Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-3 (2023) 85–90

Reinforcement Learning Enabled Safety-Critical Tracking of Automated Vehicles with Uncertainties via Integrated Control-Dependent, Time-Varying Barrier Function, and Control Lyapunov Function

Jingxiong Meng, Junfeng Zhao, Yan Chen*

**Arizona State University, Mesa, AZ 85212 USA (e-mail: Jmeng18@asu.edu, junfeng.zhao@asu.edu, yanchen@asu.edu)*

Abstract: Model uncertainties are considered in a learning-based control framework that combines control dependent barrier function (CDBF), time-varying control barrier function (TCBF), and control Lyapunov function (CLF). Tracking control is achieved by CLF, while safety-critical constraints during tracking are guaranteed by CDBF and TCBF. A reinforcement learning (RL) method is applied to jointly learn model uncertainties that related to CDBF, TCBF, and CLF. The learning-based framework eventually formulates a quadratic programming (QP) with different constraints of CDBF, TCBF and CLF involving model uncertainties. It is the first time to apply the proposed learning-based framework for safety-guaranteed tracking control of automated vehicles with uncertainties. The control performances are validated for two different single-lane change maneuvers via Simulink/CarSim® co-simulation and compared for the cases with and without learning. Moreover, the learning effects are discussed through explainable constraints in the QP formulation.

Keywords: Automated vehicles, reinforcement learning, control-dependent barrier function, safety-critical tracking, and uncertainties

1. INTRODUCTION

Safe trajectory or path tracking is a crucial task in autonomous driving. A combined optimization method of control Lyapunov function (CLF) and control barrier function (CBF) provided a safety-guaranteed framework (Romdlony et al., 2016). However, model uncertainties always exist and usually degrade (if not fail) the desired control performance. Thus, how to address model uncertainties and maintain desired control performance with model uncertainties is a significant challenge for autonomous driving.

In the existing studies of safety-critical tracking control, CLF and CBF play crucial roles, in which CLF is often used for tracking control, while CBF is responsible for vehicle safety and/or stability. Writing CLF in constraints, a CLF-based quadratic problem (QP) was formulated for manipulation control (Ames et al., 2013). After that, some researchers combined CBF with CLF in constraints of QP to achieve an adaptive cruise control to balance the speed following condition via CLF and force-based constraints on acceleration and braking via CBF (Ames et al., 2014).

On the other hand, when CBF was applied to a vehicle system, the CBF method has a limitation that the vehicle stability region was considered as a fixed invariant set. However, the lateral stability region of vehicle systems will vary with respect to the control input (e.g., steering angle). Hence, a new method called control-dependent barrier function (CDBF) was proposed to resolve the issue (Huang et al., 2021). After that, the authors (Huang and Chen, 2021) also included time-varying CBF (TCBF) for constraints changing over time and then integrated CDBF, CLF, and TCBF in a QP to achieve safety-guaranteed tracking control of automated vehicles.

However, the QP method combining CLF and CBF heavily relies on model accuracy. Other methods with more robust, such as adaptive CBF, were proposed to overcome the model accuracy issues (Lopez et al., 2020; Xiao et al., 2021). Yet, these methods did not handle model uncertainties in a systematic way. Moreover, the CDBF design did not consider the uncertainties as well.

Recently, with the booming of data-driven approaches, many researchers apply learning methods to CLF and CBF to systematically handle the control issue with model uncertainties. For example, a learning-based approach for safe controller synthesis was proposed based on CBF (Robey et al., 2020). An imitation learning was used to learn neural network-based controllers that could satisfy CBF constraints with disturbances (Yaghoubi et al., 2020). Neural networks were also applied to jointly learn CBF and CLF to generate a safe and goal-reaching policy (Jin et al., 2020). To tackle model uncertainties, a machine learning framework was developed to learn model uncertainties of CLF and yield a stable CLF-based controller (Taylor et al., 2019). A learning framework was also applied to reduce model uncertainties of CBF and obtain a controller for safe behaviors (Taylor et al., 2020). Moreover, a new framework that could jointly learn model uncertainties of CBF and CLF using reinforcement learning was proposed for robot control (Choi et al., 2020). These frameworks all learned model uncertainties and then added them into the original control method.

Based on the literature review, the CDBF method can resolve CBF limitation for vehicle control, but model uncertainties are not considered. The learning-based method is powerful to tackle the uncertainty problem. Therefore, we propose a reinforcement learning (RL)-based control framework that learns uncertainties between the nominal model and plant

10.1016/j.ifacol.2023.12.005

system model. Then, the learned uncertainties will be applied in TCBF, CDBF, and CLF to generate new constraints of a QP problem, which will be finally solved to obtain control outputs. The picture of the overall framework is presented in Figure 1. We use a neural network to learn the model uncertainties. The inputs of the network are the vehicle states and corresponding rewards. The outputs of the network are model uncertainty terms, which are added to the constraints of a QP problem correspondingly to generate the control inputs.

