Correctness Guarantees for the Composition of Lane Keeping and Adaptive Cruise Control

Xiangru Xu, Jessy W. Grizzle, Paulo Tabuada, Aaron D. Ames

***Abstract*—This paper develops a control approach with correctness guarantees for the simultaneous operation of lane keeping and adaptive cruise control. The safety specifications for these driver assistance modules are expressed in terms of set invariance. Control barrier functions are used to design a family of control solutions that guarantee the forward invariance of a set, which implies satisfaction of the safety specifications. The control barrier functions are synthesized through a combination of sum-of-squares program and physics-based modeling and optimization. A real-time quadratic program is posed to combine the control barrier functions with the performance-based controllers, which can be either expressed as control Lyapunov function conditions or as black-box legacy controllers. In both cases, the resulting feedback control guarantees the safety of the composed driver assistance modules in a formally correct manner. Importantly, the quadratic program admits a closed-form solution that can be easily implemented. The effectiveness of the control approach is demonstrated by simulations in the industry-standard vehicle simulator Carsim.**

***Index Terms*—Correct-by-construction, Control barrier functions, Safety, Quadratic program, Sum of squares optimization**

I. Introduction

Recent years have witnessed a growing number of safety or convenience modules for automobiles [1], [2]. Lane keeping, also called active lane keeping or lane keeping assist, is an evolution of lane departure warning [3], [4], [5], where instead of simply warning of imminent lane departure through vibration of the steering wheel or an audible alarm, the system corrects the vehicle's direction to keep it within its lane. Early systems corrected vehicle direction by differential braking, but current systems actively control steering to maintain lane centering, which is what we will term Lane Keeping (LK) in this paper. Another such system is Adaptive Cruise Control (ACC), which is a driver assistance system that significantly enhances conventional cruise control [6]. When there is no preceding vehicle, an ACC-equipped vehicle maintains a constant speed set by the driver, just as in conventional cruise control; when a preceding vehicle is detected and is driving at a speed slower than the preset speed, an ACC-equipped vehicle changes its control objective to maintaining a safe following distance. ACC uses deceleration/acceleration bounds that are much less than a vehicle's maximal capabilities to ensure driver comfort; when the ACC specification (such as the minimum time-headway) cannot be maintained with comfort-based deceleration bounds, almost all vehicles equipped with ACC either have a warning system or an emergency braking system. Though ACC is legally considered a convenience feature, its specification is still considered as a "safety constraint" in this paper, consistent with how an OEM would treat it.

Worldwide, most of the major manufacturers are now offering passenger vehicles equipped with ACC and LK. Moreover, these features can be activated simultaneously at highway speeds and require limited driver supervision. In terms of the levels of automation defined by the National Highway Traffic Safety Administration (NHTSA) [7], such vehicles are already at Level 2 (automation of at least two primary control functions designed to work in unison without driver intervention), and they are approaching Level 3 (limited self-driving). The simultaneous control of the longitudinal and lateral dynamics of a vehicle is an important milestone in the bottom-up approach to full autonomy, and therefore, it is crucial to prove that the controllers associated with ACC and LK behave in a formally correct way when they are both activated.

Applying formal methods to the field of (semi-)autonomous driving has attracted much attention in recent years [8],[9],[10]. Particularly, formal correctness guarantees on individual ACC or LK system have been developed by various means. For example, the verification of cruise control systems has been accomplished using formal methods such as satisfiability modulo theory, theorem proving [11] and a counter-example guided approach [12]; safety guarantees for LK have been established using Lyapunov stability analysis by assuming the longitudinal speed is constant [13], [14] or varying [3], [15]. Furthermore, the so-called correct-by-construction control design, which aims to synthesize controllers to guarantee the closed-loop system satisfies the specification by construction and hence eliminating the need for verification, has also emerged as a viable means of achieving safety. For example, in [16], two provably correct control design methods were proposed for ACC, which rely on fixed-point computations of certain set-valued mappings on the continuous state space or a finite-state abstraction, respectively.

For the simultaneous operation of two (or even more) safety or convenience modules, which are typically coupled through the vehicle's dynamics, it is more challenging to establish correctness guarantees. A contract-based design method employing assume-guarantee reasoning is a potential recipe for

This research is supported by NSF CPS Award 1239037.

X. Xu and J.W. Grizzle are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. email: xuxiangr@umich.edu, grizzle@umich.edu.

P. Tabuada is with the Department of Electrical Engineering, University of California, Los Angles, CA, USA. email: tabuada@ucla.edu.

A. D. Ames is with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA, USA. email: ames@caltech.edu.

the compositional design of complex systems [17], [18]. Its main idea is to formally define the assumptions and guarantees of each subsystem, which are called contracts among the subsystems, and based on the contracts, to design (or establish) formal guarantees on the overall system. Different types of assume-guarantee formalisms have been proposed, such as temporal logic formulas [19] and supply/demand rates [20]. The composition of LK and ACC has been studied on the basis of contracts. A passivity-based approach was proposed in [21], where the ACC and LK dynamics are described as multi-modal port Hamiltonian systems, based on which some energy functions are constructed to prove trajectories of the composed system do not enter a specified unsafe region. In [22], the system dynamics are represented as discrete-time linear parameter-varying systems, and contracts are established for the variables that couple the two subsystems; controlled-invariant sets are constructed for the ACC and LK subsystems individually to meet the terms of the contracts using an iterative algorithm, such that the overall controller is guaranteed to ensure the safety of the composed system. Despite these very interesting initial contributions, many safety guarantee problems on the composition of LK and ACC are still largely open and deserve further investigation.

Turning now to the more general literature on safety specifications, when safety is expressed as set invariance, controlled invariant sets are used to encode both the correct behavior of the closed-loop system and a set of feedback control laws that will achieve it (see [23], [24], [25], [26] and references therein). Under the name of *invariance control*, [23] and [24] extended Nagumo's Theorem to allow higher-order derivative conditions on the boundary of the controlled invariant set. As an add-on control scheme, *reference and command governors* utilize the notion of controlled invariance to enforce constraint satisfaction and ensure that the modified reference command is as close as possible to the original reference command [25]. In [26], characterization of the controlled invariant set for LK was given, which was then used to derive a feedback control strategy to maintain a vehicle in its lane. The common intent of these methods is to construct a controlled invariant set that encodes the safety specifications, and then construct a feedback law that ensures that trajectories of the controlled systems are confined within the set.

A barrier function (certificate) is another means to prove a safety property of a system based on set invariance. It seeks a function whose sub-level sets (or super-level sets, depending on the context) are all invariant, without the difficult task of computing the system's reachable set [27], [28]. In [29], the barrier condition in [27] was relaxed by only requiring a single super-level set of the function, which represents the safe region, to be invariant. A control barrier function (CBF) extends barrier functions from dynamical systems to control systems. When CBFs are unified with a performance controller, which can be expressed as a control Lyapunov function (CLF) condition or a legacy controller, through a quadratic programming (QP) framework, safety can be always guaranteed while the performance objective is overridden when safety and performance are in conflict (see Figure 1). The QP-based approach was introduced in [29] where it was applied to ACC safety control design. Further work along this line is available in [30], [31], [32], [33], [34].

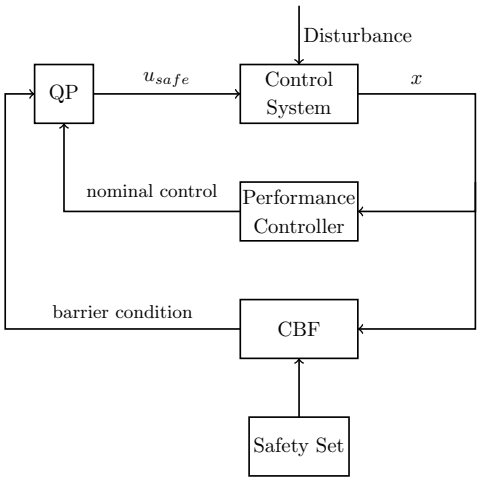

Fig. 1. The QP-based framework that unifies the safety constraint and the control performance. The controller u_{safe} is designed to satisfy the barrier condition, which ensures the control system satisfy the safety constraint (i.e., the state x stays within the safety set), and is as close as possible to the given nominal control, which makes the control system achieve the performance objective when it is not conflicting with the safety constraint.

This paper develops a modular correct-by-construction control approach that allows for individual and simultaneous activation of LK and ACC, whose dynamics are described by a continuous-time linear system and a linear parameter-varying system, respectively. The interactions of the two modules through the dynamics of the vehicle and the environment are captured in a "contract", which consists of a set of assumptions and guarantees. CBFs are constructed to respect the contract where the CBF for LK is synthesized with the help of sum-of-squares (SOS) optimization [35], [36], [37] and the CBF for ACC is constructed by physics-based modeling and optimization. Through a QP that unifies the CBF and the performance controller, the designed controller mediates the safety and performance and is correct by construction under a clearly delineated set of assumptions. The practicality of the approach will be illustrated through calculations and simulation on a model of a mid-size passenger vehicle.

The rest of the paper is organized as follows. Section II introduces the definition and some theoretic results about control barrier functions. Section III gives the dynamical models and the safety specifications for LK and ACC, based on which the composition problem studied in the paper is formulated and the assumptions and guarantees between LK and ACC are given. Section IV constructs CBFs for LK and ACC, respectively. Section V gives the QP-based framework, based on which the input that solves the composition problem is provided. Simulation results are presented in Section VI and finally, some conclusions in Section VII.

Notation. The boundary and the interior of a set $\mathcal{S}$ are denoted as $\partial\mathcal{S}$ and $\mathrm{Int}(\mathcal{S})$ (or $\mathring{\mathcal{S}}$), respectively. The commutative ring of real valued polynomials in n variables $x_1, ..., x_n$ is denoted as $\mathcal{R}[x_1, ..., x_n]$, and as $\mathcal{R}_m[x_1, ..., x_n]$ if its degree is m. The set of sum of squares polynomials in n variables

$x_1, ..., x_n$ is denoted as $\Sigma[x_1, ..., x_n]$, and as $\Sigma_m[x_1, ..., x_n]$ if its degree is m.

II. Preliminaries on Control Barrier Functions

This section introduces the background on control barrier functions that will be used later.

Consider a nonlinear system on $\mathbb{R}^n$,

$$\dot{x} = f(x), \tag{1}$$

with f locally Lipschitz continuous. The solution of (1) with initial condition $x_0 \in \mathbb{R}^n$ is denoted by $x(t, x_0)$ (or simply $x(t)$). A set $\mathcal{S}$ is called *forward invariant* if for every $x_0 \in \mathcal{S}$, $x(t, x_0) \in \mathcal{S}$ for all $t \in I(x_0)$, where $I(x_0)$ is the *maximal interval of existence* of $x(t, x_0)$.

Given a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, define a closed set $\mathcal{C}$ as follows

$$\mathcal{C} = \{x \in \mathbb{R}^n : h(x) \geq 0\}. \tag{2}$$

In what follows, it will also be assumed that $\mathcal{C}$ is nonempty and has no isolated points, that is, $\mathrm{Int}(\mathcal{C}) \neq \emptyset$ and $\overline{\mathrm{Int}(\mathcal{C})} = \mathcal{C}$. The Lie derivative of $h(x)$ along the vector field $f(x)$ is denoted as $L_f h(x)$, that is, $L_f h(x) = \frac{\partial h}{\partial x} f(x)$.

