Safe and efficient collision avoidance control for autonomous vehicles

Qiang Wang[1,2], Dachuan Li[2,3], and Joseph Sifakis*[2,4]

[1] SUSTech Academy for Advanced Interdisciplinary Studies, Shenzhen, China
[2] School of Computer Science and Engineering, SUSTech , Shenzhen, China
[3] SUSTech Intelligent transportation Center, Shenzhen, China
[4] Verimag, Université Grenoble Alpes, France

Abstract. We study a novel principle for safe and efficient collision avoidance that adopts a mathematically elegant and general framework making as much as possible abstraction of the controlled vehicle's dynamics and of its environment. Vehicle dynamics is characterized by pre-computed functions for accelerating and braking to a given speed. Environment is modeled by a function of time giving the free distance ahead of the controlled vehicle under the assumption that the obstacles are either fixed or are moving in the same direction. The main result is a control policy enforcing the vehicle's speed so as to avoid collision and efficiently use the free distance ahead, provided some initial safety condition holds.

The studied principle is applied to the design of two discrete controllers, one synchronous and another asynchronous. We show that both controllers are safe by construction. Furthermore, we show that their efficiency strictly increases for decreasing granularity of discretization. We present implementations of the two controllers, their experimental evaluation in the Carla autonomous driving simulator and investigate various performance issues.

Keywords: Safe and efficient collision avoidance, Autonomous vehicles, Model based design

1 Introduction

As a fundamental requirement for autonomous vehicle control, the problem of collision avoidance has been widely investigated using a big variety of approaches and frameworks. These involve control-based techniques, game theory, formal methods including reachability analysis and logic-based controller synthesis or the design of specific protocols. Furthermore, the assumptions underlying the adopted frameworks vary regarding the level of modeling of the dynamics of the controlled vehicle, the number of vehicles and the type of their trajectories or the nature of the controller stimuli.

In this problem, the key issue is the development of control algorithms of tractable complexity guaranteeing collision avoidance and making efficient use of the available space. It should be emphasized that most of the existing results fail to satisfy at least one of these requirements. Most results focus on performance optimization and only partially satisfy safety requirements. Some results involve decision processes requiring

computationally heavy analysis and others propose theoretically correct solutions that are not robust when discretized. Finally, some results put emphasis exclusively on safety under various scenarios and neglect performance which is not acceptable for cars; lack of performance can become a safety issue as for instance in an overtaking maneuver.

We study a novel principle for safe and efficient collision avoidance. We adopt a mathematically simple and general framework making abstraction of the controlled vehicle's specific dynamics and of its environment, and using only three functions: (1) the free distance function $F(t)$ which determines for the vehicle the estimated free distance from the closest obstacle ahead at time t; (2) the accelerating function $A(V, v)$ which gives the distance travelled by the vehicle when accelerating from initial speed V to speed v; (3) the braking function $B(V, v)$ which gives the distance travelled by the vehicle when braking from V to speed v ($v < V$).

The principle consists in the application of a simple induction rule. If at some time t the speed of the vehicle with respect to the distance $F(t)$ is safe, i.e. $B(V, 0) \leq F(t)$, then the speed will be controlled to remain safe under the assumption that $F(t)$ does not change faster than the vehicle can brake. This assumption always holds when the obstacles ahead are fixed or move in the same direction as the controlled vehicle. Furthermore, if safety can be guaranteed for speed V and $B(V, 0) \leq F(t)$ then in order to efficiently use the available space $F(t) - B(V, 0)$ we apply an A/B (Accelerating/Braking) policy: we accelerate to a certain speed $v > V$, such that after acceleration it is still possible to safely brake from v. So efficiency boils down to computing the maximum target speed v such that $0 \leq F - (A(V, v) + B(v, 0))$. This computation may be costly depending on the properties of the accelerating and braking distance functions. The control principle consists in the dynamic application of A/B policies for a set of possible speed levels between speed 0 and the limit speed of the vehicle. For each speed level, it uses precomputed conditions for safely switching to adjacent speed levels depending on the free distance ahead.

We provide two different controllers for safe and efficient collision avoidance. The first controller is synchronous driven by periodically sampled values of the free distance F. The second controller is asynchronous receiving sporadically available values of F. We prove that both controllers are safe and efficient, where efficiency means that based on the most recent value of F, getting closer to the obstacle ahead may jeopardize safety. We also present their implementations and experimental evaluations in Carla autonomous driving simulator and investigate various performance issues.

Our approach is characterized by the following:

1. It makes abstraction of the vehicle dynamics through the use of accelerating and braking functions that provide all the information needed for safe and efficient control. These functions are a kind of "contact" between the controller and the controlled vehicle. Their use frees us from the obligation to model vehicle dynamics. Furthermore, it leaves completely open the way features related to comfort such as the jerk profile are implemented.
2. Although we consider the problem in one dimension and the environment is modeled by a free distance function $F(t)$, the result can be easily extended to two dimensions. In that case $F(t)$ and $B(v, V)$ become areas and the safety test consists in checking their inclusion.

3. The control principle is robust and easy to adapt to varying uncertainty in the measurement of F or in the estimation of the functions A and B.
4. The proposed implementations do not have any specific hardware requirements and require very limited computing resources as they combine pre-computed control policies.
5. Finally, the adopted control principle is simple and inductive: if at some step the distance is safe then a speed increase by some quantity will not jeopardize safety. This induction hypothesis is used to prove correctness.

The paper is organized as follows. Section 2 reviews related work. Section 3 presents the framework and the principle of safe and efficient collision avoidance control. Section 4 presents the design of the collision avoidance controllers. Section 5 presents the implementations and performance evaluations using the Carla simulator. Section 6 concludes about the relevance of the results and outlines directions for future work.