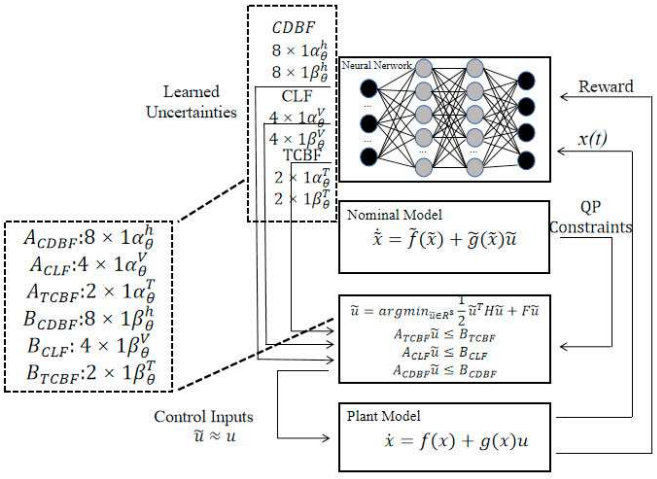

Figure 1. RL-CDBF-TCBF-CLF control framework.

Two main contributions of this paper are summarized as follows.

1) Propose a RL-enabled TCBF-CDBF-CLF control method for safety-guaranteed tracking control of automated vehicles with model uncertainties.

2) Analyze the effects of RL on TCBF-CDBF-CLF based vehicle control with model uncertainties and verify the proposed method via co-simulation between Simulink and CarSim®.

The rest of the paper is organized as follows. In Section **Error! Reference source not found.**, we briefly describe the kinematic and dynamic vehicle models, CDBF, and TCBF. Section 3 discusses how to learn model uncertainties for CLF, CDBF, and TCBF constraints, respectively. In Section 4, we present the RL settings. In Section 5, we describe the simulation setups and results and discuss the results. Section 6 makes conclusions.

2. PRELIMINARIES

2.1 Vehicle Model

Vehicle kinematic and lateral dynamic models are introduced in this section. The vehicle states are selected as $x = [Y\ \theta\ V_y\ r]^T$, where Y is the global lateral displacement, θ is the heading angle, V_y is the lateral velocity, and r is the yaw rate. The control input u consists of the front wheel steering angle δ_f and yaw moment M_{DYC}, $u = [\delta_f\ M_{DYC}]$. Vehicle lateral kinematic and dynamic models are shown in (1).

$$\begin{aligned} \dot{Y} &= V_y + V_x sin(\theta)\,, \\ \dot{\theta} &= r\,, \\ \dot{V}_y &= \frac{1}{m}\left(F_{yf}cos\delta_f + F_{yr}\right) - V_x r\,, \\ \dot{r} &= \frac{1}{I_z}\left(l_f F_{yf}cos\delta_f - l_r F_{yr} + M_{DYC}\right). \end{aligned} \tag{1}$$

$m, V_x, l_f, l_r, I_z, F_{yf},$ and F_{yr} are the vehicle mass, longitudinal velocity, front wheelbase, rear wheelbase, yaw moment of inertia, front, and rear tire lateral forces, respectively. F_{yf}, F_{yr} can be calculated by a tire model (Huang et al., 2021).

2.2 CDBF and TCBF

The CDBF definition (Huang et al., 2021) is to describe the control-dependent situation when the invariant set will change related to the change of control inputs. TCBF is used to handle time-varying constraints (Lindemann et al., 2019), such as vehicle's global lateral displacement (Y) and the heading angle.

CDBF *(Huang et al., 2021): The CDBF considers the control as a new state that augments the original system. A set ψ is defined by a continuously differentiable function h(x,u). If there exists a control $u \in U, \omega \in \Omega$ (such that $\dot{u} = \omega$ points inward of U) and an extended class K function γ, and the following inequality (2) about CDBF is satisfied, the set $\psi(x,u)$ is a control dependent invariant set and h(x,u) is a zeroing CDBF. $L_{\hat{f}}$ is the Lie derivative with respect to $\hat{f}(\hat{x},\omega)$, where $\hat{x}$ is the augmented sate of x.*

$$L_{\hat{f}}h(x,u) + \gamma\big(h(x,u)\big) \geq 0. \tag{2}$$

TCBF *(Lindemann et al, 2019): Define the differentiable function b: $\mathcal{D} \times [t0, t1] \to \mathbb{R}$, where $\mathcal{D} \subseteq \mathbb{R}^n$, the set $\mathcal{C}(t)$ should satisfy $x \in \mathcal{D}$ and $b(x,t) \geq 0$. $b(x,t)$ is a candidate control barrier function, which is changed with respect to time.*

3. METHODOLOGY

This section presents how reinforcement learning can learn the model uncertainties represented by a mismatch between the nominal and plant models. Model uncertainties of dynamics system in terms of the CDBF, TCBF, and CLF constraints are analyzed, respectively, and then combine together.