Definition 1. *[30] Consider a dynamical system* (1) *and the set $\mathcal{C}$ defined by* (2) *for some continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$. If there exist a constant $\gamma > 0$ and a set $\mathcal{D}$ with $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^n$ such that*

$$L_f h(x) \geq -\gamma h(x), \forall\, x \in \mathcal{D}, \tag{3}$$

then the function h is called a (zeroing) barrier function.

Existence of a (zeroing) barrier function implies the forward invariance of $\mathcal{C}$, as shown by the following theorem.

Theorem 1. *[30] Given a dynamical system* (1) *and a set $\mathcal{C}$ defined by* (2) *for some continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, if h is a barrier function defined on the set $\mathcal{D}$ with $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^n$, then $\mathcal{C}$ is forward invariant.*

Consider an affine control system of the following form

$$\dot{x} = f(x) + g(x)u, \tag{4}$$

with f and g locally Lipschitz continuous, $x \in \mathbb{R}^n$ and $u \in U \subset \mathbb{R}^m$.

Definition 2. *[30] Given a set $\mathcal{C} \subset \mathbb{R}^n$ defined by* (2) *for a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$, the function h is called a (zeroing) control barrier function defined on set $\mathcal{D}$ with $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^n$, if there exists a constant $\gamma > 0$ such that*[1]

$$\sup_{u \in U} [L_f h(x) + L_g h(x) u + \gamma h(x)] \geq 0,\ \forall x \in \mathcal{D}. \tag{5}$$

Given a CBF h, for all $x \in \mathcal{D}$, define the set

$$K_{\mathrm{zcbf}}(x) = \{u \in U : L_f h(x) + L_g h(x) u + \gamma h(x) \geq 0\}. \tag{6}$$

The following result guarantees the forward invariance of $\mathcal{C}$ when inputs are selected from $K_{\mathrm{zcbf}}(x)$.

[1]A more general definition for the (zeroing) CBFs that involves extended class $\mathcal{K}$ functions can be found in [30].

Theorem 2. *[30] Assume given a set $\mathcal{C} \subset \mathbb{R}^n$ defined by* (2) *for a continuously differentiable function h. If h is a CBF on $\mathcal{D}$, then any locally Lipschitz continuous controller $u : \mathcal{D} \to U$ such that $u(x) \in K_{\mathrm{zcbf}}(x)$ will render the set $\mathcal{C}$ forward invariant.*

In some cases, seeking a CBF that is everywhere continuously differentiable can be too restrictive. Below, it is briefly pointed out how the assumption of continuous differentiability can be relaxed to a continuous function constructed from a finite set of continuously differentiable functions.

Consider p ($p \geq 2$) continuously differentiable functions $h_1, ..., h_p$ where $h_i : \mathbb{R}^n \to \mathbb{R}$. Assume that the sets $\mathcal{C}_i := \{x \in \mathbb{R}^n : h_i(x) \geq 0\}$ satisfy $\mathrm{Int}(\mathcal{C}_i) \neq \emptyset$ and $\overline{\mathrm{Int}(\mathcal{C}_i)} = \mathcal{C}_i$. Suppose that $\mathbb{R}^n$ is partitioned into p closed sets $\mathcal{S}_1, ..., \mathcal{S}_p$ such that $\cup_{i=1}^p \mathcal{S}_i = \mathbb{R}^n$ and $\mathrm{Int}(\mathcal{S}_i) \cap \mathrm{Int}(\mathcal{S}_j) = \emptyset$, $\forall i, j, i \neq j$. For any i, j such that $\mathcal{S}_i \cap \mathcal{S}_j \neq \emptyset$, assume that $h_i(x) = h_j(x)$ for $x \in \mathcal{S}_i \cap \mathcal{S}_j$. Then the function $h : \mathbb{R}^n \to \mathbb{R}$ by

$$h|_{\mathcal{S}_i}(x) = h_i(x) \tag{7}$$

is well defined and continuous, and the set $\mathcal{C} = \{x \in \mathbb{R}^n | h(x) \geq 0\}$ is closed, has non-empty interior, and does not have isolated points.

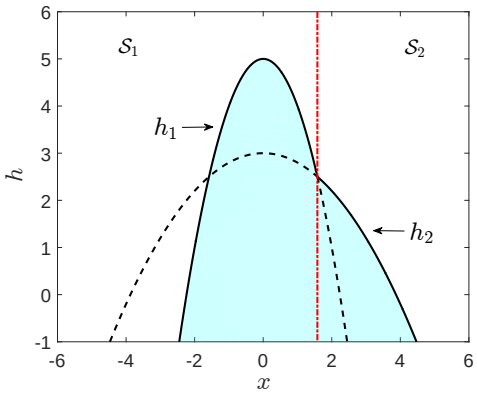

Fig. 2. The safe set $\mathcal{C}$ defined in Example 1 is the shaded region.

Example 1. *Consider two functions $h_1(x) = -x^2 + 5$, $h_2(x) = -0.2x^2 + 3$ and the partition $\mathcal{S}_1 = (-\infty, \sqrt{10}/2]$, $\mathcal{S}_2 = [\sqrt{10}/2, \infty)$. Clearly, $\mathrm{Int}(\mathcal{S}_1) \cap \mathrm{Int}(\mathcal{S}_2) = \emptyset$, $\mathcal{S}_1 \cup \mathcal{S}_2 = \mathbb{R}$ and $h_1(x) = h_2(x)$ when $x = \sqrt{10}/2$. Then, the safe set $\mathcal{C} := \{x | h(x) \geq 0\}$ with $h(x)$ defined in* (7) *is shown as the shaded area in Figure 2.*

For all $x \in \mathcal{S}_i$ ($1 \leq i \leq p$), suppose that there exists a constant $\gamma_i > 0$ such that

$$\sup_{u \in U} [L_f h_i(x) + L_g h_i(x) u + \gamma_i h_i(x)] \geq 0,$$

and define the set $K^i(x)$ as

$$K^i(x) = \{u \in U | L_f h_i(x) + L_g h_i(x) u + \gamma_i h_i(x) \geq 0\}.$$

For any $x \in \mathbb{R}^n$, define the set $K(x)$ as

$$K(x) = \cap_{s \in \{i | x \in S_i\}} K^s(x). \tag{8}$$

$$\begin{pmatrix} \dot{y} \\ \dot{\nu} \\ \dot{\Delta\psi} \\ \dot{r} \end{pmatrix} = \begin{pmatrix} 0 & 1 & v_f & 0 \\ 0 & -\frac{C_f+C_r}{mv_f} & 0 & \frac{bC_r-aC_f}{mv_f} - v_f \\ 0 & 0 & 0 & 1 \\ 0 & \frac{bC_r-aC_f}{I_z v_f} & 0 & -\frac{a^2C_f+b^2C_r}{I_z v_f} \end{pmatrix} \begin{pmatrix} y \\ \nu \\ \Delta\psi \\ r \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{C_f}{m} \\ 0 \\ \frac{aC_f}{I_z} \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \end{pmatrix} d. \tag{9}$$

$$\begin{pmatrix} \dot{v}_f \\ \dot{v}_l \\ \dot{D} \end{pmatrix} = \begin{pmatrix} -\frac{c_0+c_1 v_f}{m} \\ a_L \\ v_l - v_f \end{pmatrix} + \begin{pmatrix} \frac{1}{m} \\ 0 \\ 0 \end{pmatrix} u_2 + \begin{pmatrix} -\nu r \\ 0 \\ 0 \end{pmatrix}. \tag{10}$$

The following proposition extends Theorem 2 to the case of a safe set defined by a continuous function. Its proof is similar to that of Theorem 2 and is omitted here.

Proposition 1. *Given p ($p \geq 2$) continuously differentiable functions $h_1, ..., h_p$ where $h_i : \mathbb{R}^n \to \mathbb{R}$ and a partition of $\mathbb{R}^n$ by p closed sets $\mathcal{S}_1, ..., \mathcal{S}_p$ such that if $\mathcal{S}_i \cap \mathcal{S}_j \neq \emptyset$, then $h_i(x) = h_j(x)$ for $x \in \mathcal{S}_i \cap \mathcal{S}_j$, define a function h as in* (7) *and the set $\mathcal{C} = \{x \in \mathbb{R}^n | h(x) \geq 0\}$. If $K(x) \neq \emptyset$ for each $x \in \mathbb{R}^n$, then any locally Lipschitz continuous controller $u : \mathbb{R}^n \to K(x)$ will render the set $\mathcal{C}$ forward invariant under the closed-loop system associated with* (4)*.*

III. Problem Formulation

In this section, we first introduce the individual models and specifications for LK and ACC, respectively. Then, we formulate a composition problem that is studied in the paper, and propose an assume-guarantee contract between LK and ACC.

A. Dynamic Models

The LK model used in this paper is the lateral-yaw model as described in [3], [4], [38], and the ACC model is the point-mass model in [16], [29]. Their dynamics are given in (9) and (10), respectively.

Equation (9) describes the lateral-yaw dynamics of the controlled vehicle. In (9), $\mathbf{x}_1 := (y, \nu, \Delta\psi, r)'$ is the state, where $y, \nu, \Delta\psi$ and r represent the lateral displacement from the center of the lane, the lateral velocity, the yaw angle deviation in road-fixed coordinates, and the yaw rate, respectively; the input $u_1 = \delta_f$ is the steering angle of the front wheels; and d is the desired yaw rate, which is viewed as a time-varying external disturbance and computed from road curvature by $d = v_f/R_0$ where R_0 is the (signed) radius of the road curvature and v_f is the vehicle's longitudinal velocity. Moreover, m is the total mass of the vehicle, and C_f, C_r, a and b are parameters of the tires and vehicle geometry that are all positive numbers.

Equation (10) describes the longitudinal dynamics of the preceding vehicle and controlled vehicle. In (10), $\mathbf{x}_2 := (v_f, v_l, D)'$ is the state, which represent the following car's speed, the lead car's speed and the distance between them, respectively; $u_2 = F_w$ is the input that represents the longitudinal force developed by the wheels; $F_r = c_0 + c_1 v_f + c_1 v_f^2$ is the aerodynamic drag, with constants c_0, c_1, c_2 that can be determined empirically; a_L is the overall acceleration/deceleration of the lead car.

