehicle 43, (b) Velocities of AV and vehicle 43, (c) Lateral accelerations of AV and vehicle 43, (d) Lateral jerks of AV and vehicle 43.

7. Conclusions

In this study, we have introduced a novel approach for addressing the lane-changing challenge in autonomous vehicles (AVs). This innovative methodology makes use of the online information of the state obtained from vehicular sensors and control input to iteratively solve the algebraic Riccati equation online using approximate/adaptive dynamic programming framework, assuming no knowledge of the system parameters. It was observed that the vehicle lateral dynamics depend on the longitudinal velocity, which leads to a parameter varying AV lateral dynamics. Assuming constant longitudinal velocity might lead to conservative lane change maneuvers, which might increase the risk of accidents, especially during lane abortion maneuvers in a non-cooperative scenario. This study extends the applicability of learning-based optimal control to nonlinear and/or parameter-varying systems by proposing a gain-scheduling-based learning-based control technique. We have conducted rigorous theoretical analysis, demonstrating that the stability of the proposed learning-based gain-scheduling controller is guaranteed when scheduling points are selected in close proximity. Additionally, we have implemented a lane-change decision-making algorithm to ensure safe and efficient lane changes, including the ability to abort lane changes in non-cooperative scenarios. Empirical results indicate a reduction in lane-change time when utilizing optimal controller gains compared to non-optimal counterparts. Safety assessments were performed using the lane change risk index for both the proposed learning-based gain-scheduling controller and model-based model predictive control (MPC). Our findings suggest that the proposed controller offers enhanced safety with reduced computational effort when compared to MPC, which is necessary for safety-critical applications like lane changing. The effectiveness of our methodology has been validated through extensive simulations conducted using MATLAB, SUMO and NGSIM dataset, reaffirming its practical viability and potential for application in autonomous driving systems.

In future work, we aim to study a multi-agent framework, encompassing scenarios where multiple AVs engage in lane-changing maneuvers within mixed traffic environments. This investigation will encompass the exploration of direct communication possibilities among AVs, as well as a comprehensive analysis of potential challenges arising from the presence of potentially misleading communication signals. Also, we plan to study the interaction between AVs and nearby human-driven vehicles in situations where direct communication is unattainable, yet their behavior remains estimable through other means. Furthermore, we plan to validate our proposed methodologies with more results using real-world datasets. This will contribute to a deeper

understanding of the complex interactions within mixed traffic scenarios, advancing our knowledge of autonomous driving systems' performance and safety in real-world conditions.

CRediT authorship contribution statement

Sayan Chakraborty: Conceptualization, Investigation, Methodology, Review, Analysis of results. **Leilei Cui:** Conceptualization, Methodology, Review, Analysis of results. **Kaan Ozbay:** Conceptualization, Methodology, Review, Analysis of results, Supervision. **Zhong-Ping Jiang:** Conceptualization, Methodology, Review, Analysis of results, Supervision.

Acknowledgments

The work in this paper is partially funded by NYU's C2SMARTER center through a grant from the U.S. Department of Transportation's University Transportation Centers Program and partially by the U.S. National Science Foundation grants CPS-2227153 and CNS-2148309. However, the U.S. Government assumes no liability for the contents or use thereof. The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein.

Appendix A. Proof of Theorem 4.14

Proof. Since the elements of $\mathbf{A}(\alpha)$ and $\mathbf{B}(\alpha)$ are analytic functions of α, for small ϵ, we have $\mathbf{A}(\alpha) = \mathbf{A}(\alpha_l)+\mathcal{O}(\epsilon)$, and $\mathbf{B}(\alpha) = \mathbf{B}(\alpha_l)+\mathcal{O}(\epsilon)$. Thus, for any α the Lyapunov equation (18) can be written as the following:

$$\mathbf{A}_k(\alpha)^T\mathbf{P}_k(\alpha) + \mathbf{P}_k(\alpha)\mathbf{A}_k(\alpha) + \mathbf{Q} + \mathbf{K}_k(\alpha)^T\mathbf{R}\mathbf{K}_k(\alpha) = \mathbf{0}, \tag{A.1}$$