3.1 CDBF + RL

Based on the Definition 1 and the authors' previous research work (Huang et al., 2021), the CDBF design for the nominal model is written in
(3),

$$\begin{aligned} h_j(x,u) &= b_j\left[V_y - s_1(u)\right] + c_j[r - s_2(u)] + d_j\,, \\ s_1(u) &= \frac{V_x l_r}{l_r + l_f}u\,,\ s_2(u) = \frac{V_x}{l_r + l_f}u. \end{aligned} \tag{3}$$

Similar to (Huang et al., 2021), 4 constraint functions to describe a dynamic stability region are shown in (4)-(7),

$$h_1(x,u) = b_1\left[V_y - s_1(u)\right] + c_1[r - s_2(u)] + d_1\,. \tag{4}$$

$$h_2(x,u) = b_2\left[V_y - s_1(u)\right] + c_2[r - s_2(u)] + d_2\,. \tag{5}$$

$$h_3(x,u) = c_3[r - s_2(u)] - b_3\left[V_y - s_1(u)\right] + d_3\,. \tag{6}$$

$$h_4(x,u) = c_4[r - s_2(u)] - b_4\left[V_y - s_1(u)\right] + d_4\,. \tag{7}$$

Consider a nonlinear affine system (8),

$$\dot{x} = f(x) + g(x)u\,. \tag{8}$$

The nominal model is an approximation of the affine system and expressed in (9),

$$\dot{\tilde{x}} = \tilde{f}(\tilde{x}) + \tilde{g}(\tilde{x})\tilde{u}\,. \tag{9}$$

The CDBF design of the nominal model (9) should satisfy (10) based on (4)-(7), j = 1,2,3,4.

$$\dot{h}_{j\_model}(\tilde{x},\tilde{u}) = L_{\tilde{f}}h_{j\_model}(\tilde{x},\tilde{u}) + L_{\tilde{g}}h_{j\_model}(\tilde{x},\tilde{u})\tilde{u}\,. \tag{10}$$

The CDBF design of the real plant similarly should satisfy (11),

$$\dot{h}_{j\_plant}(x,u) = L_f h_{j\_plant}(x,u) + L_g h_{j\_plant}(x,u)u\,. \tag{11}$$

We want the control design of the nominal model to be still applicable to the real plant. Thus, (11) minus (10) to get (12),

$$\begin{aligned}\dot{h}_{j\_plant}(x,u) = \dot{h}_{j\_model}(\tilde{x},\tilde{u}) + (L_g h_{j\_plant}(x,u) \\ - L_{\tilde{g}}h_{j\_model}(\tilde{x},\tilde{u}))\tilde{u} \\ + L_f h_{j\_plant}(x,u) \\ - L_{\tilde{f}}h_{j\_model}(\tilde{x},\tilde{u})\,.\end{aligned} \tag{12}$$

(12) is rewritten in (13).

$$\dot{h}_{j\_plant}(x,u) = \dot{h}_{j\_model}(\tilde{x},\tilde{u}) + \Delta_1^h(x) + \Delta_2^h(x)\tilde{u}\,, \tag{13}$$

where,

$$\Delta_1^h = L_f h_j(x,u) - L_{\tilde{f}}h_j(\tilde{x},\tilde{u})\,, \tag{14}$$

$$\Delta_2^h = L_g h_j(x,u) - L_{\tilde{g}}h_j(\tilde{x},\tilde{u})\,. \tag{15}$$

Next, the reinforcement learning (RL) is applied to learn the effects of the uncertainties in (13) and get (16) below,

$$\dot{h}^{rl}_{j\_plant}(x,u) = \dot{h}^{rl}_{j\_model}(\tilde{x},\tilde{u}) + \alpha_\theta^h(x) + \beta_\theta^h(x)\tilde{u}\,. \tag{16}$$

α_θ^h and β_θ^h are learning parameters, which are a part of the action network output to learn the model uncertainties. The goal of RL is to make $\dot{h}^{rl}_{j\_model}(\tilde{x},\tilde{u})$ as close as possible to the $\dot{h}^{rl}_{j\_plant}(x,u)$.

From the constraint (2) of CDBF, the CDBF constraints of the real plant are written as (17),

$$L_{\hat{f}}h_{j\_plant}(x,u) + \gamma_j(h_{j\_plant}(x,u)) \geq 0\,. \tag{17}$$

The inequality (17)is expanded as (18),

$$\frac{\partial h_{j\_plant}(x,u)}{\partial V_y}\dot{V}_y + \frac{\partial h_{j\_plant}(x,u)}{\partial r}\dot{r} + \frac{\partial h_{j\_plant}(x,u)}{\partial \tilde{u}}\dot{u} + \gamma_j(h_{j\_plant}(x,u)) \geq 0\,. \tag{18}$$

Substituting $\dot{V}_y$ and $\dot{r}$ from the vehicle model combining with (16) into (18) gets (19),

$$\begin{aligned}\frac{\partial h_{j\_model}(\tilde{x},\tilde{u})}{\partial V_y}\tilde{f}(V_y) + \alpha_\theta^h(V_y) + \left(\frac{\partial h_{j\_model}(\tilde{x},\tilde{u})}{\partial V_y}\tilde{g}(V_y) + \beta_\theta^h(V_y)\right)\tilde{u} + \frac{\partial h_{j\_model}(\tilde{x},\tilde{u})}{\partial r}\tilde{f}(r) + \alpha_\theta^h(r) + \\ \left(\frac{\partial h_{j\_model}(\tilde{x},\tilde{u})}{\partial r}\tilde{g}(r) + \beta_\theta^h(r)\right)\tilde{u} + \frac{\partial h_{j\_model}(\tilde{x},\tilde{u})}{\partial \tilde{u}}\dot{\tilde{u}} + \\ \gamma_j(h_{j\_model}(\tilde{x},\tilde{u}) \geq 0\,.\end{aligned} \tag{19}$$