Equations (9) and (10) are rewritten compactly as follows:

$$\dot{\mathbf{x}}_1 = f_1(\mathbf{x}_1, v_f) + g_1(\mathbf{x}_1)u_1 + \Delta f_1(d), \tag{11}$$

$$\dot{\mathbf{x}}_2 = f_2(\mathbf{x}_2) + g_2(\mathbf{x}_2)u_2 + \Delta f_2(\nu r, a_L), \tag{12}$$

where

$$f_1(\mathbf{x}_1, v_f) = A_1(v_f)\mathbf{x}_1,\ g_1(\mathbf{x}_1) = B_1,\ \Delta f_1(d) = E_1 d,$$
$$f_2(\mathbf{x}_2) = A_2\mathbf{x}_2,\ g_2(\mathbf{x}_2) = B_2,\ \Delta f_2(\nu r, a_L) = E_2,$$

with

$$A_1(v_f) = \begin{pmatrix} 0 & 1 & v_f & 0 \\ 0 & -\frac{C_f+C_r}{mv_f} & 0 & \frac{bC_r-aC_f}{mv_f} - v_f \\ 0 & 0 & 0 & 1 \\ 0 & \frac{bC_r-aC_f}{I_z v_f} & 0 & -\frac{a^2C_f+b^2C_r}{I_z v_f} \end{pmatrix},$$

$$B_1 = \begin{pmatrix} 0 \\ \frac{C_f}{m} \\ 0 \\ \frac{aC_f}{I_z} \end{pmatrix}, \quad E_1 = \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 1 & 0 \end{pmatrix}, B_2 = \begin{pmatrix} \frac{1}{m} \\ 0 \\ 0 \end{pmatrix},$$

$$E_2 = \begin{pmatrix} -\nu r - \frac{F_r}{m} \\ a_L \\ 0 \end{pmatrix}.$$

It is supposed that a bound is imposed on the steering angle δ_f of the controlled vehicle, that is, the set of admissible inputs for LK is

$$U_{lk} := [-\hat{\delta}_f, \hat{\delta}_f], \tag{13}$$

where $\hat{\delta}_f > 0$ is the maximum steering angle. At highway speeds, this number will be much smaller than the maximum turning angle of the vehicle, say two or three degrees, versus 35 degrees.

It is also supposed that the acceleration/deceleration of the controlled car is bounded, that is, the set of admissible inputs for ACC is

$$U_{acc} := [-a_f mg, a'_f mg], \tag{14}$$

where $a_f, a'_f > 0$. Typical bounds for driver comfort would be two or three tenths of gravitational acceleration. Note that

$-a_f g$ is normally much less than the maximal deceleration capability of the car. Additionally, it is supposed that the deceleration/acceleration of the lead car is bounded. That is, $a_L \in [-a_l g, a_l' g]$ for some $a_l, a_l' > 0$, which can be also expressed as $a_L \in \mathcal{D}_{a_L}$ where

$$\mathcal{D}_{a_L} := \{a \in \mathbb{R} | (a + a_l g)(a_l' g - a) \geq 0\}. \tag{15}$$

The lead and controlled vehicles may have different allowable deceleration capabilities, i.e., a_l and a_f need not be equal.

B. Specifications

Specifications for LK and ACC are given in this subsection. Among these specifications, the safety specifications are "hard constraints" that must be satisfied for all time, while the performance objectives are "soft constraints" that can be overridden when they are in conflict with safety.

LK Specifications. The primary safety constraint of LK is to keep the car within its lane. That is, the absolute value of the lateral displacement y is less than some given constant y_m, which is related to the width of the lane and the car. Specifically, this specification is expressed as

$$|y| \leq y_m, \tag{16}$$

where y_m is a given positive real number.

In addition to the lateral displacement, the other state variables should also be bounded. These bounds are stated as the following specifications:

$$|\nu| \leq \nu_m, \ |\Delta\psi| \leq \Delta\psi_m, \ |r| \leq r_m, \tag{17}$$

where ν_m, $\Delta\psi_m$, r_m are given positive real numbers.

A soft constraint for LK is for the vehicle's yaw rate $r(t)$ to track $d(t)$, the yaw or turning rate of the road, which is expressed as

$$\lim_{t\to\infty} r(t) - d(t) = 0. \tag{18}$$

Another optional soft constraint is for the lateral acceleration to be upper bounded by a number that respects driver comfort, e.g., $|\dot{\nu}| \leq 0.25g$.

We define the set $\mathcal{X}_{LK}$ as

$$\begin{aligned}\mathcal{X}_{LK} := \{\mathbf{x}_1 \in \mathbb{R}^4 | &|y| \leq y_m, |\nu| \leq \nu_m, \\ &|\Delta\psi| \leq \Delta\psi_m, |r| \leq r_m\}.\end{aligned} \tag{19}$$

In what follows, we assume that $|d| \leq d_{\max}$ for some given $d_{\max} > 0$, that is, $d \in \mathcal{D}_d$ where

$$\mathcal{D}_d = \{d \in \mathbb{R} | d_{\max}^2 - d^2 \geq 0\}. \tag{20}$$

ACC Specifications. The primary constraint for ACC is that the controlled vehicle *maintain a safe distance from the lead car*. There are numerous formulations of this safety concept such as Time Headway and Time to Collision. In this paper, we use the following hard constraint [39]:

$$D \geq \tau_d v_f + D_0, \tag{21}$$

where τ_d is the desired time headway and D_0 is the minimal distance between cars when they are fully stopped.

The soft constraint for ACC, which is the performance objective of the controlled car, is to achieve a desired speed v_d set by the driver. This specification can be expressed as

$$\lim_{t\to\infty} v_f(t) - v_d = 0, \tag{22}$$

for a given positive constant v_d. Clearly, if the lead car's speed is less than v_d, then (22) cannot be achieved. In our control formulation, this is automatically taken into account without the need for if-then-else statements defining various modes of operation.

C. Formulating the Composition Problem

The correctness guarantee for the composition of LK and ACC is formulated as the following problem.

Given LK model (11) *and ACC model* (12)*, find feedback controllers* $u_1 \in U_{lk}$ *and* $u_2 \in U_{acc}$ *such that for any* $d \in \mathcal{D}_d$ *and* $a_L \in \mathcal{D}_{a_L}$*, the hard constraints* (16)*,* (17) *and* (21) *are always satisfied, and the soft constraints* (18) *and* (22) *are achieved when they are not in conflict with the hard constraints.*

Although the safety specifications for LK and ACC are given separately, the dynamics of LK and ACC interact with each other because $A_1(v_f)$ depends on v_f and $\Delta f_2(\nu r, a_L)$ depends on the product of lateral velocity and yaw rate through the term νr. Furthermore, the external inputs d, a_L and the assumption that u_1, u_2 are bounded make the compositional problem harder to slove (see Figure 3).

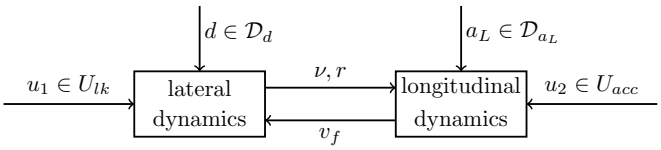

Fig. 3. Interconnection of the lateral and longitudinal dynamics.

D. A Contract Between LK and ACC

One approach for the compositional design of complex systems is the *contract-based method*, which formally defines a set of assume-guarantee protocols among subsystems in a "circular" manner [17], [18]. The basic idea is that each subsystem ensures that its behavior (i.e., set of trajectories) satisfies its own guarantees, under the assumption that all the other subsystems do the same, so that a formal guarantee on the behavior of the overall system can be established. For the composition of LK and ACC, we provide assumptions and guarantees between them in terms of the bounds of their respective coupling variables.

Let $\bar{v}, \underline{v} > 0$ be the upper and lower bound of v_f, respectively, where $\bar{v} \geq v_d$. Define a set $\mathcal{D}_{v_f}$ as

$$\mathcal{D}_{v_f} = \{v_f \in \mathbb{R} | (\bar{v} - v_f)(v_f - \underline{v}) \geq 0\}. \tag{23}$$

The contract between the LK and the ACC subsystems is given as follows: *The LK subsystem assumes that* $v_f \in \mathcal{D}_{v_f}$ *and guarantees that* $\mathbf{x}_1 \in \mathcal{X}_{LK}$ *for any* $d \in \mathcal{D}_d$*, while the ACC*

subsystem assumes that $|\nu r| \leq \nu_m r_m$ *and guarantees that* $v_f \in \mathcal{D}_{v_f}$ *for any* $a_L \in \mathcal{D}_{a_L}$.

The given contract allows us to design the controller for LK and ACC separately. As long as the individual specifications are respected in a formally correct way, the overall closed-loop system will also satisfy all the safety constraints in the compositional problem. In Section IV, we will construct CBFs for LK and ACC individually, such that the contract will be respected provided that the two subsystems constrain their behaviors in a controlled invariant set corresponding to their individual CBF. In Section V, we will synthesize a provably correct solution to the composition problem via a QP that unifies CBFs and the performance controllers.

IV. Control Barrier Functions For LK and ACC

In this section, we provide CBFs for the LK model (9) and the ACC model (10), respectively. To this end, we translate the hard constraints of Section III-B into conditions that a CBF must satisfy so that, by the results of Section II, the trajectories of the closed-loop system will satisfy the safety portion of the specification.

A. The CBF For LK

For the LK subsystem, we construct a polynomial function $h_{lk}(\mathbf{x}_1) \in \mathcal{R}_\alpha[\mathbf{x}_1]$ that has the following form:

$$h_{lk}(\mathbf{x}_1) = \kappa - \hat{h}_{lk}(\mathbf{x}_1) \tag{24}$$

where α is some given positive integer indicating the polynomial degree, $\kappa \in \mathbb{R}$ is some positive number, $\hat{h}_{lk}(\mathbf{x}_1) \in \mathcal{R}_\alpha[\mathbf{x}_1]$ is a polynomial that is nonnegative. Then the safe set for LK is defined as

$$\mathcal{C}_{lk} := \{\mathbf{x}_1 \in \mathbb{R}^4 | h_{lk}(\mathbf{x}_1) \geq 0\}.$$

Here, κ is a variable used to enlarge the volume of $\mathcal{C}_{lk}$, which will be discussed in detail later.

According to the contract in Subsection III-D, the CBF $h_{lk}(\mathbf{x}_1)$ will be designed to satisfy the following properties:

$$\text{(LK-P1) } \mathcal{C}_{lk} := \{\mathbf{x}_1 \in \mathbb{R}^4 | h_{lk}(\mathbf{x}_1) \geq 0\} \neq \emptyset, \tag{25}$$

$$\text{(LK-P2) } \mathcal{C}_{lk} \subset \mathring{\mathcal{X}}_{LK}, \tag{26}$$

$$\text{(LK-P3) } \forall \mathbf{x}_1 \in \mathcal{C}_{lk}, \forall v_f \in \mathcal{D}_{v_f}, \forall d \in \mathcal{D}_d,$$
$$\sup_{u_1 \in U_{lk}} [L_{f_1+\Delta f_1} h_{lk}(\mathbf{x}_1) + L_{g_1} h_{lk}(\mathbf{x}_1) u_1 + \gamma h_{lk}(\mathbf{x}_1)] \geq 0, \tag{27}$$

where $\gamma > 0$ is a given number representing the gain in (5).

The property (LK-P1) ensures that the safe set $\mathcal{C}_{lk}$ is non-empty, property (LK-P2) ensures that $\mathbf{x}_1 \in \mathcal{X}_{LK}$ as long as $\mathbf{x}_1 \in \mathcal{C}_{lk}$, and property (LK-P3) ensures that h_{lk} is a control barrier function for any longitudinal velocities in $\mathcal{D}_{v_f}$ and any desired yaw rates in $\mathcal{D}_d$. Note that properties (LK-P1)-(LK-P3) imply the satisfaction of the hard constraints (16) and (17).