where, $\mathbf{A}_k(\alpha) = \mathbf{A}(\alpha) - \mathbf{B}(\alpha)\mathbf{K}_k(\alpha)$. Let, $\mathbf{K}_{k+1}(\epsilon, \mathbf{P}_k(\alpha)) := \mathbf{K}_{k+1}(\alpha) = \mathbf{R}^{-1}\mathbf{B}(\alpha_l)^T\mathbf{P}_k(\alpha) + \mathcal{O}(\epsilon)$. Then, one can obtain $\mathbf{A}_k(\alpha) = \mathbf{A}(\alpha_l) - \mathbf{B}(\alpha_l)\mathbf{K}_k(\alpha) + \mathcal{O}(\epsilon)$. Next, define the following function:

$$\mathbf{F}_k(\epsilon, \mathbf{P}_k(\alpha)) = \mathbf{A}_k(\alpha)^T\mathbf{P}_k(\alpha) + \mathbf{P}_k(\alpha)\mathbf{A}_k(\alpha) + \mathbf{Q} + \mathbf{K}_k(\epsilon, \mathbf{P}_{k-1}(\alpha))^T\mathbf{R}\mathbf{K}_k(\epsilon, \mathbf{P}_{k-1}(\alpha)). \tag{A.2}$$

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

$$\mathbf{F}_1(0, \mathbf{P}_1(\alpha_l)) = \mathbf{A}_1(\alpha_l)^T\mathbf{P}_1(\alpha_l) + \mathbf{P}_1(\alpha_l)\mathbf{A}_1(\alpha_l) + \mathbf{Q} + \mathbf{K}_1(\alpha_l)^T\mathbf{R}\mathbf{K}_1(\alpha_l). \tag{A.3}$$

Note that $\mathbf{K}_1(\alpha_l)$ is the known initial stabilizing controller gain that is used to start the iteration for the model-free learning (see Section 4.1). Thus, $\mathbf{F}(0, \mathbf{P}_1(\alpha_l)) = 0$ has a unique solution $\mathbf{P}_1(\alpha_l)$ as $\mathbf{A}_1(\alpha_l)$ is stable. Also, we have that:

$$\mathbf{K}_2(0, \mathbf{P}_1(\alpha_l)) = \mathbf{R}^{-1}\mathbf{B}(\alpha_l)^T\mathbf{P}_1(\alpha_l). \tag{A.4}$$

Now,

$$\mathbf{F}_1(\epsilon, \mathbf{P}_1(\alpha)) = \mathbf{A}_1(\alpha)^T\mathbf{P}_1(\alpha) + \mathbf{P}_1(\alpha)\mathbf{A}_1(\alpha) + \mathbf{Q} + \mathbf{K}_1(\alpha)^T\mathbf{R}\mathbf{K}_1(\alpha), \tag{A.5}$$

where $\mathbf{K}_1(\alpha) = \mathbf{K}_1(\alpha_l) + \frac{\mathbf{K}_1(\alpha_{l+1})-\mathbf{K}_1(\alpha_l)}{\alpha_{l+1}-\alpha_l}(\alpha - \alpha_l)$. Note that $\mathbf{K}_1(\alpha_{l+1})$ is also a known initial stabilizing controller gain for the scheduling point α_{l+1}. Using Lemma 4.6 for (A.5) and taking the derivative with respect to $\mathrm{vec}(\mathbf{P}_1(\alpha))$ at the point $(\epsilon = 0, \mathbf{P}_1(\alpha_l))$, we have the following:

$$\left.\frac{\partial \mathrm{vec}(\mathbf{F}_1(\epsilon, \mathbf{P}_1(\alpha)))}{\partial \mathrm{vec}(\mathbf{P}_1(\alpha))}\right|_{(0, \mathbf{P}_1(\alpha_l))} = \mathbf{I} \otimes \mathbf{A}_1(\alpha_l)^T + \mathbf{A}_1(\alpha_l)^T \otimes \mathbf{I} \tag{A.6}$$