Finally, the inequality can be rewritten linearly in (20),

$$A_{CDBF}\tilde{u} \leq B_{CDBF}\,, \tag{20}$$

where,

$$A_{CDBF} = -\begin{bmatrix} \frac{\partial h_{1\_model}(\tilde{x},\tilde{u})}{\partial V_y}\tilde{g}(V_y) + \beta_\theta^{h_1}(V_y) & \frac{\partial h_{1\_model}(\tilde{x},\tilde{u})}{\partial r}\tilde{g}(r) + \beta_\theta^{h_1}(r) \\ \frac{\partial h_{2\_model}(\tilde{x},\tilde{u})}{\partial V_y}\tilde{g}(V_y) + \beta_\theta^{h_2}(V_y) & \frac{\partial h_{2\_model}(\tilde{x},\tilde{u})}{\partial r}\tilde{g}(r) + \beta_\theta^{h_2}(r) \\ \frac{\partial h_{3\_model}(\tilde{x},\tilde{u})}{\partial V_y}\tilde{g}(V_y) + \beta_\theta^{h_3}(V_y) & \frac{\partial h_{3\_model}(\tilde{x},\tilde{u})}{\partial r}\tilde{g}(r) + \beta_\theta^{h_3}(r) \\ \frac{\partial h_{4\_model}(\tilde{x},\tilde{u})}{\partial V_y}\tilde{g}(V_y) + \beta_\theta^{h_4}(V_y) & \frac{\partial h_{4\_model}(\tilde{x},\tilde{u})}{\partial r}\tilde{g}(r) + \beta_\theta^{h_4}(r) \end{bmatrix} \tag{21}$$

$$B_{CDBF} = \begin{bmatrix} \frac{\partial h_{1\_model}(\tilde{x},\tilde{u})}{\partial \tilde{x}}\tilde{f}(\tilde{x}) + \frac{\partial h_{1\_model}(\tilde{x},\tilde{u})}{\partial \tilde{u}}\dot{\tilde{u}} + \gamma_1(h_{1\_model}(\tilde{x},\tilde{u})) + \alpha_\theta^{h_1}(x) \\ \frac{\partial h_{2\_model}(\tilde{x},\tilde{u})}{\partial \tilde{x}}\tilde{f}(\tilde{x}) + \frac{\partial h_{2\_model}(\tilde{x},\tilde{u})}{\partial \tilde{u}}\dot{\tilde{u}} + \gamma_2(h_{2\_model}(\tilde{x},\tilde{u})) + \alpha_\theta^{h_2}(x) \\ \frac{\partial h_{3\_model}(\tilde{x},\tilde{u})}{\partial \tilde{x}}\tilde{f}(\tilde{x}) + \frac{\partial h_{3\_model}(\tilde{x},\tilde{u})}{\partial \tilde{u}}\dot{\tilde{u}} + \gamma_3(h_{3\_model}(\tilde{x},\tilde{u})) + \alpha_\theta^{h_3}(x) \\ \frac{\partial h_{4\_model}(\tilde{x},\tilde{u})}{\partial \tilde{x}}\tilde{f}(\tilde{x}) + \frac{\partial h_{4\_model}(\tilde{x},\tilde{u})}{\partial \tilde{u}}\dot{\tilde{u}} + \gamma_4(h_{4\_model}(\tilde{x},\tilde{u})) + \alpha_\theta^{h_4}(x) \end{bmatrix} \tag{22}$$

3.2 TCBF + RL

In real traffic scenarios, some practical driving constraints need to be considered. For example, the heading angle and global lateral displacement should be bounded in a reasonable range to avoid vehicle drift or spin. Thus, these time-varying geometric constraints need to be addressed.

Safety sets of the heading angle and global lateral displacement can be defined in (23) and (24) (Huang et al., 2021),

$$\psi_1 = \{x|h_5(x,t) = (Y_{max}(t) - Y)(Y - Y_{min}(t)) \geq 0\}\,. \tag{23}$$

$$\psi_2 = \{x|h_6(x,t) = (\theta_{max}(t) - \theta)(\theta - \theta_{min}(t)) \geq 0\}\,. \tag{24}$$

The above safety conditions can be described via TCBF in (25) and (26) (Huang et al., 2021),

$$\begin{aligned}h_5(x,t) = (Y_{max} + Y_{min} - 2Y)\big(V_y + V_x\theta\big) + \big(\dot{Y}_{max} + \dot{Y}_{min}\big)Y - \big(\dot{Y}_{max}Y_{min} + Y_{max}\dot{Y}_{min}\big) + \\ \alpha_1\Big((Y_{max}(t) - Y)\big(Y - Y_{min}(t)\big)\Big) \geq 0\,.\end{aligned} \tag{25}$$

$$\begin{aligned}h_6(x,t) = (\theta_{max} + \theta_{min} - 2\theta)r + (\dot{\theta}_{max} + \dot{\theta}_{min})\theta - (\dot{\theta}_{max}\theta_{min} + \dot{\theta}_{min}\theta_{max}) + \\ \alpha_2((\theta_{max}(t) - \theta)(\theta - \theta_{min}(t))) \geq 0\,.\end{aligned} \tag{26}$$