In summary, h_{lk} with properties (LK-P1)-(LK-P3) guarantees that, for any speed $v_f \in \mathcal{D}_{v_f}$ and desired yaw rate $d \in \mathcal{D}_d$, there exists steering angle $\delta_f \in U_{lk}$ such that the trajectory of the LK system stays in the set $\mathcal{C}_{lk}$ (and thus set $\mathcal{X}_{LK}$) if starting from $\mathcal{C}_{lk}$.

Recalling that the set $\mathcal{D}$ is the region for which the CBF condition holds (cf. Definition 1), Theorem 3 provides a sufficient condition for the existence of h_{lk} with $\mathcal{D} = \mathcal{X}_{LK}$, which is the key step in translating the properties (LK-P1)-(LK-P3) into a set of sufficient conditions that can then be synthesized by sum-of-squares programs [35], [40]. In fact, we assume that $\mathcal{D} = \mathcal{X}_{LK}$ for now, which simplifies the procedure to construct h_{lk}, and we will discuss the case $\mathcal{D} = \mathcal{C}_{lk}$ later.

Theorem 3. *Given the LK model* (11)*, the variable bounds* $y_m, \nu_m, \Delta\psi_m, r_m$*, the admissible set* U_{lk} *defined in* (13)*, a positive definite polynomial* $p(\mathbf{x}_1)$ *and the sets* $\mathcal{D}_{v_f}, \mathcal{D}_d$ *defined in* (23) *and* (20)*, if there exist* $\gamma > 0$*,* $\rho > 0$*, some positive integer* α*, polynomial* $h_{lk}(\mathbf{x}_1) \in \mathcal{R}_\alpha[\mathbf{x}_1]$*, non-negative polynomials* $s_0, s_1, ..., s_4 \in \Sigma[\mathbf{x}_1]$ *and* $s_5, ..., s_{10} \in \Sigma[\mathbf{x}_1, d, v_f]$ *such that*

$$h_{lk}(\mathbf{x}_1) - (\rho - p(\mathbf{x}_1)) s_0(\mathbf{x}_1) \geq 0, \tag{28}$$
$$(y^2 - y_m^2) s_1 + h_{lk}(\mathbf{x}_1) < 0, \tag{29}$$
$$(\nu^2 - \nu_m^2) s_2 + h_{lk}(\mathbf{x}_1) < 0, \tag{30}$$
$$(\Delta\psi^2 - \Delta\psi_m^2) s_3 + h_{lk}(\mathbf{x}_1) < 0, \tag{31}$$
$$(r^2 - r_m^2) s_4 + h_{lk}(\mathbf{x}_1) < 0, \tag{32}$$
$$\begin{aligned} &\sup_{u_1 \in U_{lk}} [L_{f_1+\Delta f_1} h_{lk}(\mathbf{x}_1) + L_{g_1} h_{lk}(\mathbf{x}_1) u_1 + \gamma h_{lk}(\mathbf{x}_1)] \\ &- (y_m^2 - y^2) s_5 - (\nu_m^2 - \nu^2) s_6 - (\Delta\psi_m^2 - \Delta\psi^2) s_7 \\ &- (r_2^m - r^2) s_8 - (d_{\max}^2 - d^2) s_9 - (\bar{v} - v_f)(v_f - \underline{v}) s_{10} \geq 0, \end{aligned} \tag{33}$$

then $h_{lk}(\mathbf{x}_1)$ *satisfies properties (LK-P1)-(LK-P3) defined in* (25)-(27)*.*

Proof. Condition (28) implies (LK-P1) because $h_{lk}(\mathbf{x}_1) \geq 0$ whenever $p(\mathbf{x}_1) \leq \rho$, which means that $\{\mathbf{x}_1 | p(\mathbf{x}_1) \leq \rho\} \subseteq \mathcal{C}_{lk}$ and therefore $\mathcal{C}_{lk} \neq \emptyset$. Based on the S-procedure, conditions (29)-(32) imply (LK-P2) because $|y| < y_m$, $|\nu| < \nu_m$, $|\Delta\psi| < \Delta\psi_m$ and $|r| < r_m$ whenever $h_{lk}(\mathbf{x}_1) \geq 0$. Condition (33) implies condition (LK-P3), since (27) holds whenever $\mathbf{x}_1 \in \mathcal{X}_{LK}, v_f \in \mathcal{D}_{v_f}, d \in \mathcal{D}_d$. □

In addition to the properties (LK-P1)-(LK-P3), it is desirable for h_{lk} to maximize the volume of $\mathcal{C}_{lk}$, the portion of the safe set that the CBF renders controlled invariant. To this end, we normalize the coefficients of $\hat{h}_{lk}$ by adding the constraint $\hat{h}_{lk}((1, ..., 1)^\top) = 1$ (i.e., the sum of the coefficients is equal to 1) and maximizing κ. Note that the normalization is necessary to make the maximization of κ valid, since otherwise the coefficients of $\hat{h}_{lk}$ can be scaled accordingly with κ that results in the same $\mathcal{C}_{lk}$, which makes maximizing κ have no meaning. The normalization method has also been used in finding the maximal region of attraction of a control system using Lyapunov function and SOS [37], [41].

The terms ρs_0 in (28) and γh_{lk} in (33) involve products of the unknowns, but it can be turned into linear constraints on the unknowns through bisecting ρ and γ. Moreover, the maximization over the allowed steering angles in (33) also involves a product of the unknowns, which can be overcome by finding an explicit formula for the controller and iterating between improving the current control solution and the CBF.

In what follows, we explain in detail the iterative procedure to construct h_{lk}, which satisfies (28)-(33) and maximizes the volume of $\mathcal{C}_{lk}$, using SOS programs.

1.Initialization. The iteration process is initialized by first computing an LQR controller for a nominal value of $v_f \in \mathcal{D}_{v_f}$ and $d = 0$. More specifically, we first choose weight matrices $Q \in \mathbb{R}^{4\times4}$ penalizing the state and $R \in \mathbb{R}$ penalizing the input, where look-ahead of the road curvature can be taken into account in Q [3]. Then we solve for the LQR gain $K \in \mathbb{R}^{1\times4}$ for the system $(A(v_f), B)$ with a given and fixed $v_f \in \mathcal{D}_{v_f}$. Based on the gain K, we fix the control $u_1 = \frac{-K\mathbf{x}_1}{1+\eta(K\mathbf{x}_1)^2}$ and solve for h_{lk} using a feasibility SOS program, with η selected as $\eta = 1/(2\hat{\delta}_f)^2$. This value implies that $|\frac{-K\mathbf{x}_1}{1+\eta(K\mathbf{x}_1)^2}| \leq \hat{\delta}_f$ and therefore $u_1 \in U_{lk}$.

Remark 1. *The motivation for scaling the LQR control $-K\mathbf{x}_1$ by $1+\eta(K\mathbf{x}_1)^2$ includes: (i) an LQR controller gain K is easily computed; (ii) the value function of the LQR controller corresponds to a quadratic form, and there always exists a sufficiently small sublevel set of the quadratic form contained in $\mathcal{X}_{LK}$, for which the values of the controller over that sublevel set lie in U_{lk}; and (iii), while there is no guarantee that the controller u_1 computed in this manner will result in a feasible SOS program (i.e., the following $(\mathcal{P}_0)$) for all $v_f \in \mathcal{D}_{v_f}$ and $d \in \mathcal{D}_d$, this has not been a problem in the examples we have worked when ρ is sufficiently small. Of course, the user is free to use other control synthesis methods to initiate the iteration process. It is also a choice whether or not to feed forward the desired yaw rate.*

Given an LQR gain K, we choose values for $\gamma > 0$ and sufficiently small $\rho_0 > 0, \varepsilon > 0$ and use the following feasibility SOS program for initialization.

$(\mathcal{P}_0)$:
find $h_{lk} \in \mathcal{R}_\alpha[\mathbf{x}_1], s_0, s_1, ..., s_4 \in \Sigma[\mathbf{x}_1], s_5, ..., s_{10} \in \Sigma[\mathbf{x}_1, d, v_f]$
such that

$$h_{lk} - (\rho_0 - p)s_0 \in \Sigma[\mathbf{x}_1], \tag{34}$$
$$-h_{lk} - (y^2 - y_m^2)s_1 - \varepsilon \in \Sigma[\mathbf{x}_1], \tag{35}$$
$$-h_{lk} - (\nu^2 - \nu_m^2)s_2 - \varepsilon \in \Sigma[\mathbf{x}_1], \tag{36}$$
$$-h_{lk} - (\Delta\psi^2 - \Delta\psi_m^2)s_3 - \varepsilon \in \Sigma[\mathbf{x}_1], \tag{37}$$
$$-h_{lk} - (r^2 - r_m^2)s_4 - \varepsilon \in \Sigma[\mathbf{x}_1], \tag{38}$$
$$\begin{aligned}&\frac{\partial h_{lk}}{\partial \mathbf{x}_1}[A(v_f)\mathbf{x}_1 + Ed][1+\eta(K\mathbf{x}_1)^2]v_f + \frac{\partial h_{lk}}{\partial \mathbf{x}_1}B(-K\mathbf{x}_1)v_f\\ &+\gamma h_{lk}[1+\eta(K\mathbf{x}_1)^2]v_f - (y_m^2 - y^2)s_5 - (\nu_m^2 - \nu^2)s_6\\ &-(\Delta\psi_m^2 - \Delta\psi^2)s_7 - (r_m^2 - r^2)s_8 - (d_{\max}^2 - d^2)s_9\\ &-(\bar{v} - v_f)(v_f - \underline{v})s_{10} \in \Sigma[\mathbf{x}_1, d, v_f],\end{aligned} \tag{39}$$

where $\eta = 1/(2\hat{\delta}_f)^2$.

Note that (34) implies (LK-P1), (35)-(38) imply (LK-P2) based on the S-procedure, and (39) implies that for any $\mathbf{x}_1 \in \mathcal{X}_{LK}$, $v_f \in \mathcal{D}_{v_f}$, $d \in \mathcal{D}_d$,

$$\frac{\partial h_{lk}}{\partial \mathbf{x}_1}[A(v_f)\mathbf{x}_1 + B\frac{-K\mathbf{x}_1}{1+\eta(K\mathbf{x}_1)^2} + Ed] + \gamma h_{lk} \geq 0. \tag{40}$$

Thus, (40) means that the control $u_1 = \frac{-K\mathbf{x}_1}{1+\eta(K\mathbf{x}_1)^2} \in U_{lk}$ results in the CBF condition $\dot{h}_{lk}(\mathbf{x}_1, v_f, d, u_1) + \gamma h_{lk}(\mathbf{x}_1) \geq 0$ holding. Because $A(v_f)$ is a rational matrix with the denominator v_f in some of its entries, it needs to be multiplied with v_f so that it becomes a polynomial [40]. Note that in $(\mathcal{P}_0)$ and in what follows, the degrees of the multipliers s_i are not specified explicitly and assumed to be chosen appropriately.

If $(\mathcal{P}_0)$ is infeasible, we repeat it by modifying parameters $Q, R, \gamma, \rho_0, \varepsilon$ and increasing α; otherwise, h_{lk} is obtained as a polynomial that satisfies properties (LK-P1)-(LK-P3). Then, the following two steps will be used to find a polynomial h_{lk} that increases the volume of $\mathcal{C}_{lk}$.