2 Related work

Collision avoidance has been extensively studied in the context of adaptive cruise control. Most work addresses the problem by applying optimization techniques [24, 17, 23]. For instance, [24] models the ego vehicle and the obstacles around as convex sets, and generates collision-free trajectories by solving a set of smooth non-convex constraints. In [17], safe trajectories are calculated based on a non-linear model predictive control approach for both lateral and longitudinal movements. The work in [23] applies the concept of artificial potential field and identifies five stages in the process of obstacle avoidance. In [1], a hierarchical framework consisting of a nominal controller and an emergency controller has been studied. The former is based on model predictive control (MPC) strategy and operates under normal conditions to achieve passenger comfort without considering safety guarantee, while the latter takes over if the headway approaches clearance distance constraints and ensures provable safety. However, the scheme considers multiple leading vehicles and designs a controller for each of them, thus incurring increased computational cost. Finally, several works deal with collision avoidance methods relying on rich environment information (e.g., position, speed, width of the surrounding vehicles) from V2I or V2V communication [14, 11]. Despite the promising results achieved by such approaches, optimization-based and potential field-based collision avoidance strategies do not allow safety guarantees, which are essential for autonomous vehicles. Additionally, they may lead to high computation cost in real-world implementations.

To ensure guaranteed safety and achieve correctness-by-construction, Mobileye [20] advocated the application of model-based techniques. The proposed concept of Responsibility Sensitive Safety (RSS) relies on the computation of the safe distance between vehicles. It is argued that if a vehicle maintains the required safe distance from other vehicles, it will never be responsible for an accident even if it might still become involved in an accident. Different estimates of safety distance are proposed under the assumption of constant response time for acceleration and deceleration. Furthermore, in order to avoid the unnecessarily large gap between vehicles caused by situation-unaware strategies in [20], it is shown how to improve the safe distance conditions by

taking into account the state of the ego-vehicle (pause, acceleration, deceleration) [10]. Nonetheless, this work focuses only on conditions guaranteeing safety and does not address control issues. Similarly, nVIDIA proposes a formal safety model, namely the Safety Force Field (SFF) [16], which brings in the concept of Safety Potential to evaluate the safety of traffic actors. An SFF function is defined to derive safety procedures that move an actor down the gradient of safety potential, resulting in actors repelling from each other when safety procedures are about to overlap. As RSS, SFF exclusively focuses on safety, and does not address efficiency issues. As a matter of fact, a study [25] reveals that simply implementing RSS requirements leads to undesirable clearance distance and thus to decrease of traffic efficiency.

Model-based design for autonomous vehicles have been an active research area since 30 years. In the California PATH (Partners for Advanced Transportation and Highways) program, the concept of platoon has been proposed to mitigate the highway congestion. A platoon is a group of closely spaced vehicles under automatic control. In [22], the design of platoon controllers has been investigated and a multi-layer automated highway system architecture has been proposed in order to achieve a fully automated platoon control. In [13], the analysis of [22] is refined using hybrid controllers and sufficient safety conditions are provided. Finally, [7] presents the safety and performance analysis of a hybrid system modeling an autonomous vehicle.

There are several works involving application of formal methods to autonomous vehicle control. In [21][19], safety of the adaptive cruise control is verified by predicting and checking reachable states of ego and other vehicles, which is however computationally extensive. In [2], barrier certificates provide safety guarantee by defining a 'barrier' that prevents transitions from safe states to unsafe ones. [3] studies the distributed coordination of autonomous vehicles in order to avoid collisions in intersections. The coordination protocol is modeled in a constraint specification language and the automated constraint solver (i.e. Z3) is used to verify safety. [9] presents the design and formal verification of a supervisor switching between nominal planners and a safe stop routine if nominal operational conditions are violated. In [8], a supervisor for an advanced driver assistance system is automatically synthesized from the specifications modeled by finite state machines. The correctness of the switching logic is also formally verified. In [15], a controller is synthesized from linear temporal logic specifications for adaptive cruise control. In [6] an approach is presented for proving collision freedom of multi-lane traffic with lane-change maneuver. The multi-lane motorway traffic is modeled as an abstract transition system and the collision freedom property is specified in spatial logic. Then the safety verification problem boils down to checking that the abstract transition system satisfies spatial logic formulas. In [5], linear temporal logic is used to formalize traffic rules for both overtaking and merging maneuvers. Furthermore, these rules are verified on the automata modeling the behavior of an autonomous vehicle. A motion planner modeled as a maneuver automaton is presented in [18]. For each state of this model a particular motion control primitive is applied. The desired plan is specified by a formula of linear temporal logic, and logical correctness is reduced to to checking satisfiability of the formula. Finally, several papers discuss the application of formal verification to the decision and control software of autonomous vehicles e.g., [12, 26].

3 Safe and efficient collision avoidance control

The aim is to control the movement of a vehicle travelling in a one-way lane, so as to 1) avoid collision with other objects that may be fixed or moving in the same direction (i.e., safety); and 2) use the available free distance ahead in the best possible manner to minimize travelling time (i.e., efficiency).

Our work relies on a mathematically abstract framework characterized by three functions. We denote by v the speed variable of the vehicle and by V its initial speed.