Similar to the process of (16) to (20), the inequality of TCBF is obtained. Since the original TCBF did not consider the model uncertainties, the new version with model uncertainties is shown as follows.

$$A_{TCBF}\tilde{u} \leq B_{TCBF}\,, \tag{27}$$

where,

$$A_{TCBF} = -\begin{bmatrix} \frac{\partial h_5(\tilde{x},t)}{\partial V_y}\tilde{g}\big(V_y\big) + \beta_\theta^{T1} \\ \frac{\partial h_6(\tilde{x},t)}{\partial r}\tilde{g}(r) + \beta_\theta^{T2} \end{bmatrix}, \tag{28}$$

$$B_{TCBF} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \tag{29}$$

$$\begin{aligned}B_1 = \frac{\partial h_5(\tilde{x},t)}{\partial V_y}\tilde{f}\big(V_y\big) + \frac{\partial h_5(\tilde{x},t)}{\partial \theta}\dot{\theta} + \frac{\partial h_5(\tilde{x},t)}{\partial t} + \\ \frac{\partial h_5(\tilde{x},t)}{\partial Y}\dot{Y} + \alpha_1(Y_{max}(t) - Y)\big(Y - Y_{min}(t)\big) + \alpha_\theta^{T1}\,,\end{aligned} \tag{30}$$

$$\begin{aligned}B_2 = \frac{\partial h_6(\tilde{x},t)}{\partial r}\tilde{f}(r) + \frac{\partial h_6(\tilde{x},t)}{\partial \theta}\dot{\theta} + \frac{\partial h_6(\tilde{x},t)}{\partial t} + \\ \alpha_2(\theta_{max}(t) - \theta)(\theta - \theta_{min}(t)) + \alpha_\theta^{T2}\,.\end{aligned} \tag{31}$$

3.3 CLF + RL

The CLF tracks the reference path by minimizing the tracking errors. The references of lateral displacement and heading angle are Y_{ref} and θ_{ref}, and the tracking errors are written as $e_Y = Y - Y_{ref}$ and $e_\theta = \theta - \theta_{ref}$. Since the lateral

displacement and heading angle cannot be controlled directly, V_y and r are used to represent the tracking performance. The desired V_{yd} and r_d are obtained from tracking requirements of Y_{ref} and θ_{ref}, described in (32) and (33), k > 0,

$$V_{yd} = -ke_Y + \dot{Y}_{ref} - V_x\theta\,. \tag{32}$$

$$r_d = -ke_\theta + \dot{\theta}_{ref}\,. \tag{33}$$

The CLFs are constructed to track the desired V_{yd} and r_d, which are written in (34) and (35),

$$V_1(x) = \left(V_y - V_{yd}\right)^2. \tag{34}$$

$$V_2(x) = (r - r_d)^2. \tag{35}$$

The constructed CLFs need to satisfy certain conditions for tracking purposes (Ames et al., 2013). For the affine system (8), a continuously differentiable function $V: X \to \mathbb{R}$ is an exponentially stabilizing CLF if there exist positive constants c_1, c_2 and $c_3 > 0$ such that (36) and (37) hold.

$$c_1\|x\|^2 \le V(x) \le c_2\|x\|^2. \tag{36}$$

$$inf_{u\in U}\left[L_fV(x) + L_gV(x)u + c_3V(x)\right] \le 0\,. \tag{37}$$

Following the process of the previous derivation from (11)-(16), the formation of $\dot{V}^{rl}_{i\_plant}(x)$ is obtained in (38), i = 1, 2.

$$\dot{V}^{rl}_{i\_plant}(x) = \dot{V}^{rl}_{i\_model}(\tilde{x}) + \alpha_\theta^{v_i}(x) + \beta_\theta^{v_i}(x)\tilde{u}\,. \tag{38}$$

α_θ^v and β_θ^v are learning parameters, which are a part of the action network output to learn model uncertainties. Here, a specific mathematical expression for $\dot{V}^{rl}_{i\_plant}(x)$ is not needed since the values are obtained from a high-fidelity simulation software, CarSim®. Next, plug (38) into the constraint (37), then (37) is rewritten as (39),

$$\begin{aligned} inf_{u\in U}\{&L_{\tilde{f}}V(\tilde{x},z) + \alpha_\theta^{v_i}(x) + (L_{\tilde{g}}V(\tilde{x},z) + \beta_\theta^{v_i}(x))\tilde{u} \\ &+ cV(x,z)\} \le 0\,. \end{aligned} \tag{39}$$