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

$$\mathbf{P}_1(\alpha) = \mathbf{P}_1(\alpha_l) + \mathcal{O}(\epsilon) \tag{A.7}$$

$$\mathbf{K}_2(\epsilon, \mathbf{P}_1(\alpha)) = \mathbf{K}_2(0, \mathbf{P}_1(\alpha_l)) + \mathcal{O}(\epsilon), \tag{A.8}$$

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

$$\mathbf{F}_2(0, \mathbf{P}_2(\alpha_l)) = \mathbf{A}_2(\alpha_l)^T\mathbf{P}_2(\alpha_l) + \mathbf{P}_2(\alpha_l)\mathbf{A}_2(\alpha_l) + \mathbf{Q} + \mathbf{K}_2(0, \mathbf{P}_1(\alpha_l))^T\mathbf{R}\mathbf{K}_2(0, \mathbf{P}_1(\alpha_l)). \tag{A.9}$$

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

$$\mathbf{K}_3(0, \mathbf{P}_2(\alpha_l)) = \mathbf{R}^{-1}\mathbf{B}(\alpha_l)^T\mathbf{P}_2(\alpha_l) \tag{A.10}$$

Now,

$$\mathbf{F}_2(\epsilon, \mathbf{P}_2(\alpha)) = \mathbf{A}_2(\alpha)^T\mathbf{P}_2(\alpha) + \mathbf{P}_2(\alpha)\mathbf{A}_2(\alpha) + \mathbf{Q} + \mathbf{K}_2(\alpha)^T\mathbf{R}\mathbf{K}_2(\alpha). \tag{A.11}$$

From (A.8), $\mathbf{K}_2(\alpha) = \mathbf{K}_2(\alpha_l) + \mathcal{O}(\epsilon)$. Thus, following the similar steps for the $k = 1$ case, by using the implicit function theorem, one can obtain an unique solution for $\mathbf{P}_2(\alpha)$ with $\mathbf{F}_2(\epsilon, \mathbf{P}_2(\alpha)) = 0$ that is analytic in ϵ. Hence, we have the following:

$$\mathbf{P}_2(\alpha) = \mathbf{P}_2(\alpha_l) + \mathcal{O}(\epsilon) \tag{A.12}$$

$$\mathbf{K}_3(\epsilon, \mathbf{P}_2(\alpha)) = \mathbf{K}_3(0, \mathbf{P}_2(\alpha_l)) + \mathcal{O}(\epsilon), \tag{A.13}$$

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

Repeating the above analysis for $k = 3, 4, \ldots$, the statement of the theorem is proved.

Appendix B. Proof of Theorem 4.15

Proof. From Theorem 4.14, we have:

$$\hat{\mathbf{K}}^*(\alpha_{l+1}) = \hat{\mathbf{K}}^*(\alpha_l) + \mathcal{O}(\epsilon), \tag{B.1}$$

Thus, (14) implies that:

$$\mathbf{K}(\alpha) = \hat{\mathbf{K}}^*(\alpha_l) + \mathcal{O}(\epsilon), \tag{B.2}$$

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

$$\mathbf{A}_c(\alpha) = \mathbf{A}(\alpha) - \mathbf{B}(\alpha)\mathbf{K}(\alpha) = \mathbf{A}_c(\alpha_l) + \mathcal{O}(\epsilon) \tag{B.3}$$

Thus, using the results on the analytic perturbation of eigenvalues (Lancaster and Tismenetsky, 1985), the statement of the theorem holds.

References

Alcala, E., Puig, V., Quevedo, J., Escobet, T., 2018. Gain-scheduling LPV control for autonomous vehicles including friction force estimation and compensation mechanism. IET Control Theory Appl. 12 (12), 1683–1693.

Bellman, R., 1966. Dynamic programming. Science 153 (3731), 34–37.

Bertsekas, D., 2012. Dynamic Programming and Optimal Control, vol. 1, Athena scientific.