- The function $F(t)$ gives the free distance at time t between the controlled vehicle and the closest obstacle ahead, which either moves in the same direction or is stopped.
- The braking function $B(V, v)$ is a partial function defined in the interval $0 \leq v \leq V$. It gives the distance travelled by the controlled vehicle when braking from the initial speed V to a target speed v. In Fig.1 it is graphically illustrated by the green curves. When the target speed $v = 0$ (i.e, the vehicle brakes to stop), this function is abbreviated as $B(V)$ for simplicity.
- The accelerating function $A(V, v)$ is a partial function defined in the interval $V \leq v \leq V_L$, where V_L is a given limit speed for each vehicle. It gives the distance travelled by the vehicle when accelerating from an initial speed V to a target speed v. In Fig.1 it is graphically illustrated by the black curves.

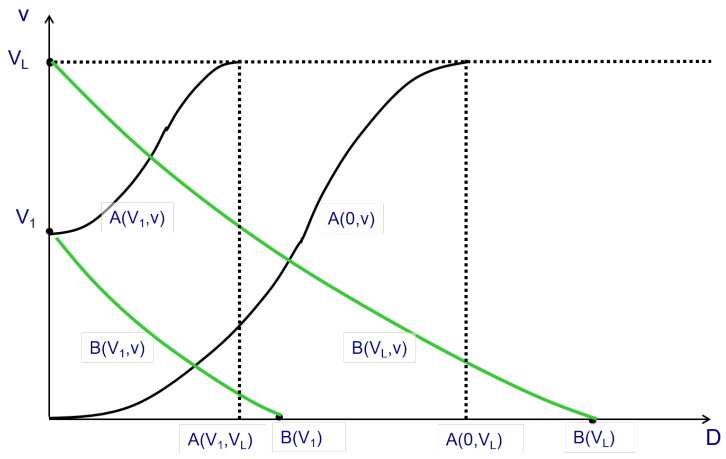

Fig. 1. Braking and acceleration distance functions (D is the distance travelled and v is the speed)

We make no specific assumptions about the implementation of accelerating and braking functions, e.g. whether acceleration and deceleration are constant or variable. Nonetheless, we require that the following properties hold:

- $B(V, V) = 0$ and $A(V, V) = 0$.

– Additivity property:

$$B(V, v_1) + B(V_1, v_2) = B(V, v_2), \ \ where \ \ v_1 = V_1$$

$$A(V, v_1) + A(V_1, v_2) = A(V, v_2), \ \ where \ \ v_1 = V_1$$

– Strict monotonicity:

$$B(V, v_1) < B(V, v_2), \ \ when \ \ v_1 < v_2$$

$$A(V, v_1) < A(V, v_2), \ \ when \ \ v_1 < v_2$$

The additivity property implies that for $0 \leq j < i \leq n$,

$$B(v_i, v_j) = \sum_{k=0}^{i-j-1} B(v_{i-k}, v_{i-k-1})$$

$$A(v_j, v_i) = \sum_{k=0}^{i-j-1} A(v_{j+k}, v_{j+k+1})$$

This says that the distance needed to brake or accelerate to a given speed is the same no matter how braking and acceleration commands have been applied.

As an example, when acceleration and deceleration rates are constant, respectively a and b, these functions are given by the following formulas:

$$B(V, v) = V * (V - v)/b - 1/2 * (V - v)^2/b$$

$$A(V, v) = V * (v - V)/a + 1/2 * (v - V)^2/a$$

We progressively study the safe and efficient collision avoidance problem for a vehicle moving in a one-way lane. We first study the problem for a stationary obstacle ahead. Then we study algorithms that solve the problem for dynamically changing free distance. We assume that the movement is controlled using commands for accelerating and braking from a speed V to some target speed v whose effect is modeled by the functions $A(V, v)$ and $B(V, v)$, respectively.

3.1 Control for safety

The basic idea for avoiding collision is to moderate the speed of the vehicle and anticipate the changes of the free space ahead so as to have enough distance and time to adjust and brake. If the vehicle moves with speed V at time t, then for safety the free space ahead $F(t)$ should be longer than the braking distance $B(V)$, which is the minimal safe braking distance for speed V. The Theorem below formalizes this idea.

Theorem 1. *If at time t the speed V_t of the vehicle is safe with respect to $F(t)$, i.e., $B(V_t) \leq F(t)$ and for any time $t + \triangle t$ it is possible to set the speed to a value $V_{t+\triangle t}$ such that the condition $F(t) - F(t + \triangle t) \leq B(V_t) - B(V_{t+\triangle t})$ holds, then the vehicle is always safe.*

Proof. The condition $F(t)-F(t+\triangle t) \leq B(V_t)-B(V_{t+\triangle t})$ relates changes of $F(t)$ to the changes of speed V. It simply says that the free space ahead does not change faster than the distance that the vehicle travels in some interval $\triangle t$. It can be deduced from the safety assumption $0 \leq F(t) - B(V_t)$ and from the condition that $0 \leq F(t) - B(V_t) \leq F(t + \triangle t) - B(V_{t+\triangle t})$.