Finally, the expression of (40) is obtained,

$$A_{CLF}\tilde{u} \le B_{CLF}, \tag{40}$$

where,

$$A_{CLF} = -\begin{bmatrix} L_{\tilde{g}}V_1(\tilde{x},z) + \beta_\theta^{v_1}(x) \\ L_{\tilde{g}}V_2(\tilde{x},z) + \beta_\theta^{v_2}(x) \end{bmatrix}. \tag{41}$$

$$B_{CLF} = \begin{bmatrix} L_{\tilde{f}}V_1(\tilde{x},z) + cV_1(x,z) + \alpha_\theta^{v_1}(x) \\ L_{\tilde{f}}V_2(\tilde{x},z) + cV_2(x,z) + \alpha_\theta^{v_2}(x) \end{bmatrix}. \tag{42}$$

3.4 CDBF-TCBF-CLF + RL

Combining the obtained CLF, CDBF, and TCBF constraints with model uncertainties, the QP problem is described in (43),

$$\tilde{u} = argmin_{\tilde{u}\in R^3}\frac{1}{2}\tilde{u}^TH\tilde{u} + F\tilde{u}\,. \tag{43}$$

Subject to,

$$A_{CLF}\tilde{u} \le B_{CLF}\,. \tag{44}$$

$$A_{CDBF}\tilde{u} \le B_{CDBF}\,. \tag{45}$$

$$A_{TCBF}\tilde{u} \le B_{TCBF}\,. \tag{46}$$

$H \in R^{n\times n}, F \in R^n$ are weightings selected according to the desired control input.

4. REINFORCEMENT LEARNING FRAMEWORK

In this section, the RL framework is designed to learn model uncertainties in the constraints of CDBF, TCBF, and CLF. This framework is shown in Figure 1. The RL networks receive the states of the dynamic vehicle system as the inputs. Then, uncertainty parameters α_θ^h, β_θ^h, α_θ^T, β_θ^T, α_θ^v, and β_θ^v are output to combine with CLF, TCBF, and CDBF constraints, designed from the nominal model. The combined terms are considered as constraints in the QP to generate control inputs that satisfy these constraints all the time.

The reward function needs to be carefully designed for the RL agent, which is closely related to the desired goal, which is to minimize the estimation error. Hence, the reward function is represented in (47),

$$R(x,\theta) = -\sum_{i=1}^{n_v}\omega_{v_i}l_{v_i} - \sum_{j=1}^{n_h}\omega_{h_j}l_{h_j}, \tag{47}$$

where l_v is the loss function of CLF (48), l_h is the loss function of TCBF and CDBF (49), ω_v and ω_h are selected constants.

$$l_v = \left\|V_y - V_{yd}\right\|^2 + \|r - r_d\|^2. \tag{48}$$

$$l_h = \left\|\dot{h}_{rl} - \dot{h}_{plant}\right\|^2. \tag{49}$$

In (49), $\dot{h}_{rl}$ is the derivative of barrier functions designed from the nominal model and $\dot{h}_{plant}$ is the derivative of barrier functions designed from the real plant.

The learning problem is then defined in (50).

$$\max_\theta exp\int_0^T R(x(\tau),\theta)d\tau, \tag{50}$$

subject to (9).

5. SIMULATION RESULTS AND DISCUSSIONS

5.1 Simulation Settings

A high-speed single-lane change scenario on a high-μ (μ=0.85) road is applied to verify the proposed control design. The vehicle mass is m=1270 kg, the yaw moment of inertia is 1500 Nm and the front and rear wheelbase are 1.11 m and 1.8 m, respectively. The RL agent was trained by using deep deterministic policy gradient (DDPG) (Lillicrap et al., 2016). The input of the actor network is the vehicle states. The output is the 28 dimensional uncertainty terms. Both the observation network and action network have 4 hidden layers and every hidden layer has 15 neurons.

A high-fidelity vehicle model in CarSim® is used as the plant model and the mathematical model (1) is the nominal model. Thus, a mismatch between the nominal and plant models introduces the model uncertainties. The 28 learning parameters are used to calculate the uncertainties even if we do not know the specific sources of the uncertainties.

We design two maneuvers to compare the original method and the RL method. The only difference between the two maneuvers is the vehicle mass. The first maneuver design is the vehicle mass m=1270kg to finish the single lane change. Moreover, we train and test the RL network in this maneuver. Several attempts were made to train the network until it converges and then the trained network was used in the first maneuver. We test at the other maneuver that vehicle mass becomes m=1370 kg in CarSim®, while the mass in the nominal model is still 1270 kg. We want to increase the

model uncertainties to sharpen the comparison of the results and test these two methods' robustness.

5.2 Simulation Results and Discussions

In Figure 2, the vehicle controlled by the TCBF-CDBF-CLF-RL method tracks the reference path of a single lane change better than the TCBF-CDBF-CLF without learning. First, the response time with learning is faster (1.5 seconds vs. 2.5 seconds). Second, the overshooting (magnitude of deviation from the reference path) is lower for the learning method. Finally, the convergence time of the proposed method is much quicker than the original method without learning (13 seconds vs. more than 16 seconds).