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

Bianchi, F.D., Mantz, R., Christiansen, C., 2004. Control of variable-speed wind turbines by LPV gain scheduling. Wind Energy: Int. J. Progress Appl. Wind Power Convers. Technol. 7 (1), 1–8.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al., 2016. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

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

Chen, C.-T., 1999. Linear System Theory and Design, third ed. Oxford University Press, New York.

Chovan, J., Tijerina, L., Alexander, G., Hendricks, D., 1994. Examination of Lane Change Crashes and Potential IVHS Countermeasures. Final Report. Technical Report.

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

Codevilla, F., Müller, M., López, A., Koltun, V., Dosovitskiy, A., 2018. End-to-end driving via conditional imitation learning. In: 2018 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 4693–4700.

Codevilla, F., Santana, E., López, A.M., Gaidon, A., 2019. Exploring the limitations of behavior cloning for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9329–9338.

Farazi, N.P., Zou, B., Ahamed, T., Barua, L., 2021. Deep reinforcement learning in transportation research: A review. Transp. Res. Interdiscipl. Perspect. 11, 100425.

Folkers, A., Rick, M., Büskens, C., 2019. Controlling an autonomous vehicle with deep reinforcement learning. In: 2019 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 2025–2031.

Gao, W., Gao, J., Ozbay, K., Jiang, Z.-P., 2019. Reinforcement-learning-based cooperative adaptive cruise control of buses in the Lincoln tunnel corridor with time-varying topology. IEEE Trans. Intell. Transp. Syst. 20 (10), 3796–3805.

Gao, W., Jiang, Z.P., 2016. Adaptive dynamic programming and adaptive optimal output regulation of linear systems. IEEE Trans. Autom. Control 61 (12), 4164–4169.

Gao, W., Jiang, Z.-P., Ozbay, K., 2015. Adaptive optimal control of connected vehicles. In: 2015 10th International Workshop on Robot Motion and Control (RoMoCo). IEEE, pp. 288–293.

Gao, W., Jiang, Z.P., Ozbay, K., 2016. Data-driven adaptive optimal control of connected vehicles. IEEE Trans. Intell. Transp. Syst. 18 (5), 1122–1133.

Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G., 2020. A survey of deep learning techniques for autonomous driving. J. Field Robot. 37 (3), 362–386.

Hecker, S., Dai, D., Van Gool, L., 2018. End-to-end learning of driving models with surround-view cameras and route planners. In: Proceedings of the European Conference on Computer Vision. Eccv, pp. 435–453.

Hoel, C.-J., Driggs-Campbell, K., Wolff, K., Laine, L., Kochenderfer, M.J., 2019. Combining planning and deep reinforcement learning in tactical decision making for autonomous driving. IEEE Trans. Intell. Veh. 5 (2), 294–305.

Hoel, C.-J., Wolff, K., Laine, L., 2018. Automated speed and lane change decision making using deep reinforcement learning. In: 2018 21st International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 2148–2155.

Hoffmann, C., Werner, H., 2014. A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. IEEE Trans. Control Syst. Technol. 23 (2), 416–433.

Horn, R.A., Johnson, C.R., 2012. Matrix Analysis. Cambridge University Press.