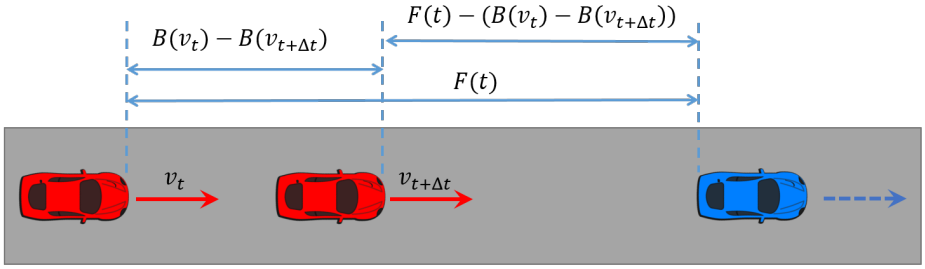

Fig. 2. Illustration of the control for safety

Notice as an application of the above theorem, that if the vehicle brakes from speed V_t and the obstacles ahead do not move in the opposite direction, then the condition $F(t)-F(t+\triangle t) \leq B(V_t)-B(V_{t+\triangle t})$ trivially holds. In fact, when the vehicle brakes from V_t for time $\triangle t$, it will reach the speed $V_{t+\triangle t} < V_t$ and it will have traveled the distance $B(V_t, V_{t+\triangle t}) = B(V_t) - B(V_{t+\triangle t})$, by application of the additivity property. Then we have that $F(t) - (B(V_t) - B(V_{t+\triangle t}))$ is the distance ahead at time $t + \triangle t$ for the controlled vehicle, as shown in Fig.2. By the assumption that the obstacles are moving forward or stopped, we have that $F(t) - (B(V_t) - B(V_{t+\triangle t})) \leq F(t + \triangle t)$. Thus Theorem1 can trivially be applied if obstacles ahead do not move in the opposite direction.

This theorem suggests a simple and safe control policy that ensures collision freedom. For any time t, the vehicle only needs to keep track of the free distance ahead $F(t)$ and check in real-time whether $F(t)$ is greater than the minimal safe braking distance $B(V_t)$ for the current speed V_t. It starts braking as soon as $F(t)$ reaches the minimal safe braking distance. In this way, it is guaranteed that if the obstacles ahead do not move in the opposite direction, no collision would happen.

3.2 Achieving efficiency for fixed obstacles

The above result provides a basis for ensuring collision freedom. Nonetheless, it leaves open the question of how the vehicle can efficiently use the available distance ahead by minimizing the travelling time. What would be an efficient driving policy when the free headway distance is greater than the minimal safe braking distance? We consider that a policy defines the speed function $v(t)$ in response to a free distance $F(t)$. An Accelerating/Braking policy (A/B policy) is a policy of accelerating first to some speed and then braking. Similarly, an Braking/Accelerating policy (B/A policy) is the policy

of braking first to some speed and then accelerating. A Constant speed/Braking policy (C/B policy) is the policy of moving at constant speed and then braking. A policy is safe if the relative distance between the controlled vehicle and the obstacle ahead is positive. It is efficient if increasing the speed value $v(t)$ enforced by the policy at any point would compromise safety.

The problem is to minimize the travelling time for a given distance, which implies to maximize the average speed. Consider the scenario where the speed of the vehicle is V and there is a stationary obstacle ahead at distance F, which is greater than the braking distance $B(V)$. The application of an A/B policy consists in computing an appropriate target speed $v, V < v \leq V_L$, accelerate the vehicle to v and then brake to full stop. To ensure collision freedom, the total travelled distance $A(V, v) + B(v)$ must be such that $A(V, v) + B(v) \leq F$. The maximal target speed is given by the following condition.

$$v_M = \max\{v \mid F \geq A(V, v) + B(v)\}$$

Such a speed exists as both acceleration and braking functions are monotonically increasing with respect to the target speed v. Notice that either $v_M \leq V_L$ and $F = A(V, v_M) + B(v_M)$ or $v_M = V_L$ and $F > A(V, v_M) + B(v_M)$.

As an example, for motion at constant acceleration and deceleration (a and b, respectively), we have $A(V, v) = v*(v-V)/a+(v-V)^2/2*a$ and $B(v) = v^2/2*b$. Then the safety condition becomes $F \geq v * (v - V)/a + (v - V)^2/2 * a + v^2/2 * b$, from which we deduce $v \leq \sqrt{(2 * a * b * F + b * V^2)/(a + b)}$. Thus the maximal target speed $v_M = \sqrt{(2 * a * b * F + b * V^2)/(a + b)}$. As we require that $v \geq V$, we have $F \geq V^2/2 * b = B(V)$ and thus the maximal target speed always exists. Let v_F denote the speed reached by accelerating along distance $F(t)$, i.e., $v_F^2 - V^2 = 2 * F(t) * a$, then the formula can be simplified as $v_M = v_F * \sqrt{b/(a + b)}$.

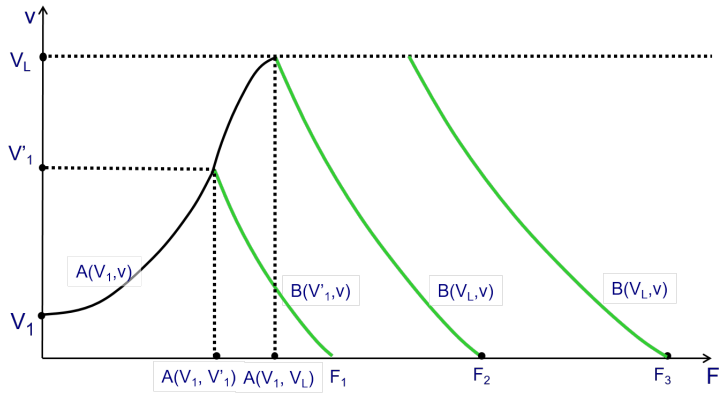

Fig. 3. The A/B control policy for different values of the free distance F ahead

Fig. 3 illustrates the A/B control policy where F is the free distance ahead and v is the speed of the controlled vehicle. The green curves illustrate braking phases and the

black the accelerating phase from an initial speed V_1. For $F = F_1$, the maximal target speed V_1' is less than the limit speed V_L. The A/B policy consists in accelerating to V_1', and then braking until the vehicle stops having travelled exactly distance F_1. If the free distance ahead is $F = F_2$, the maximal target speed will be the limit speed V_L. The A/B policy will similarly accelerate first to the limit speed and then brake to stop at F_2. Finally, if $F = F_3 > F_2$, then after accelerating to the limit speed V_L, the vehicle will maintain constant speed V_L for distance $F_3 - A(V_1, V_L) - B(V_L)$ and then brake for the remaining distance to stop at F_3.