2.Synthesize Controller. Given the polynomial h_{lk} from the initialization step, which is denoted as h_{lk}^{old} in the subsequent $(\mathcal{P}_1)$, we use the following maximization SOS program to find a new controller $u \in \mathcal{R}_\beta[\mathbf{x}_1, d, v_f]$ with some positive integer β, $\kappa \in \mathbb{R}$ and multipliers $s_i (0 \leq i \leq 22)$, such that $u \in U_{lk}$, κ is maximized and for any $\mathbf{x}_1 \in \mathcal{X}_{LK}, v_f \in \mathcal{D}_{v_f}, d \in \mathcal{D}_d$, the CBF condition (27) holds.

$(\mathcal{P}_1)$:
max κ
over $\kappa \in \mathbb{R}, u \in \mathcal{R}_\beta[\mathbf{x}_1, d, v_f], s_0, s_1, ..., s_4 \in \Sigma[\mathbf{x}_1]$,
$s_5, ..., s_{22} \in \Sigma[\mathbf{x}_1, d, v_f]$, such that

$$h_{lk} - h_{lk}^{old}s_0 \in \Sigma[\mathbf{x}_1], \tag{41}$$
(35) − (38) hold,
$$\begin{aligned}&\frac{\partial h_{lk}}{\partial \mathbf{x}_1}[A(v_f)\mathbf{x}_1 + Bu(\mathbf{x}_1, d, v_f) + Ed]v_f + \gamma h_{lk}v_f\\ &-(y_m^2 - y^2)s_5 - (\nu_m^2 - \nu^2)s_6 - (\Delta\psi_m^2 - \Delta\psi^2)s_7\\ &-(r_m^2 - r^2)s_8 - (\bar{v} - v_f)(v_f - \underline{v})s_9\\ &-(d_{\max}^2 - d^2)s_{10} \in \Sigma[\mathbf{x}_1, d, v_f],\end{aligned} \tag{42}$$
$$\begin{aligned}&u(\mathbf{x}_1, d, v_f) + \hat{\delta}_f - (y_m^2 - y^2)s_{11} - (\nu_m^2 - \nu^2)s_{12}\\ &-(\Delta\psi_m^2 - \Delta\psi^2)s_{13} - (r_m^2 - r^2)s_{14} - (d_{\max}^2 - d^2)s_{15}\\ &-(\bar{v} - v_f)(v_f - \underline{v})s_{16} \in \Sigma[\mathbf{x}_1, d, v_f],\end{aligned} \tag{43}$$
$$\begin{aligned}&-u(\mathbf{x}_1, d, v_f) + \hat{\delta}_f - (y_m^2 - y^2)s_{17} - (\nu_m^2 - \nu^2)s_{18}\\ &-(\Delta\psi_m^2 - \Delta\psi^2)s_{19} - (r_m^2 - r^2)s_{20} - (d_{\max}^2 - d^2)s_{21}\\ &-(\bar{v} - v_f)(v_f - \underline{v})s_{22} \in \Sigma[\mathbf{x}_1, d, v_f].\end{aligned} \tag{44}$$

Condition (42) means that with such u, the CBF condition $\dot{h}_{lk}(\mathbf{x}_1, v_f, d, u) + \gamma h_{lk}(\mathbf{x}_1) \geq 0$ holds, for any $\mathbf{x}_1 \in \mathcal{X}_{lk}, d \in \mathcal{D}_d, v_f \in \mathcal{D}_{v_f}$. Conditions (43)-(44) mean that the synthesized $u \in \mathcal{R}_\beta[\mathbf{x}_1, d, v_f]$ satisfies $|u| \leq \hat{\delta}_f$, which implies that $u \in U_{lk}$, for any $\mathbf{x}_1 \in \mathcal{C}_{lk}, d \in \mathcal{D}_d, v_f \in \mathcal{D}_{v_f}$.

When the procedure goes from $(\mathcal{P}_0)$ to $(\mathcal{P}_1)$, which will happen only once, it is not guaranteed that $(\mathcal{P}_1)$ will be feasible. In case of infeasibility, we can increase the degree of u and repeat $(\mathcal{P}_1)$. However, in all examples we have worked, $(\mathcal{P}_1)$ has been feasible, even when u is chosen to be of degree two. On the other hand, $(\mathcal{P}_1)$ will always be feasible when executed after $(\mathcal{P}_2)$, another SOS program that will be discussed shortly. Therefore, we assume that $(\mathcal{P}_1)$ is feasible at initialization and proceed.

Remark 2. *If we choose the controller in* $(\mathcal{P}_1)$ *to be a rational function in the form of* $\tilde{K}_1\mathbf{x}_1/(1+\mathbf{x}_1^\top \tilde{K}_2\mathbf{x}_1)$ *where* $\tilde{K}_1 \in \mathbb{R}^{1\times 4}$, $\tilde{K}_2 \in \mathbb{R}^{4\times 4}$ *positive definite, then* $(\mathcal{P}_1)$ *is guaranteed to be feasible because* (40) *holds and* u_1 *in* $(\mathcal{P}_0)$ *is a rational polynomial function of the same form. Higher order terms can also be included in the numerator/denominator of the rational function. The following algorithms remain true after appropriate modifications, if the rational functions template are used for the controller. This also validates the above assumption that* $(\mathcal{P}_1)$ *is always feasible.*

3.Synthesize Barrier. Given the controller $u(\mathbf{x}_1, d, v_f)$ and the CBF h_{lk} from $(\mathcal{P}_1)$, which will be denoted as h_{lk}^{old} in the subsequent $(\mathcal{P}_2)$, the following SOS program finds a new CBF h_{lk} and multipliers $s_0, s_1, ..., s_{10}$ to maximize κ.

$$
\begin{aligned}
&(\mathcal{P}_2):\\
&\max\ \kappa\\
&\text{over } \kappa\in\mathbb{R}, \hat{h}_{lk}\in\mathcal{R}_\alpha[\mathbf{x}_1], s_0, s_1, ..., s_4\in\Sigma[\mathbf{x}_1],\\
&\qquad s_5, ..., s_{10}\in\Sigma[\mathbf{x}_1, d, v_f], \text{such that}\\
&(35)-(38), (41) \text{ and } (42) \text{ hold.}
\end{aligned}
$$

Note that $(\mathcal{P}_2)$ is always feasible since u and κ in $(\mathcal{P}_1)$ constitute a feasible solution, and the resulting h_{lk} is a polynomial satisfying (28)-(33) and therefore properties (LK-P1)-(LK-P3).

With the new CBF h_{lk} constructed, we return to Step 2 to continue the iterative procedure until convergence. Because $\mathcal{X}_{LK}$ is a compact set and the constructed set $\mathcal{C}_{lk}$ in each step of $(\mathcal{P}_1)$-$(\mathcal{P}_2)$ is no smaller than in the previous step, asymptotic convergence is guaranteed. In practice, we can either terminate the algorithm when the change of κ is below some threshold, or simply set in advance the number of iterations. Furthermore, when we return to $(\mathcal{P}_1)$, it is guaranteed to be feasible since u in the last step, i.e., $(\mathcal{P}_2)$, is a feasible solution.

Algorithm 1 and Proposition 2 summarize the above results.

Algorithm 1 Synthesis of Control Barrier Functions for LK

Input: $y_m, \nu_m, \Delta\psi_m, r_m, \hat{\delta}_f, d_{\max}, \bar{v}, \underline{v}, Q, R, \gamma, \varepsilon, \rho_0, p, \alpha, \beta$
Output: $\kappa, \hat{h}_{lk}(\mathbf{x}_1), u(\mathbf{x}_1, d, v_f)$

```
Solve for the LQR gain K and solve (P_0)
while (P_0) is not feasible do
    Modify Q, R, γ, ρ_0 and solve (P_0)
end while
converged = false
while ¬ converged do
    Fix ĥ_lk, find u, s_i, κ and maximize κ by solving (P_1)
    Fix u, find ĥ_lk, s_i, κ and maximize κ by solving (P_2)
    if |κ^new − κ^old| ≤ some threshold then
        converged = true
    end if
end while
```

Proposition 2. *If* $(\mathcal{P}_0)$ *is feasible, then Algorithm 1 terminates and the polynomial* $h_{lk}(\mathbf{x}_1)$ *returned by it satisfies properties (LK-P1)-(LK-P3).*

Remark 3. *There are no efficient and reliable solvers for semi-definite programs with bilinear constraints in the decision variables, which are non-convex and known to be NP-hard in general. Iterative procedures have therefore been commonly used to bypass the bilinear constraints for SOS programs; for instance, they were used to search for control Lyapunov functions in [42] and to construct an invariant funnel along trajectories in [37]. However, in contrast to the cited results, the particular controller constructed in Algorithm 1 is not important to us since it will not be implemented directly on the system; indeed, the actual control input will be generated by solving a quadratic program that will be explained in Section V. In fact, it is the CBF* h_{lk} *that is crucial to us, because it characterizes the safe set* $\mathcal{C}_{lk}$ *that can be rendered controlled invariant using input values selected from* U_{lk}*. These observations allow us to focus on the construction of the CBFs instead of the control law (recall the discussion in Remark 2 about the flexible form of the controller).*

By fixing h_{lk} and u obtained from Algorithm 1, we can further maximize γ by solving the following SOS program:

$$
\begin{aligned}
&\max\ \gamma\\
&\text{over } \gamma\in\mathbb{R}, s_0, ..., s_4\in\Sigma[\mathbf{x}_1], s_5, ..., s_{10}\in\Sigma[\mathbf{x}_1, d, v_f]\\
&\text{such that (42) holds.}
\end{aligned}
$$

With the maximal γ, we obtain the maximal allowable input set $K_{\text{zcbf}}(x)$ (cf. Definition 6) w.r.t. the CBF h_{lk}, from which the input can render the set $\mathcal{C}_{lk}$ controlled invariant under the dynamics of the LK system.

Remark 4. *Since the CBF* h_{lk} *satisfies* $\dot{h}_{lk}+\gamma h_{lk}\geq 0$ *in* $\mathcal{X}_{LK}$*, the set* $\mathcal{C}_{lk}$ *is asymptotically stable in* $\mathcal{X}_{LK}$ *under a control law taking values from* $K_{\text{zcbf}}(x)$*. Therefore, we can take into account affine disturbances in* (9) *similar to the argument in [30], by which it can be shown that the LK system is input-to-state stable with respect to the disturbances and a larger controlled invariant set containing* $\mathcal{C}_{lk}$ *can be quantitatively given.*

The argument above shows that, if we choose $\mathcal{D}=\mathcal{X}_{LK}$, then $\mathcal{C}_{lk}$ is controlled invariant and is attractive within $\mathcal{X}_{LK}$ under control from $K_{\text{zcbf}}(x)$. On the other hand, if we choose $\mathcal{D}=\mathcal{C}_{lk}$, we will get a larger set $\mathcal{C}_{lk}$ in principle (since it is not attractive outside $\mathcal{C}_{lk}$), but constructing h_{lk} that defines such $\mathcal{C}_{lk}$ would then become more involved. Note that for this case, Theorem 3 is still true if condition (42) is changed into:

$$
\begin{aligned}
&\frac{\partial h_{lk}}{\partial \mathbf{x}_1}[A(v_f)\mathbf{x}_1+Bu(\mathbf{x}_1,d,v_f)+Ed]v_f+\gamma h_{lk}v_f-h_{lk}s_5\\
&-(\bar{v}-v_f)(v_f-\underline{v})s_6-(d_{\max}^2-d^2)s_7\in\Sigma[\mathbf{x}_1,d,v_f].
\end{aligned}
\tag{45}
$$

where $s_5, s_6, s_7\in\Sigma[\mathbf{x}_1, d, v_f]$ are multipliers to be found.