Jiang, Y., Jiang, Z.P., 2012. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. Automatica 48 (10), 2699–2704.
Jiang, Y., Jiang, Z.P., 2014a. Adaptive dynamic programming as a theory of sensorimotor control. Biol. Cybernet. 108 (4), 459–473.
Jiang, Y., Jiang, Z.P., 2014b. Robust adaptive dynamic programming and feedback stabilization of nonlinear systems. IEEE Trans. Neural Netw. Learn. Syst. 25 (5), 882–893.
Jiang, Y., Jiang, Z.P., 2017. Robust Adaptive Dynamic Programming. John Wiley & Sons.
Jin, I.G., Avedisov, S.S., He, C.R., Qin, W.B., Sadeghpour, M., Orosz, G., 2018. Experimental validation of connected automated vehicle design among human-driven vehicles. Transp. Res. Part C: Emerg. Technol. 91, 335–352.
Kapsalis, D., Sename, O., Milanés, V., Martinez, J.J., 2020. Gain-scheduled steering control for autonomous vehicles. IET Control Theory Appl. 14 (20), 3451–3460.
Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A.A., Yogamani, S., Pérez, P., 2021. Deep reinforcement learning for autonomous driving: A survey. IEEE Trans. Intell. Transp. Syst. 23 (6), 4909–4926.
Kleinman, D., 1968. On an iterative technique for Riccati equation computations. IEEE Trans. Autom. Control 13 (1), 114–115.
Kuderer, M., Gulati, S., Burgard, W., 2015. Learning driving styles for autonomous vehicles from demonstration. In: 2015 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 2641–2646.
Lancaster, P., Tismenetsky, M., 1985. The Theory of Matrices: with Applications. Elsevier.
Lewis, F.L., Vrabie, D., 2009. Reinforcement learning and adaptive dynamic programming for feedback control. IEEE Circuits Syst. Mag. 9 (3), 32–50.
Lewis, F.L., Vrabie, D., Vamvoudakis, K.G., 2012. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. IEEE Control Syst. Mag. 32 (6), 76–105.
Li, X., 2022. Trade-off between safety, mobility and stability in automated vehicle following control: An analytical method. Transp. Res. B 166, 1–18.
Li, G., Qiu, Y., Yang, Y., Li, Z., Li, S., Chu, W., Green, P., Li, S.E., 2022a. Lane change strategies for autonomous vehicles: A deep reinforcement learning approach based on transformer. IEEE Trans. Intell. Veh..
Li, S., Wei, C., Wang, Y., 2022b. Combining decision making and trajectory planning for lane changing using deep reinforcement learning. IEEE Trans. Intell. Transp. Syst. 23 (9), 16110–16136.
Li, G., Yang, Y., Li, S., Qu, X., Lyu, N., Li, S.E., 2022c. Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness. Transp. Res. Part C: Emerg. Technol. 134, 103452.
Lopez, P.A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., Wießner, E., 2018. Microscopic traffic simulation using sumo. In: 2018 21st International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 2575–2582.
Luo, Y., Xiang, Y., Cao, K., Li, K., 2016. A dynamic automated lane change maneuver based on vehicle-to-vehicle communication. Transp. Res. C 62, 87–102.
Ma, K., Wang, H., Zuo, Z., Hou, Y., Li, X., Jiang, R., 2022. String stability of automated vehicles based on experimental analysis of feedback delay and parasitic lag. Transp. Res. Part C: Emerg. Technol. 145, 103927.
Markushevich, A.I., 2005. Theory of Functions of a Complex Variable, vol. 296, American Mathematical Soc..
MathWorks Inc., 2021. MATLAB version: 9.13.0 (r2021b). URL https://www.mathworks.com.
Min, K., Kim, H., Huh, K., 2018. Deep q learning based high level driving policy determination. In: 2018 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 226–231.
Min, K., Kim, H., Huh, K., 2019. Deep distributional reinforcement learning based high-level driving policy determination. IEEE Trans. Intell. Veh. 4 (3), 416–424.
Mirchevska, B., Pek, C., Werling, M., Althoff, M., Boedecker, J., 2018. High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning. In: 2018 21st International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 2156–2162.