Theorem 2. *If the speed V of the vehicle is safe with respect to F, i.e.,$B(V) \leq F$, then the A/B policy is always safe and efficient for F.*

Proof. The safety proof is given by the arguments following Theorem 1. To prove efficiency, we consider three basic driving policies: the A/B policy, the B/A policy and the C/B policy. The other possible policies, such as accelerating, driving at constant speed, accelerating and then braking, can be obtained as combinations of the three basic ones. We show that the A/B policy yields the minimal travelling time.

We decompose the free distance ahead F into two segments: one segment of length $D = F - B(V)$ and one segment of length $B(V)$. Due to the additivity property, the distance $B(V)$ is always required regardless of the applied polices in order to brake safely from speed V. So the policies may differ only in the time needed to travel distance D. In the A/B policy, the vehicle travels distance D by first accelerating to the maximal target speed v_M and then braking from v_M to V. We denote by t_A the time needed to accelerate from V to v_M and by t_B the time needed to brake from v_M to V. In the C/B policy, the vehicle first moves at constant speed V for the distance D and then brakes from V for the remaining distance $B(V)$. We denote by t_D the time needed to travel D with constant speed V. We show that t_D is greater than $t_A + t_B$. We denote the speed function during acceleration by $v'(t)$ and the speed function during deceleration by $v''(t)$. Then we have $v'(t) > v$ for $t_A > t > 0$ and $v''(t) > v$ for $t_B > t > 0$. Since $D = v * t_D = \int_0^{t_A} v'(t)dt + \int_0^{t_B} v''(t)dt > \int_0^{t_A} vdt + \int_0^{t_B} vdt = v * (t_A + t_B)$, we have $t_D > t_A + t_B$. Thus the A/B policy takes less time and it is more efficient than the C/B policy.

In the B/A policy, the vehicle first brakes and accelerates for the distance D and then brakes for the remaining distance $B(V)$. We denote by $t'_B + t'_A$ the travelling time of D by braking and accelerating. By applying a similar reasoning, we can show that $t'_B + t_A > t_D$. Thus the C/B policy is more efficient than the B/A policy. This concludes the proof.

The above result implies that for the given free distance F, the A/B policy is the most efficient and that from the given initial speed there is a maximal speed that minimizes the travel time of F.

4 Controller design for collision avoidance

4.1 The control principle

We study a control principle for collision avoidance based on the above results. We consider that the vehicle speed can change between a finite set of increasing levels

$v_0, v_1, ..., v_n$, where n is a constant, $v_0 = 0$ and v_n equals to the limit speed v_L. The triggering of acceleration and braking from one level to another is controlled according to the free distance ahead and based on bounds computed as follows, for each speed level $v_i, i \in [1, n]$,

- $B_i = B(v_i)$ is the minimal safe braking distance needed for the vehicle to fully stop from speed v_i;
- $D_i = A(v_{i-1}, v_i) + B(v_i)$ is the minimal safe distance needed for the vehicle to apply an A/B policy accelerating from speed v_{i-1} to v_i and then braking from v_i to stop.

We show that the following function specifies the highest safe speed level v as a function of the current speed of the vehicle V and the free space ahead F, provided that their initial values V_0 and F_0 are such that $B(V_0) \leq F_0$.

$$v = Control(F, V)$$

$$v = \begin{cases} v_{i+1} & \text{when} \quad V = v_i \wedge F = D_{i+1} \\ v_{i-1} & \text{when} \quad V = v_i \wedge F = B_i \\ v_i & \text{when} \quad V = v_i \wedge D_{i+1} > F > B_i \end{cases}$$

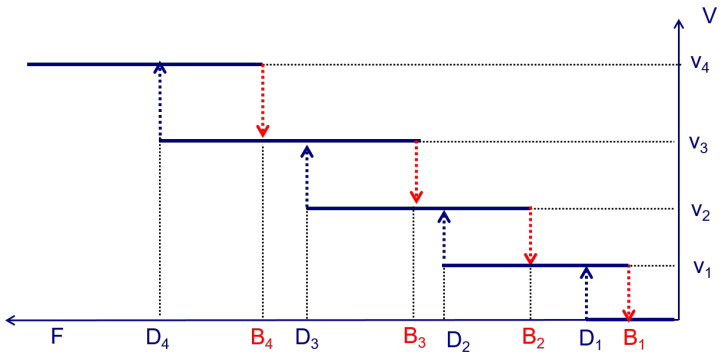

Fig. 4. Illustration of the collision avoidance principle for $n = 4$

Note that this control principle is purely functional. It assumes that changes of the free distance ahead F can be continuously monitored to instantaneously produce corre-spomding speed changes.

Fig.4 illustrates the principle for $n = 4$ speed levels. As the value of F increases, the speed of the vehicle switches between levels. Safety is preserved by construction. The vehicle can accelerate to a higher level, if it can safely and efficiently use the available distance by applying an A/B policy. It brakes to a lower level if the available distance reaches the bound for safe braking.