As shown in (45), $h_{lk}s_5$ is an additional bilinear term of the unknowns if SOS programs are applied to construct h_{lk} for $\mathcal{D}=\mathcal{C}_{lk}$. To bypass this difficulty, we can divide $(\mathcal{P}_1)$ into two steps as follows: (i) fix h_{lk}, search for $u\in\mathcal{R}_\beta[\mathbf{x}_1,d,v_f], s_i\in\Sigma[\mathbf{x}_1,d,v_f]$ by solving a feasibility SOS program, (ii) fix $\hat{h}_{lk}$, the control u and the multipliers s_i obtained in (i), search for κ and maximize it by solving a maximization SOS program.

Then, if Line 7 of Algorithm 1 is replaced with these two steps, the resulting CBF h_{lk} will satisfy properties (LK-P1)-(LK-P3).

Remark 5. *A bound on lateral acceleration $\dot{\nu}$, which was introduced as a soft constraints for LK in Subsection III-B, can be added as a (hard) constraint to the SOS programs. However, by doing this, the feasibility of $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ will no longer be guaranteed. Thus, this constraint is not considered, and will be discussed later in Section V.*

Remark 6. *Using the SOS optimization is not the only way to design CBFs. Gerdes et al. developed a Lagrangian model of the lateral dynamics and then augmented the corresponding Hamiltonian with an additional potential term to enforce the invariance of a set delineated by the lane boundaries, in the face of road curvature variations [3]. However, with this method, it is unclear how to address bounds on steering angle, yaw rate and lateral velocity, as we have done in* (29)-(32)*; moreover, there is no distinction between safety–staying within the lane markers—and performance—how much to override the driver or how close to remain centered in the lane. When applying LQR to the LK problem, the cost-to-go function resulting from solving the Riccati equation for a constant longitudinal speed v_f does yield a quadratic barrier function for the closed-loop lateral-yaw model, with v_f in a small neighborhood of the nominal speed, and hence is also a CBF for the open-loop lateral-law model for the same range of longitudinal speed. However, the LQR approach to developing a CBF cannot handle the bounded input/state or the varying road curvature constraints; moreover, our experience is that the associated safe set computed from a sub-level set of the cost-to-go function is unacceptably small.*

B. The CBF For ACC

Suppose that the CBF for the ACC subsystem has the following form:

$$h_{acc}(\mathbf{x}_2) := D - \tau_d v_f - D_0 - \hat{h}_{acc}(v_f, v_l) \tag{46}$$

where $\hat{h}_{acc}(v_f, v_l)$ is a polynomial to be determined. According to the contract in Section III-D, the CBF h_{acc} will be designed to satisfy the following properties:

$$\text{(ACC-P1) } \mathcal{C}_{acc} := \{\mathbf{x}_2 \in \mathbb{R}^3 | h_{acc}(\mathbf{x}_2) \geq 0\} \neq \emptyset, \tag{47}$$

$$\text{(ACC-P2) } \forall v_f \in \mathcal{D}_{v_f}, \forall v_l \geq \underline{v}, \hat{h}_{acc}(v_f, v_l) \geq 0, \tag{48}$$

$$\text{(ACC-P3) } \forall \mathbf{x}_2 \in \mathcal{X}_{ACC}, \forall a_L \in \mathcal{D}_{a_L}, \forall |\nu r| \leq \nu_m r_m, \sup_{u_2 \in U_{acc}} [L_{f_2+\Delta f_2} h_{acc}(\mathbf{x}_2) + L_{g_2} h_{acc}(\mathbf{x}_2) u_2] \geq 0. \tag{49}$$

When $h_{acc}(\mathbf{x}_2) = 0$, the minimal safe distance for the controlled car is given by $D_{\min} = \tau_d v_f + D_0 + \hat{h}_{acc}(v_f, v_l)$, which is no less than $\tau_d v_f + D_0$ since $\hat{h}_{acc}(v_f, v_l) \geq 0$. Therefore, the hard constraint (21) will be satisfied when $\mathbf{x}_2 \in \mathcal{C}_{acc}$. Note that condition (49) implies the controlled invariance of $\mathcal{C}_{acc}$. We also point out that the CBF cannot be simply chosen as $h_{acc}(\mathbf{x}_2) := D - \tau_d v_f - D_0$, since the set $\mathcal{C}_{acc}$ thus defined is not controlled invariant using an input $u_2 \in U_{acc}$.

It is clear that an overly conservative safe distance $D_{\min}$ is undesirable, as it will encourage other cars to cut into the lane. When the SOS program is used to construct $h_{acc}(\mathbf{x}_2)$, however, the resulting $D_{\min}$ was unnecessarily large. A physics-based optimization can be used to construct h_{acc} by noting that the ACC subsystem has the following monotone property: if $(D_1, v_f, v_l) \in \mathcal{C}_{acc}$ when $a_L = -a_l g$ and $u_2 = -a_f g$, then $(D_2, v_f, v_l) \in \mathcal{C}_{acc}$ for any $D_2 \geq D_1$. This property was exploited in our previous work to compute in closed form two sets of CBFs for ACC (see [43] and its supplemental material [44]). A set of three or four continuously differentiable functions are provided, with which the CBF h_{acc} is constructed from these functions by (7). In the derivation in [44], we assumed that the first equality in (10) is simplified to $\dot{v}_f = \frac{\hat{u}_2}{m}$, where $\hat{u}_2 \geq -\hat{a}_f mg$ for some $\hat{a}_f > 0$. For the ACC subsystem considered here, we have $\hat{u}_2 = u_2 + F_r - m\nu r$, which implies that $\hat{a}_f \geq a_f + (c_0 + c_1 \underline{v} + c_2 \underline{v}^2)/mg - \nu_m r_m / g$.

In summary, by using a deceleration bound that takes into account the aerodynamic drag and the bound of $|\nu r|$ in the contract, the closed-form CBFs proposed in [43] and [44] are used to construct $h_{acc}(\mathbf{x}_2)$ satisfying properties (ACC-P1)-(ACC-P3).

V. Compositional Control Synthesis Via Quadratic Program

In this section, a solution will be provided to the composition problem of LK and ACC formulated in Section III-C.

Once the CBFs are obtained (off line), the controls are generated by quadratic programs that combine the hard constraints (i.e., safety), which are expressed as CBF conditions, and the soft constraints (i.e., performance objectives), which indicate closeness to a nominal controller but are overridden when they conflict with the hard constraints.

The hard constraints are expressed as the controlled invariance of the sets $\mathcal{C}_{lk}$ for LK and $\mathcal{C}_{acc}$ for ACC, using the CBF condition. Because of the assumptions and guarantees between the two subsystems, the controlled invariant set for the compositional system is a Cartesian product of $\mathcal{C}_{lk}$ and $\mathcal{C}_{acc}$, and the behaviors of the LK and ACC subsystems can be decoupled as long as their states are confined within these two sets, respectively. Therefore, local controllers for LK and ACC can be synthesized by solving two separate QPs.

Based on the CBFs $h_{lk}(\mathbf{x}_1), h_{acc}(\mathbf{x}_2)$ constructed in Section IV, the hard constraints for LK and ACC can be expressed with some positive gains γ_1, γ_2 as follows

$$L_{f_1+\Delta f_1} h_{lk}(\mathbf{x}_1) + L_{g_1} h_{lk}(\mathbf{x}_1) u_1 + \gamma_1 h_{lk}(\mathbf{x}_1) \geq 0, \tag{50}$$

$$L_{f_2+\Delta f_2} h_{acc}(\mathbf{x}_2) + L_{g_2} h_{acc}(\mathbf{x}_2) u_2 + \gamma_2 h_{acc}(\mathbf{x}_2) \geq 0. \tag{51}$$

The soft constraint (18) for LK can be expressed as follows:

$$u_1 = \bar{K}(\mathbf{x}_1 - \mathbf{x}_1^f) + \delta_1, \tag{52}$$

where $\delta_1 > 0$ is a relaxation variable, $\mathbf{x}_1^f = [0, 0, 0, d]^\top$ is a feedforward term, and $\bar{K}$ is a feedback gain determined by solving a LQR problem such that $\mathbf{x}_1 \to \mathbf{x}_1^f$.

For ACC, we use a candidate CLF $V(\mathbf{x}_2) := (v_f - v_d)^2$ to express the soft constraint (22) with the following CLF condition:

$$L_{f_2+\Delta f_2} V(\mathbf{x}_2) + L_{g_1} V(\mathbf{x}_2) u_2 + cV(\mathbf{x}_2) \leq \delta_2, \tag{53}$$

where $\delta_2 > 0$ is the second relaxation variable, and $c > 0$ is a given constant related to the convergence rate.

Then, among the set of controls that satisfy constraints (50)-(51), the *min-norm controllers* [45] are obtained by solving the following two QPs:

$$\begin{aligned}
\mathbf{u}_1^*(x) = & \underset{\mathbf{u}_1=[u_1,\delta_1]^\top \in \mathbb{R}^2}{\operatorname{argmin}} \frac{1}{2}\mathbf{u}_1^\top H_{\mathrm{lk}} \mathbf{u}_1 + F_{\mathrm{lk}}^\top \mathbf{u}_1 && \text{(QP-LK)}\\
\text{s.t.} \quad & A_{lk}\mathbf{u}_1 \leq b_{lk},\\
& u_1 = -K(\mathbf{x}_1 - \mathbf{x}_1^f) + \delta_1,\\
\mathbf{u}_2^*(x) = & \underset{\mathbf{u}_2=[u_2,\delta_2]^\top \in \mathbb{R}^2}{\operatorname{argmin}} \frac{1}{2}\mathbf{u}_2^\top H_{\mathrm{acc}} \mathbf{u}_2 + F_{\mathrm{acc}}^\top \mathbf{u}_2 && \text{(QP-ACC)}\\
\text{s.t.} \quad & A_{acc}\mathbf{u}_2 \leq b_{acc},\\
& A_{acc}^{\mathrm{clf}}\mathbf{u}_2 \leq b_{acc}^{\mathrm{clf}} + \delta_2,
\end{aligned}$$

where

$$\begin{aligned}
& H_{\mathrm{lk}} = \begin{bmatrix} 1 & 0 \\ 0 & p_2 \end{bmatrix},\ F_{\mathrm{lk}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\\
& H_{\mathrm{acc}} = \begin{bmatrix} \frac{1}{m^2} & 0 \\ 0 & p_1 \end{bmatrix},\ F_{\mathrm{acc}} = -\begin{bmatrix} \frac{F_r}{m^2} \\ 0 \end{bmatrix},\\
& A_{lk} = [-L_{g_1}h_{lk}(\mathbf{x}_1), 0],\\
& b_{lk} = L_{f_1+\Delta f_1}h_{lk}(\mathbf{x}_1) + \gamma_1 h_{lk}(\mathbf{x}_1),\\
& A_{acc} = [-L_{g_2}h_{acc}(\mathbf{x}_2), 0],\\
& b_{acc} = L_{f_2+\Delta f_2}h_{acc}(\mathbf{x}_2) + \gamma_2 h_{acc}(\mathbf{x}_2),\\
& A_{acc}^{clf} = [L_{g_2}V(x), -1],\\
& b_{acc}^{clf} = -(L_{f_2+\Delta f_2}V(x) + cV(x)),
\end{aligned}$$

and $p_1, p_2 \gg 0$ are the penalizing weights for relaxation variables δ_1, δ_2, respectively. Here, H_{acc}, F_{acc} are chosen as such due to partial input/output linearization of (10) [43].