The improved tracking performance can be explained from the control inputs in Figure 3. Around 1 second, the steering angle from the proposed method is larger than the original method, so that the vehicle will make a faster lane change than the original method. From 7 to 11 seconds, the RL method's steering angle keeps larger than the original method, making the vehicle turn back to the reference path faster. Furthermore, some insights can be also seen from vehicle states in Figure 4. The vehicle states controlled by the RL method change more smoothly than the original method, which reflects that the vehicle trajectory is smoother for the RL method.

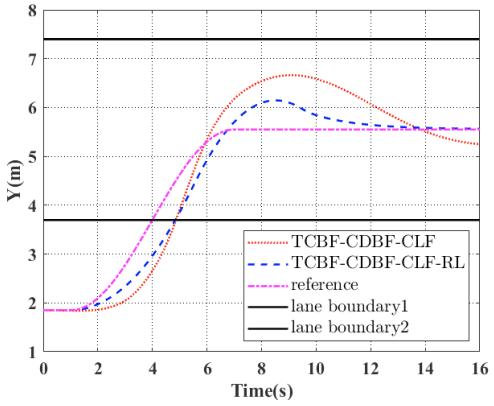

Figure 2. Vehicle tracking performance comparison for a single lane change maneuver.

To provide further analysis and understand how the RL method helps control with model uncertainties, the control constraints in the QP are investigated. In Figure 5 and Figure 6, we present the constraint region of control inputs around T=5s and 9s as examples. In Figure 5, the TCBF constraints are shown to influence the control performance most. We can see that the control region has been changed after adding the learned uncertainties. From mathematical perspective, according to the QP constraint $Au \leq B$, the relationship between u_1 and u_2 can be expanded as $A_{11}u_1 - B_1 \leq -A_{12}u_2$ and $A_{21}u_1 - B_2 \leq -A_{22}u_2$. The learned uncertainty terms can affect the feasible region by affecting the sign of A_{12} and A_{22}. In Figure 6, we can find that the original control point without learning is above the TCBF boundary, while the RL control point is below the TCBF boundary. That means the feasible domain of the control inputs has been changed after adding the learned uncertainties, which may allow the vehicle to select more reasonable actions. From Figure 6, we plot parts of the CDBF constraints, which influence the performance of the control. We can find that the learned uncertainties will expand or shift the original control region. Some control points that are out of the original region but bounded in the new region will give the vehicle more optimization rooms to improve the tracking performance.

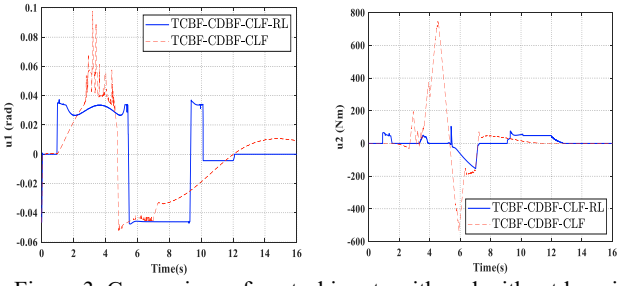

Figure 3. Comparison of control inputs with and without learning.

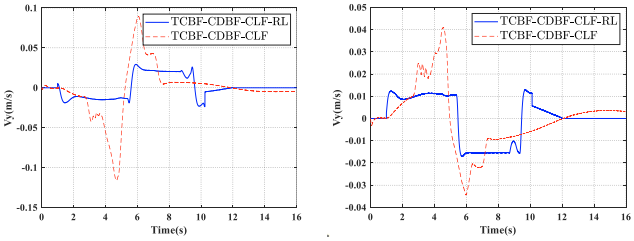

Figure 4. Comparison of vehicle states with and without learning.

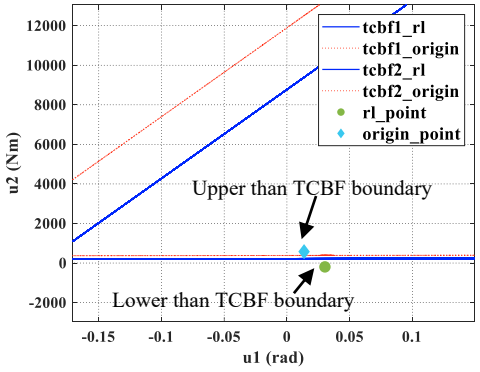

Figure 5. Control output (point) in the constraint region at simulation time T=5s.

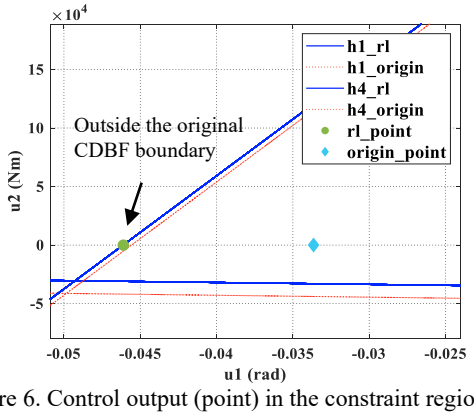

Figure 6. Control output (point) in the constraint region at simulation time T=9s.