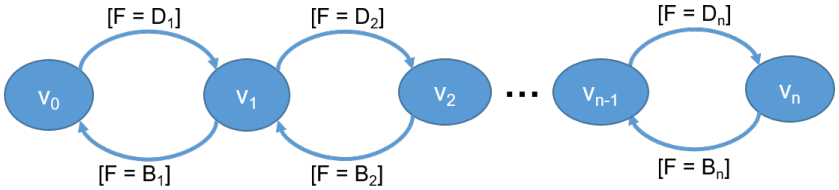

Fig. 5. Automaton modelling the collision avoidance principle

Fig.5 provides a scheme for the computation of $Control(F, V)$ in the form of a finite state automaton. The locations correspond to traveling at constant speeds $v_0, ..., v_n$. The transitions model instantaneous acceleration and braking steps triggered by conditions involving the free distance F and the precomputed bounds B_i and D_i. If the control location is v_i and the free distance ahead equals to the minimal safe acceleration distance (i.e., $F = D_{i+1}$), then the automaton moves to location v_{i+1} after the speed is accelerated to v_{i+1}. If the free distance ahead reaches the minimal safe braking distance (i.e., $F = B_i$), then the automaton moves to location v_{i-1} after the speed is decelerated to v_{i-1}. Recall that $B_i = B_{i-1} + B(v_i, v_{i-1})$. Thus, after braking to v_{i-1} there is still enough space for safe braking. Note that checking point conditions makes sense because F has no jumps and computation is instantaneous. If none of the triggering conditions holds, then the free distance ahead F is such that $B_i < F < D_{i+1}$. The automaton stays at location v_i and the speed remains unchanged.

Note that the automaton of Fig.5 cannot be implemented as a controller because we assume that F is continuously observable and changes of the controlled speed are instantaneous. In the next section, we show how to design controllers by refining this automaton.

Theorem 3. *The collision avoidance principle is safe. Moreover, its efficiency is strictly increasing for increasing number of speed levels* n.

Proof. Safety can be proved by induction on the number of speed levels. First, we prove that the transition to v_1 is safe. If $v = 0$ and the condition $F = D_1$ holds, speed can change to v_1. The transition to v_1 needs distance $A(v_0, v_1)$. When this speed is reached, the distance F' will be such that $F' \geq (F - A(v_0, v_1)) = B(v_1)$, since $D_1 = A(v_0, v_1) + B(v_1)$. Thus the vehicle is still safe at v_1 because the remaining distance is greater than the minimal safe braking distance.

Assuming safety for $v = v_i$, we prove safety for $v = v_{i+1}$. Safety for $v = v_i$ means that $F \geq B_i$ and it will remain safe as long as the speed level does not change. We distinguish two cases. If $F \geq D_{i+1} = A(v_i, v_{i+1}) + B(v_{i+1})$, then we can accelerate to speed v_{i+1} and the free headway distance will be $F' \geq (F - A(v_i, v_{i+1})) = B(v_{i+1})$, which implies that the safety condition still holds for v_{i+1}. If $B_{i+1} < F < D_{i+1}$, then the speed remains unchanged and there is enough distance to brake for v_{i+1}. If $F = B_{i+1}$, then the vehicle will brake to the lower speed v_i, which is safe by assumption. Thus it is also safe for $v = v_{i+1}$.

Note that as speed can be enforced to discrete levels, efficiency is achieved only when $F = D_i$ for some i; otherwise, the highest safe speed level is chosen. Let V_{init} be the speed of the vehicle and $F_i = A(V_{init}, V_i) + B(V_i)$ be the distance needed for the application of an A/B policy from V_{init} to V_i. Then we consider two cases:

- Either $F = F_{d1}$ such that $B(V_{init}) \leq F_{d1} \leq F_n$ in which case by construction there exists some v_i such that $v_i \leq V_{d1} \leq v_{i+1}$ as shown in Fig.3, where V_{d1} is the maximal speed such that $F_{d1} = A(V_{init}, V_{d1}) + B(V_{d1})$. In that case the controller will apply the best A/B policy to reach from V_{init} the speed level V_i. The loss in efficiency $V_{d1} - V_i$ is determined by the max of the difference $V_{i+1} - V_i$ for $i \in [1, n]$. Thus, for increasing number of speed levels, the efficiency increases.
- Or $F = F_{d2}$ such that $F_n < F_{d2}$. In that case the controller will accelerate to the allowed limit speed v_n and then will keep the speed constant for distance $F_C = F_{d2} - F_n$. Then if the obstacle ahead is fixed, it will have to brake for distance $B(v_n)$. In that case there is no loss of efficiency as the limitation comes from the limit speed of the vehicle.

This concludes the proof.

As explained, computing the exact value of the optimal speed for a given distance may be costly. Considering discrete speed levels allows pre-computing for each level both the minimal safe braking distance and the minimal safe accelerating distance between levels. In that manner, we avoid the computational complexity of adjusting in real time the vehicle speed.

4.2 Controller design

We propose two controllers applying the presented collision avoidance principle. The first controller is synchronous driven by periodic updates of the free distance variable F for an adequately chosen period. The second controller is asynchronous in the sense that the free distance variable F is updated sporadically.

Synchronous controller The controller interacts with its controlled environment (the vehicle) through input and output events as shown in Fig.7. The output s is a state variable indicating the currently applied command (i.e., accelerating, braking or constant speed). The input event $UpdateF$ signals the periodic measurement F' of the free distance with period T, while input events ca and cb signal the completion of the accelerating and braking command respectively. Initially, the speed v of the vehicle is set to a level v_i that is safe with respect to the initial distance F (i.e., $F \geq B(v_i)$).