Nageshrao, S., Tseng, H.E., Filev, D., 2019. Autonomous highway driving using deep reinforcement learning. In: 2019 IEEE International Conference on Systems, Man and Cybernetics. SMC, IEEE, pp. 2326–2331.
Neal, B., Mittal, S., Baratin, A., Tantia, V., Scicluna, M., Lacoste-Julien, S., Mitliagkas, I., 2018. A modern take on the bias-variance tradeoff in neural networks. arXiv preprint arXiv:1810.08591.
NGSIM, 2016. U.S. Department of Transportation Federal Highway Administration. next generation simulation (NGSIM) vehicle trajectories and supporting data. [US101]. http://dx.doi.org/10.21949/1504477, Provided by ITS DataHub through Data.transportation.gov. Accessed 2024-04-15 from.
Nie, J., Zhang, J., Ding, W., Wan, X., Chen, X., Ran, B., 2016. Decentralized cooperative lane-changing decision-making for connected autonomous vehicles. IEEE Access 4, 9413–9420.
Nilsson, J., Gao, Y., Carvalho, A., Borrelli, F., 2014. Manoeuvre generation and control for automated highway driving. IFAC Proc. Vol. 47 (3), 6301–6306.
Nilsson, J., Silvlin, J., Brannstrom, M., Coelingh, E., Fredriksson, J., 2016. If, when, and how to perform lane change maneuvers on highways. IEEE Intell. Transp. Syst. Mag. 8 (4), 68–78.
Park, H., Oh, C., Moon, J., Kim, S., 2018. Development of a lane change risk index using vehicle trajectory data. Accid. Anal. Prev. 110, 1–8.
Paxton, C., Raman, V., Hager, G.D., Kobilarov, M., 2017. Combining neural networks and tree search for task and motion planning in challenging environments. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 6059–6066.
Powell, W.B., 2007. Approximate Dynamic Programming: Solving the curses of dimensionality, vol. 703, John Wiley & Sons.
Rajamani, R., 2011. Vehicle Dynamics and Control. Springer Science & Business Media.
Rugh, W.J., Shamma, J.S., 2000. Research on gain scheduling. Automatica 36 (10), 1401–1425.
Saikrishna, P.S., Pasumarthy, R., Bhatt, N.P., 2016. Identification and multivariable gain-scheduling control for cloud computing systems. IEEE Trans. Control Syst. Technol. 25 (3), 792–807.
Saussié, D., Saydy, L., Akhrif, O., Bérard, C., 2011. Gain scheduling with guardian maps for longitudinal flight control. J. Guid. Control Dyn. 34 (4), 1045–1059.
Shahruz, S., Behtash, S., 1992. Design of controllers for linear parameter-varying systems by the gain scheduling technique. J. Math. Anal. Appl. 168 (1), 195–217.
Shamma, J.S., 1988. Analysis and Design of Gain Scheduled Control Systems (Ph.D. thesis). Massachusetts Institute of Technology.
Shamma, J.S., Athans, M., 1990. Analysis of gain scheduled control for nonlinear plants. IEEE Trans. Autom. Control 35 (8), 898–907.
Shamma, J.S., Cloutier, J.R., 1993. Gain-scheduled missile autopilot design using linear parameter varying transformations. J. Guidance, Control, Dyn. 16 (2), 256–263.
Suh, J., Chae, H., Yi, K., 2018. Stochastic model-predictive control for lane change decision of automated driving vehicles. IEEE Trans. Veh. Technol. 67 (6), 4771–4782.
Suo, D., Jayawardana, V., Wu, C., 2024. Model-free learning of corridor clearance: A near-term deployment perspective. IEEE Trans. Intell. Transp. Syst..
Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.
Tampuu, A., Matiisen, T., Semikin, M., Fishman, D., Muhammad, N., 2020. A survey of end-to-end driving: Architectures and training methods. IEEE Trans. Neural Netw. Learn. Syst. 33 (4), 1364–1384.
Tang, X., Huang, B., Liu, T., Lin, X., 2022. Highway decision-making and motion planning for autonomous driving via soft actor-critic. IEEE Trans. Veh. Technol. 71 (5), 4706–4717.
Vamvoudakis, K.G., Kokolakis, N.M.T., et al., 2020. Synchronous reinforcement learning-based control for cognitive autonomy. Found. Trends Syst. Control 8 (1–2), 1–175.
Vidyasagar, M., 2002. Nonlinear Systems Analysis. SIAM.
Vrabie, D., Pastravanu, O., Abu-Khalaf, M., Lewis, F.L., 2009. Adaptive optimal control for continuous-time linear systems based on policy iteration. Automatica 45 (2), 477–484.