As (QP-LK) and (QP-ACC) are convex QPs, they can be solved efficiently by current optimization solvers. Alternatively, it was shown in [43] that u_1, u_2 obtained by (QP-LK) and (QP-ACC) can be obtained in closed-form and are locally Lipshcitz continuous, which makes them particularly easy to use in embedded implementations.

Theorem 4. *The solutions* $\mathbf{u}_1^*(x)$ *and* $\mathbf{u}_2^*(x)$ *generated by* (QP-LK) *and* (QP-ACC) *constitute a locally Lipshcitz continuous control law that ensures the hard constraints* (16)*,* (17) *and* (21) *are satisfied for all time.*

Remark 7. *It is possible to add more performance objectives as soft constraints to the QPs* (QP-LK) *and* (QP-ACC)*. For instance, the soft constraint about the lateral acceleration can be expressed as* $|\dot{\nu}| \leq \dot{\nu}_{\max} + \delta_3$ *where* δ_3 *is another relaxation variable and* $\dot{\nu}_{\max}$ *is the given lateral acceleration bound. Adding this constraint and modifying the matrices* H_{lk}, F_{lk} *to* (QP-LK) *in an obvious way, we can still obtain a solution that ensures the satisfaction of the hard constraints.*

The objective functions of (QP-LK) and (QP-ACC) can be expressed alternatively as minimizing the difference of the real control and some nominal control. The corresponding QPs are expressed as follows:

$$\begin{aligned}
u_1^*(x) = & \underset{u_1 \in \mathbb{R}}{\operatorname{argmin}} \|u_1 - u_{no}^{lk}\|_2 && \text{(QP-LK2)}\\
\text{s.t.} \quad & A_{lk}u_1 \leq b_{lk},\\
u_2^*(x) = & \underset{u_2 \in \mathbb{R}}{\operatorname{argmin}} \|u_2 - u_{no}^{acc}\|_2 && \text{(QP-ACC2)}\\
\text{s.t.} \quad & A_{acc}u_2 \leq b_{acc}.
\end{aligned}$$

The nominal control u_{no}^{lk} (resp. u_{no}^{acc}) can be either determined by the CLF condition (resp. the LQR solution) shown above or any other legacy control laws such as those have been developed by OEMs. This shows a particular advantage of the proposed control approach in that it can endow a legacy controller as a correct-by-construction solution, where the safety of the closed-loop system is guaranteed by the CBF conditions. Particularly, as QPs (QP-LK2) and (QP-ACC2) essentially involve computing a minimum distance to a convex set, their closed-form solutions can be easily obtained [46]. Hence, the onboard implementation of our controller is no more burdensome than a classic PID controller.

VI. Simulation

In this section, we apply the control laws u_1, u_2, which are obtained by solving QPs (QP-LK) and (QP-ACC), to the simultaneous operation of LK and ACC in a 16 degree of freedom model in Carsim, which is a widely used vehicle simulation package in industry. Although the controllers are designed by the widely used simplified models (9) and (10), the overall system is shown to satisfy all of the safety specifications.

The parameter values related to the vehicle dynamics are extracted from the D-Class Sedan model in CarSim, and are shown in Table I. The controlled car is allowed to employ a maximal deceleration of $0.25\ g$, maximal steering angle of $0.06\ rad$ (i.e., approximately 3.5 degrees), and have a desired pre-set speed $v_d = 22\ m/s$ and time-headway setting $\tau_d = 1.8$ seconds. The bound $d_{\max}$ related to the road curvature is given as $0.1\ rad/s$, and the bounds related to the lateral dynamics are given as $y_{\max} = 0.9\ m$, $\nu_{\max} = 1\ m/s$, $\Delta\psi_{\max} = 0.05\ rad$ and $r_{\max} = 0.3\ rad/s$, respectively.

TABLE I
Parameter values and constraints

m	1650 kg	y_m	0.9 m	p_1	1000
c_0	51 N	ν_m	1.0 m/s	p_2	1000
c_1	1.26 Ns/m	$\Delta\psi_m$	0.05 rad	p_3	100
c_2	0.4342 Ns^2/m^2	r_m	0.3 rad/s	a_f	0.25
a	1.11 m	$\underline{v}$	15 m/s	a_f'	0.25
b	1.59 m	$\bar{v}$	30 m/s	a_l	0.25
C_f	133000 N/rad	$v_{\min}$	15 m/s	τ_d	1.8
C_r	98800 N/rad	$v_{\max}$	30 m/s	γ_2	2
I_z	2315.3 $kg\ m^2\ rad/s$	v_d	22 m/s	γ_1	2
$d_{\max}$	0.1	D_0	0.1 m	c	10
$\dot{\nu}_{\max}$	0.25 m/s^2	g	9.81 m/s^2	$\hat{\delta}_f$	0.06

With the given parameters, the CBF $h_{lk}(\mathbf{x}_1)$ is obtained by solving the SOS programs using the MATLAB toolbox yalmip along with the SDP solver Mosek. For ACC, the set of *optimal barriers* $h_{acc}(\mathbf{x}_2)$ developed in [44] are used. The QPs are solved using the MATLAB command *quadprog*, but they could be just as easily solved in closed-form as mentioned in Section V. The gain matrix $\bar{K}$ in (52) is obtained by

solving an LQR problem. Assume that the controlled car uses a lateral preview of approximately 0.4 seconds, which corresponds to an "output" Cx with $C = [1, 0, 10, 0]$. Given the control weight $R = 600$ and the state weight matrix $Q = K_p C^\top C + K_d C^\top A_1^\top A_1 C$, where A_1 is given in (11) and $K_p = 5, K_d = 0.4$, the feedback gain $\bar{K}$ is determined by solving an LQR problem with such Q and R.

We simulated the controller in Carsim on a curved road, with initial condition $(v_f(0), v_l(0), D(0)) = (18, 17, 65)$ and $(y(0), \nu(0), \Delta\psi(0), r(0)) = (0, 0, 0, 0)$. Each subfigure of Figure 4 is interpreted as follows:

- Subfigure (a) shows the speed profile of the lead car v_f (in blue) and the controlled car v_l (in black), where the desired speed v_d is depicted in dotted green line. During $t = 0s$ and $t = 7s$, the controlled car accelerates and achieves the desired speed v_d; during $t = 7s$ and $t = 12s$, the controlled car slows down to maintain a safe distance with the lead car; during $t = 12s$ and $t = 24s$, the controlled car achieves the desired speed again; after $t = 24s$, the controlled car slows down to keep a safe distance with the lead car and eventually maintains the same speed as the lead car. The "apparent changes in modes" are all achieved by the QP guaranteeing the hard (safety) constraint and meeting the soft (performance) constraint on speed tracking as closely as possible. There are no if-then-else or case statements involved.
- Subfigure (b) shows the state evolution of the lateral dynamics $\mathbf{x}_2$; it can be seen that $y, \nu, \Delta\psi, r$ are within their respective bounds during the simulation. The abrupt changes happening at $t = 20s, 25s, 45s$ are due to step changes in the road curvature.
- Subfigure (c) shows the wheel force u_1 divided by mg (in blue) and the bounds $-a_f, a_f'$ (in dotted red line); it can be seen that the wheel force constraint is satisfied as $u_1 \in U_{acc}$.
- Subfigure (d) shows the steering angle u_2 (in blue) and the bound $\hat{\delta}_f, -\hat{\delta}_f$ (in dotted red line); it can be seen that the steering angle constraint is satisfied as $u_2 \in U_{lk}$.
- Subfigure (e) shows the values of r (in black) and d (in blue); it can be seen that r tracks d pretty well, indicating satisfaction of the soft constraint (17).
- Subfigure (f) shows the values of the actual time headway τ (in blue) and the desired time headway τ_d (in dotted red line); it can be seen that $\tau \geq \tau_d$ as desired.
- Subfigure (g) shows the values of h_{acc} (in blue) and the zero-value line (in dotted red); it can be seen that h_{acc} is always positive as desired, which implies satisfaction of the ACC constraint (21). Note that $h_{acc} > 0$ during $t \in [12s, 24s]$ because the ACC-controlled car has achieved its desired speed and has no intension to reduce the relative distance.
- Subfigure (h) shows the values of h_{lk} (in blue) and the zero-value line (in dotted red); it can be seen that h_{lk} is always positive as desired, which implies satisfaction of the LK constraints (16) and (17).

VII. Conclusions

In this paper, we developed a control design approach with correctness guarantees for the simultaneous operation of lane keeping and adaptive cruise control, where the longitudinal force and steering angle are generated by solving quadratic programs. The safety constraints are hard constraints that are enforced by confining the states of the vehicle within determined controlled-invariant sets, which are expressed as CBF conditions. The performance objectives are soft constraints that can be overridden when they are in conflict with safety. The proposed QP-based framework can integrate legacy controllers as the performance controller and endow them with correct-by-construction solutions that guarantee safety. Additionally, the QP solution is known in closed form, and thus the proposed algorithm can be implemented without online optimization, if desired. The effectiveness of the proposed controller is shown by simulations in Carsim.

The SOS algorithm used to construct CBFs for the lane keeping is quite general and can be applied to other safety control problems as well. The assume-guarantee formalism is well adapted to modularity of the driver assistance modules studied in this paper because any control laws respecting the contracts given for lane keeping and adaptive cruise control, respectively, will guarantee safety of the closed-loop system when the two modules are activated simultaneously. This means in particular that the individual modules do not have to be provided by the same supplier as long as an OEM provides the correct contracts.

A preliminary test of the LK and ACC algorithms has been conducted on the Khepera robot and the Robotarium testbed [47]. Our future plans include testing these algorithms on a full-sized vehicle. Additional challenges include validating the models used for control design and considering sensor errors when constructing CBFs.

References

[1] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4649–4672, 2010.

[2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[3] E. J. Rossetter and C. J. Gerdes, "Lyapunov based performance guarantees for the potential field lane-keeping assistance system," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 3, pp. 510–522, 2006.

[4] K. L. Talvala, K. Kritayakirana, and J. C. Gerdes, "Pushing the limits: From lanekeeping to autonomous racing," *Annual Reviews in Control*, vol. 35, no. 1, pp. 137–148, 2011.

[5] J. Huang and H.-S. Tan, "Development and validation of an automated steering control system for bus revenue service," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 227–237, 2016.