The controller is a refinement of the ideal controller where we assumed that speed changes were instantaneous. Its is described by the following set of guarded commands

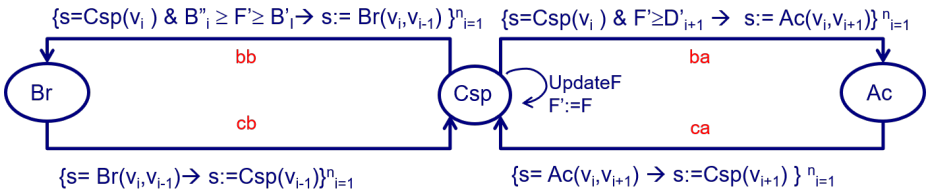

Fig. 6. Extended automaton modelling the synchronous controller

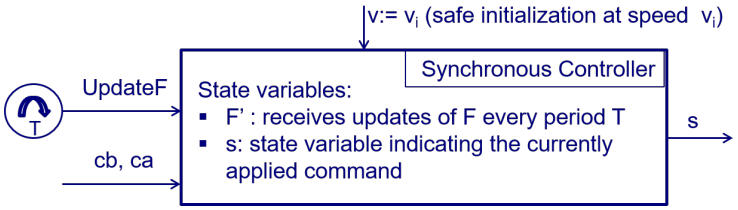

Fig. 7. Inputs and outputs of the synchronous controller

and also depicted as an extended automaton for the sake of clarity in Fig.6.

$$
\begin{aligned}
&do\\
&\Box\ \exists i \in [1,n].s = Csp(v_i) \wedge F' \geq D'_{i+1}\\
&\qquad \rightarrow s := Ac(v_i, v_{i+1});\ ba\\
&\Box\ \exists i \in [1,n].s = Csp(v_i) \wedge B''_i \geq F' \geq B'_i\\
&\qquad \rightarrow s := Br(v_i, v_{i-1});\ bb\\
&\Box\ \exists i \in [1,n].ca \wedge s = Ac(v_i, v_{i+1}) \rightarrow s := Csp(v_{i+1})\\
&\Box\ \exists i \in [1,n].cb \wedge s = Br(v_i, v_{i-1}) \rightarrow s := Csp(v_{i-1})\\
&\Box\ UpdateF \rightarrow F' := F\\
&od
\end{aligned}
$$

For guarded commands we adopt the usual semantics: whenever the condition on the left hand side holds, the actions on the right hand side are executed. Note that the input events appear as conditions while the output event appear as actions. The variable s keeps track of the kinematic state of the vehicle that is abstracted by the control states **Ac** (accelerating), **Br** (braking) and **Csp** (moving with constant speed).

We denote by $Ac(v_i, v_{i+1})$, $Br(v_i, v_{i-1})$ and $Csp(v_i)$ the commands of accelerating from speed v_i to v_{i+1}, braking from v_i to v_{i-1} and moving with speed v_i, respectively. When the vehicle is moving with constant speed, transition **UpdateF** is triggered periodically to receive the most recent measurement of F. Once the triggering condition of accelerating (braking) is met, transition **ba** (**bb**) is taken to initiate the command and move to location **Ac** (**Br**) waiting for its completion.

We do not make any assumption about the time spent at locations **Ac** and **Br**. We simply assume that the distances needed for accelerating and braking are $A(v_{i-1}, v_i)$

and $B(v_i, v_{i-1})$, respectively. We explain below how the guards of the controllable transtions bb and ba are computed.

We estimate the maximal safe approximations of the triggering conditions $F \geq D_i$ and $F = B_i$ of the ideal controller in terms of F', the most recently updated value of F. When the vehicle moves at speed v_i, the variables F and F' satisfy a relation of the form $F = F' - k_i(t)$, where $k_i(t) = v_i * (t \ mod \ T)$. That is $k_i(t) = 0$ when F is updated and $k_i(t) < v_i * T$. We assume that T is small enough so that $D_i - v_i * T \geq B_i$, that is, we do not miss the braking threshold value B_i in a period. This is reasonable given that in practice the updating period T is usually less than 50 milliseconds. Notice that the minimal value of F will be reached for $F = F' - v_n * T$. Thus, it is enough to require that $F' \geq D_i + v_n * T$ holds for accelerating and that $B_i + 2 * v_n * T \geq F' \geq B_i + v_n * T$ holds for braking. So we adjust the triggering bound for accelerating to $D'_i = D_i + v_n * T$ and the least and upper bounds of the interval triggering a braking to $B'_i = B_i + v_n * T, B''_i = B_i + 2 * v_n * T$.

Asynchronous controller Inputs and outputs of the asynchronous controller shown in Fig.8 differ in that the input event $UpdateF$ receiving the measurement F' of the free distance occurs sporadically. Furthermore, this controller needs an internal clock event $tick$ with period $\triangle t$ to estimate the vehicle's position.

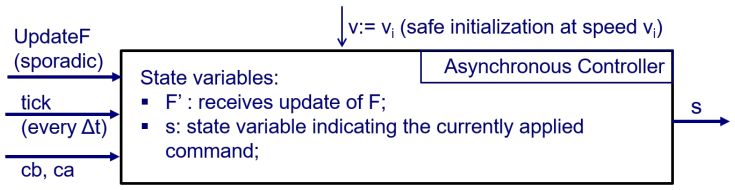

Fig. 8. Inputs and outputs of the asynchronous controller

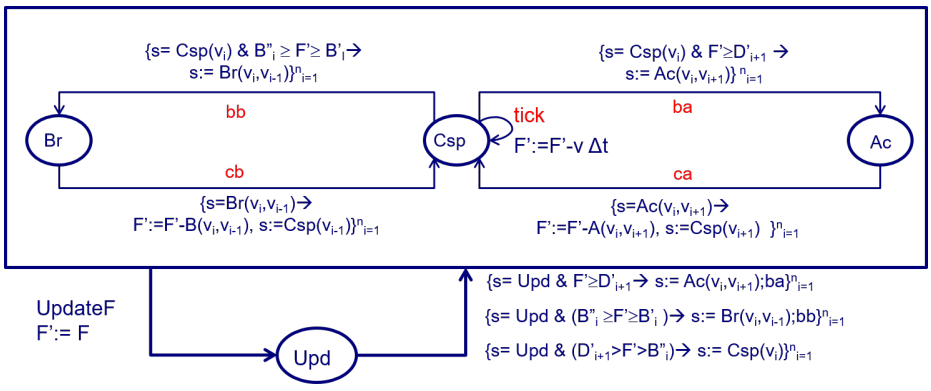

Fig. 9. Extended automaton modelling the asynchronous controller

The asynchronous controller differs from the synchronous controller in the way of estimating the values of the free distance F. Its is described by the following guarded commands and also depicted as an extended automaton in Fig.9.