Wang, P., Chan, C.Y., de La Fortelle, A., 2018. A reinforcement learning based approach for automated lane change maneuvers. In: 2018 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 1379–1384.
Wang, S., Cui, L., Lai, J., Yang, S., Chen, X., Zheng, Y., Zhang, Z., Jiang, Z.P., 2020a. Gain scheduled controller design for balancing an autonomous bicycle. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, pp. 7595–7600.
Wang, P., Li, H., Chan, C.Y., 2019a. Continuous control for automated lane change behavior based on deep deterministic policy gradient algorithm. In: 2019 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 1454–1460.
Wang, Z., Shi, X., Li, X., 2019b. Review of lane-changing maneuvers of connected and automated vehicles: models, algorithms and traffic impact analyses. J. Indian Inst. Sci. 99 (4), 589–599.
Wang, Z., Zhao, X., Chen, Z., Li, X., 2021a. A dynamic cooperative lane-changing model for connected and autonomous vehicles with possible accelerations of a preceding vehicle. Expert Syst. Appl. 173, 114675.
Wang, Z., Zhao, X., Xu, Z., Li, X., Qu, X., 2020b. Modeling and field experiments on lane changing of an autonomous vehicle in mixed traffic. Comput.-aided Civ. Infrastruct. Eng..
Wang, Z., Zhao, X., Xu, Z., Li, X., Qu, X., 2021b. Modeling and field experiments on autonomous vehicle lane changing with surrounding human-driven vehicles. Comput.-Aided Civ. Infrastruct. Eng. 36 (7), 877–889.
Xie, D.-F., Fang, Z.Z., Jia, B., He, Z., 2019. A data-driven lane-changing model based on deep learning. Transp. Res. Part C: Emerg. Technol. 106, 41–60.
Xu, H., Gao, Y., Yu, F., Darrell, T., 2017. End-to-end learning of driving models from large-scale video datasets. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2174–2182.
Xu, G., Liu, L., Ou, Y., Song, Z., 2012. Dynamic modeling of driver control strategy of lane-change behavior and trajectory planning for collision prediction. IEEE Trans. Intell. Transp. Syst. 13 (3), 1138–1155.
Xu, M., Luo, Y., Yang, G., Kong, W., Li, K., 2019. Dynamic cooperative automated lane-change maneuver based on minimum safety spacing model. In: 2019 IEEE Intelligent Transportation Systems Conference. ITSC, IEEE, pp. 1537–1544.
Ye, F., Wang, P., Chan, C.-Y., Zhang, J., 2021. Meta reinforcement learning-based lane change strategy for autonomous vehicles. In: 2021 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 223–230.
Ye, Y., Zhang, X., Sun, J., 2019. Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment. Transp. Res. C 107, 155–170.
Yuan, C., Liu, Y., Wu, F., Duan, C., 2016. Hybrid switched gain-scheduling control for missile autopilot design. J. Guid. Control Dyn. 39 (10), 2352–2363.
Zhang, Z., Zhang, L., Deng, J., Wang, M., Wang, Z., Cao, D., 2021. An enabling trajectory planning scheme for lane change collision avoidance on highways. IEEE Trans. Intell. Veh..
Zhang, H., Zhang, X., Wang, J., 2014. Robust gain-scheduling energy-to-peak control of vehicle lateral dynamics stabilisation. Veh. Syst. Dyn. 52 (3), 309–340.
Zhang, Z., Zhang, L., Wang, C., Wang, M., Cao, D., Wang, Z., 2022. Integrated decision making and motion control for autonomous emergency avoidance based on driving primitives transition. IEEE Trans. Veh. Technol. 72 (4), 4207–4221.
Zhou, Y., Wang, M., Ahn, S., 2019. Distributed model predictive control approach for cooperative car-following with guaranteed local and string stability. Transp. Res. Part B: Methodol. 128, 69–86.
Zhu, S., Gelbal, S.Y., Aksun-Guvenc, B., Guvenc, L., 2019. Parameter-space based robust gain-scheduling design of automated vehicle lateral control. IEEE Trans. Veh. Technol. 68 (10), 9660–9671.
Zhu, Z., Zhao, H., 2021. A survey of deep RL and IL for autonomous driving policy learning. IEEE Trans. Intell. Transp. Syst. 23 (9), 14043–14065.