[6] P. A. Ioannou and C.-C. C. Chien, "Autonomous intelligent cruise control," *Vehicular Technology, IEEE Transactions on*, vol. 42, no. 4, pp. 657–672, 1993.

[7] "National Highway Traffic Safety Administration (NHTSA)," 2014. [Online]. Available: http://www.nhtsa.gov

[8] M. Asplund, A. Manzoor, M. Bouroche, S. Clarke, and V. Cahill, "A formal approach to autonomous vehicle coordination," in *FM 2012: Formal Methods*. Springer, 2012, pp. 52–67.

[9] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Formal methods for semi-autonomous driving," in *Proceedings of the 52nd Annual Design Automation Conference*. ACM, 2015, pp. 148:1–148:5.

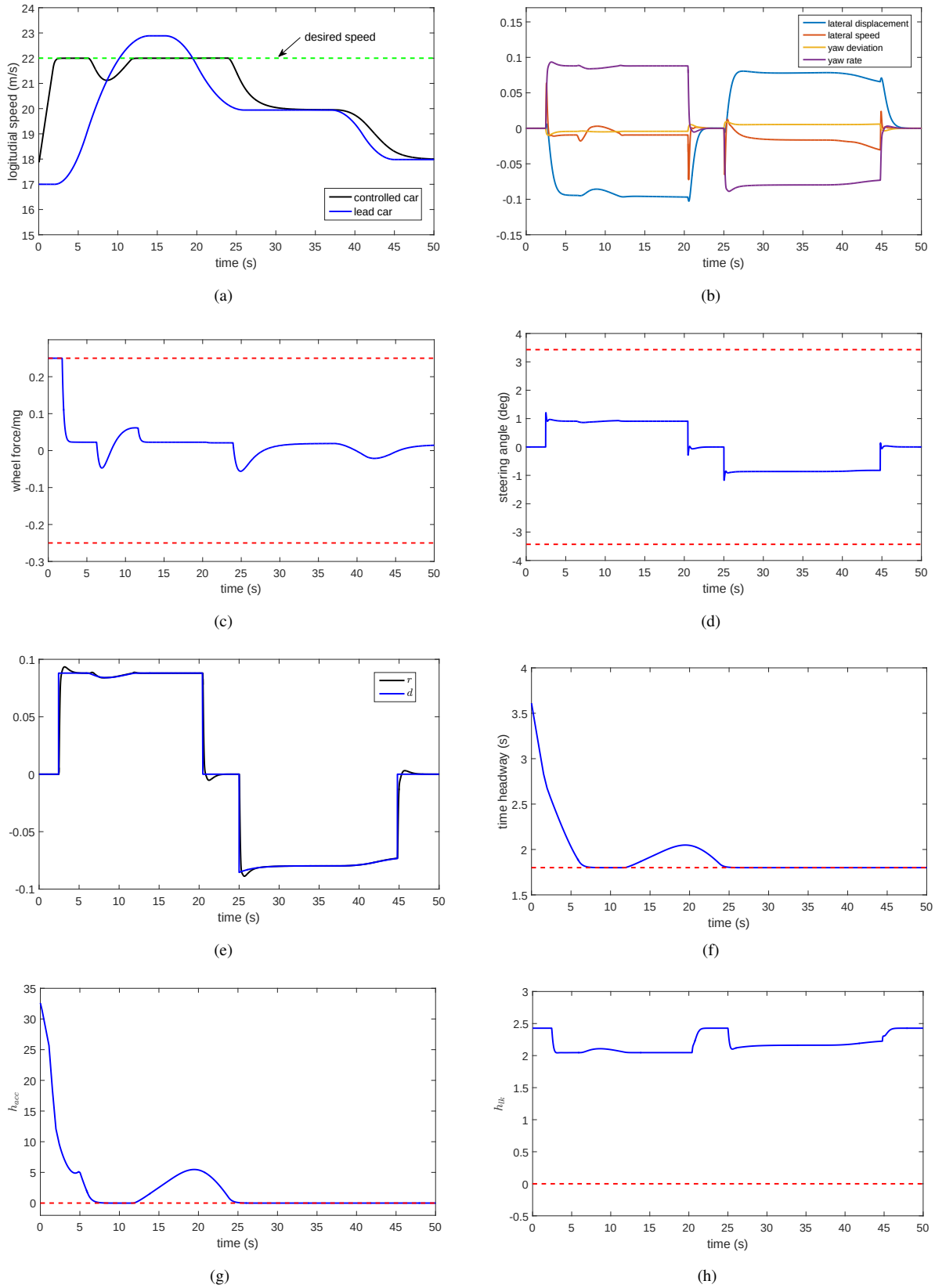

Fig. 4. (a) Speed of the controlled car v_f (in black), speed of the lead car v_l (in blue), and the desired speed v_d (in dotted green). (b) State evolution of the lateral dynamics $y, \nu, \Delta\psi, r$. (c) The wheel force u_1 divided by mg (in blue) and its bound ± 0.25 (in dotted red). (d) The steering angle u_2 and its bound ± 3.4 deg (in dotted red). (e) Values of r (in black) and d (in blue). (f) Values of the actual time headway (in blue) and the desired time headway (in dotted red) (g) Values of the CBF h_{acc} (in blue), where non-negativeness implies satisfaction of the ACC constraint (21). (h) Values of the CBF h_{lk} (in blue), where non-negativeness implies satisfaction of the LK constraints (16) and (17).

[10] M. Forghani, J. M. McNew, D. Hoehener, and D. Del Vecchio, "Design of driver-assist systems under probabilistic safety specifications near stop signs," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 43–53, 2016.

[11] S. M. Loos, A. Platzer, and L. Nistor, "Adaptive cruise control: Hybrid, distributed, and now formally verified," in *FM 2011: Formal Methods*. Springer, 2011, pp. 42–56.

[12] O. Stursberg, A. Fehnker, Z. Han, and B. H. Krogh, "Verification of a cruise control system using counterexample-guided search," *Control Eng. Pract.*, vol. 12, no. 10, pp. 1269–1278, 2004.

[13] J. Guldner, H.-S. Tan, and S. Patwardhan, "Analysis of automatic steering control for highway vehicles with look-down lateral reference systems," *Vehicle System Dynamics*, vol. 26, no. 4, pp. 243–269, 1996.

[14] Y. S. Son, W. Kim, S.-H. Lee, and C. C. Chung, "Robust multirate control scheme with predictive virtual lanes for lane-keeping system of autonomous highway driving," *Vehicular Technology, IEEE Transactions on*, vol. 64, no. 8, pp. 3378–3391, 2015.

[15] K. L. R. Talvala and C. J. Gerdes, "Lanekeeping at the limits of handling: Stability via Lyapunov functions and a comparison with stability control," in *ASME Dynamic Systems and Control Conference*, 2008, pp. 361–368.

[16] P. Nilsson, O. Hussien, A. Balkan, Y. Chen, A. Ames, J. Grizzle, N. Ozay, H. Peng, and P. Tabuada, "Correct-by-construction adaptive cruise control: Two approaches," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1294–1307, 2016.

[17] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming dr. frankenstein: Contract-based design for cyber-physical systems," *European journal of control*, vol. 18, no. 3, pp. 217–238, 2012.

[18] L. Benvenuti, A. Ferrari, E. Mazzi, and A. S. Vincentelli, "Contract-based design for computation and verification of a closed-loop hybrid system," in *HSCC*. Springer, 2008, pp. 58–71.

[19] P. Nuzzo, H. Xu, N. Ozay, J. B. Finn, A. L. Sangiovanni-Vincentelli, R. M. Murray, A. Donzé, and S. A. Seshia, "A contract-based methodology for aircraft electric power system design," *Access, IEEE*, vol. 2, pp. 1–25, 2014.

[20] E. S. Kim, M. Arcak, and S. A. Seshia, "Compositional controller synthesis for vehicular traffic networks," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 6165–6171.

[21] S. Dai and X. Koutsoukos, "Safety analysis of automotive control systems using multi-modal port-hamiltonian systems," in *19th ACM International Conference on Hybrid Systems: Computation and Control*, 2016.

[22] S. Smith, P. Nilsson, and N. Ozay, "Interdependence quantification for compositional control synthesis: An application in vehicle safety systems," in *IEEE CDC*, 2016 (to appear).

[23] J. Mareczek, M. Buss, and M. W. Spong, "Invariance control for a class of cascade nonlinear systems," *Automatic Control, IEEE Transactions on*, vol. 47, no. 4, pp. 636–640, 2002.

[24] J. Wolff and M. Buss, "Invariance control design for constrained nonlinear systems," in *Proceedings of the 16th IFAC World Congress*. Elsevier, 2005, pp. 37–42.

[25] I. Kolmanovsky, E. Garone, and S. Di Cairano, "Reference and command governors: A tutorial on their theory and automotive applications," in *American Control Conference*. IEEE, 2014, pp. 226–241.

[26] D. Hoehener, G. Huang, and D. D. Vecchio, "Lane departure assist: A formal approach," in *preprint*.

[27] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Systems: Computation and Control*, 2004, pp. 477–492.

[28] S. Prajna, A. Jadbabaie, and G. J. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *Automatic Control, IEEE Transactions on*, vol. 52, no. 8, pp. 1415–1428, 2007.

[29] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6271–6278.

[30] X. Xu, P. Tabuada, A. D. Ames, and J. W. Grizzle, "Robustness of control barrier functions for safety critical control," in *IFAC Conference on Analysis and Design of Hybrid Systems*, 2015, pp. 54–61.

[31] A. Mehra, W.-L. Ma, F. Berg, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Adaptive cruise control: Experimental validation of advanced controllers on scale-model cars," in *American Control Conference*, 2015.

[32] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[33] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic walking," in *American Control Conference*. IEEE, 2015, pp. 4542–4548.

[34] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *American Control Conference*, 2016.

[35] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, Citeseer, 2000.

[36] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[37] A. Majumdar, A. A. Ahmadi, and R. Tedrake, "Control design along trajectories with sums of squares programming," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4054–4061.

[38] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[39] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 427 – 433, 2003.

[40] J. Anderson and A. Papachristodoulou, "Robust nonlinear stability and performance analysis of an F/A-18 aircraft model using sum of squares programming," *International Journal of Robust and Nonlinear Control*, vol. 23, no. 10, pp. 1099–1114, 2013.

[41] W. Tan, "Nonlinear control analysis and synthesis using sum of squares programming," Ph.D. dissertation, University of California, Berkeley, 2006.

[42] W. Tan and A. Packard, "Searching for control lyapunov functions using sums of squares programming," in *Allerton conference on communication, control and computing*, 2004, pp. 210–219.

[43] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, 2017, doi:10.1109/TAC.2016.2638961.

[44] "Supplemental material," http://web.eecs.umich.edu/~grizzle/CBF.

[45] R. A. Freeman and P. V. Kokotovic, "Inverse optimality in robust stabilization," *SIAM Journal on Control and Optimization*, vol. 34, no. 4, pp. 1365–1391, 1996.

[46] D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons, 1969.

[47] X. Xu, T. Waters, D. Pickem, P. Glotfelter, M. Egerstedt, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds." 2017, submitted.