Next, the second test is applied to test the robustness of the proposed algorithm. In this case, we change the vehicle mass to m=1370 kg and the other parameters keep the same as those of the first test. By changing the vehicle mass, the

vehicle will be more challenging to control since the controller without learning does not know the mass variation. From Figure 7, the proposed RL-based method can control the vehicle. However, the original method without learning vehicle mass change (or due to model uncertainties) cannot complete the single lane change maneuver. The two control outputs are shown in Figure 8. A similar analysis can be conducted as described for the previous case. For steering angle control, around 1-3 seconds, the RL control method generates a larger steering angle so that the vehicle will respond quicker and track the reference trajectory. In the process, the magnitudes of steering angle and yaw moment are smaller than those of the original method. The vehicle controlled by the RL method will not choose sharp or dangerous actions, making it less likely to lose control and better track the reference path.

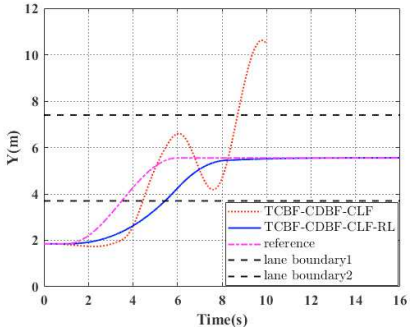

Figure 7. Vehicle tracking performance comparison with unknown vehicle mass.

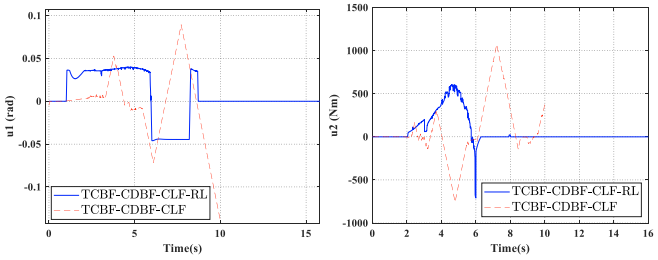

Figure 8. Comparison of control inputs for a varied vehicle mass with and without learning.

6. CONCLUSIONS

In this paper, RL is integrated to CLF, CDBF, and TCBF to solve the model uncertainties to generate a more reasonable vehicle tracking strategy. A vehicle lateral stability control problem is formulated. The algorithm is validated by two single lane change maneuvers in CarSim® and Simulink co-simulation. The RL based control outperforms the original control in the two maneuvers.

REFERENCES

Romdlony, M. Z., and Jayawardhana, B. (2016). Stabilization with guaranteed safety using control Lyapunov–barrier function. *Automatica*, 66,pp. 39-47.

Ames, A. D., and Powell, M. (2013). Towards the unification of locomotion and manipulation through control Lyapunov functions and quadratic programs. *In Control of Cyber-Physical Systems: Workshop held at Johns Hopkins University*, pp. 219-240.

Ames, A. D., Grizzle, J. W., and Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. *In 53rd IEEE Conference on Decision and Control.* pp. 6271-6278.

Lopez, B. T., Slotine, J. J. E., and How, J. P. (2020). Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters,* Vol.5, No.3, pp.1031-1036.

Xiao, W., Belta, C., and Cassandras, C. G. (2021). Adaptive control barrier functions. *IEEE Transactions on Automatic Control,* Vol.67, No.5, pp.2267-2281.

Huang, Y., Yong, S. Z., Chen, Y. (2021). Stability Control of Autonomous Ground Vehicles Using Control-Dependent Barrier Functions. *IEEE Transactions on Intelligent Vehicles*. pp.699-710

Huang, Y., and Chen, Y. (2021). Safety-guaranteed driving control of automated vehicles via integrated CLFs and CDBFs. *IFAC-Papers Online*, Vol.54, No.20, pp.153-159.

Robey, A., Hu, H., Lindemann, L., Zhang, H., Dimarogonas, D. V., Tu, S., and Matni, N. (2020). Learning control barrier functions from expert demonstrations. *In 2020 59th IEEE Conference on Decision and Control (CDC)*, pp. 3717-3724.

Yaghoubi, S., Fainekos, G., and Sankaranarayanan, S. (2020). Training neural network controllers using control barrier functions in the presence of disturbances. *In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp.1-6.

Jin, W., Wang, Z., Yang, Z., and Mou, S. (2020). Neural certificates for safe control policies. *ArXiv, abs/2006.08465*.

Taylor, A. J., Dorobantu, V. D., Le, H. M., Yue, Y., and Ames, A. D. (2019). Episodic learning with control lyapunov functions for uncertain robotic systems. *In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6878-6884.

Taylor, A., Singletary, A., Yue, Y., and Ames, A. (2020). Learning for safety-critical control with control barrier functions. *In Learning for Dynamics and Control,* pp. 708-717.

Choi, J., Castaneda, F., Tomlin, C. J., and Sreenath, K. (2020). Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions. *ArXiv, abs/2004.07584.*

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., and Wierstra, D. (2016). Continuous control with deep reinforcement learning. *ICLR*

Lindemann, L. and Dimarogonas, D. V. (2019). Control barrier functions for signal temporal logic tasks. *IEEE. Control System Letters,* Vol. 3, No. 1, pp. 96-101.