$$
\begin{array}{l}
do \\
\Box\ \exists i \in [1,n].(s = Csp(v_i) \vee s = Upd) \wedge F' \geq D'_{i+1} \\
\qquad \rightarrow s := Ac(v_i, v_{i+1});\ ba \\
\Box\ \exists i \in [1,n].(s = Csp(v_i) \vee s = Upd) \wedge B''_i \geq F' \geq B'_i \\
\qquad \rightarrow s := Br(v_i, v_{i-1});\ bb \\
\Box\ \exists i \in [1,n].s = Upd \wedge D'_{i+1} > F' > B''_i \rightarrow s := Csp(v_i) \\
\Box\ \exists i \in [1,n].ca \wedge s = Ac(v_i, v_{i+1}) \\
\qquad \rightarrow F' := F' - A(v_i, v_{i+1}); s := Csp(v_{i+1}) \\
\Box\ \exists i \in [1,n].cb \wedge s = Br(v_i, v_{i-1}) \\
\qquad \rightarrow F' := F' - B(v_i, v_{i-1}); s := Csp(v_{i-1}) \\
\Box\ \exists i \in [1,n].tick \rightarrow F' := F' - v_i * \triangle t \\
\Box\ UpdateF \rightarrow F' := F; s := Upd \\
od
\end{array}
$$

As previously we adopt similar notations. The variable s keeps track of the kinematic state of the vehicle that is abstracted by the control states of the automaton **Ac** (accelerating), **Br** (braking) and **Csp** (moving with constant speed).

As the updates of F are sporadic, we use a local variable F' to keep track of the possible changes of F since its latest update as follows. When the vehicle completes an accelerating or braking step (i.e., when the completion transitions ca or cb occur), F' is updated by $F' - A(v_i, v_{i+1})$ and $F' - B(v_i, v_{i-1})$, respectively. When the vehicle is moving with constant speed v, F' is updated by $F' - v * \triangle t$ for every time period $\triangle t$, where $\triangle t$ is a time constant such that the upper bound of the uncertainty $\epsilon = v_n * \triangle t$ in the estimation of F remains small.

When an **UpdateF** occurs, from any state the controller moves to an update state **Upd**. This is represented by grouping the three locations of the automaton into a macro state with an outgoing transition to location **Upd**. Then without delay from location **Upd**, the controller compares F' to the corresponding bounds and moves to a safe target location accordingly.

As for the synchronous controller, the upper and lower bounds of the triggering conditons of **bb** and **ba** are modified so as to take into account the uncertainty ϵ in the computation of F. In fact if x is the remaining distance to travel since the last update of F, we have $x \geq F' \geq x - \epsilon$. So the condition for accelerating to speed v_{i+1} from v_i becomes $F' \geq D'_{i+1}$, where $D'_{i+1} = D_{i+1} + \epsilon$ and the condition for braking from v_i becomes $B''_i \geq F' \geq B'_i$ where $B''_i = B_i + 2 * \epsilon$ and $B'_i = B_i + \epsilon$. The safety argument still holds for this asynchronous controller because it simply applies after each update of F an A/B policy for the considered speed levels.

Theorem 4. *Both the synchronous and the asynchronous controller yield safe control policies for collision avoidance.*

Proof. The safety proof for the synchronous controller follows the same reasoning as in the previous theorem .

The safety proof for the asynchronous controller is by induction on the updates of the free distance F. Assume that an update starts and the vehicle is at speed level v_i that is safe for $F' = F$. Then until the next update occurs, the controller will keep track of the free distance by updating F' in every $\triangle t$ time: $F' := F' - v_i * \triangle t$ and in that manner at any time F' will be such that $F' + \epsilon \geq x$, where x is the remaining distance to safely travel since the last update of F. The controller applies an A/B policy, which is safe because the conditions for accelerating and braking have been modified to take into account the maximal deviation ϵ. So, the vehicle will move safely until the next update or stop after F' reaches a value $B_1'' \geq F' \geq B_1'$ and trigger the braking command $Br(v_1, 0)$.

Following the same reasoning as in the previous theorem, we can deduce that the efficiency of the two controllers strictly increases for increasing number of speed levels n. Furthermore, the efficiency depends on how frequently F is updated as the accelerating and braking conditions take into account the uncertainty about the values of F. Thus, the controllers may not be able to fully utilize actually available free distance. While as explained for synchronous controller, the loss in efficiency depends on the value of $v_n * T$, for the asynchronous one it depends on the maximal time difference between two successive updates of F.

5 Experimental evaluations

We have implemented both the synchronous and the asynchronous controller in the open-source autonomous driving simulator Carla [4]. In the experiments, we consider scenarios where the controlled vehicle is driving towards a moving vehicle ahead as shown in Fig.10. The speed of the front vehicle is described by the periodic function $v_f(t) = v_{f0} + v_{f0} * sin(\omega * t)$, where $\omega = 2 * \pi / T_f$, T_f is the period of this speed function and v_{f0} is a constant. We choose $v_{f0} = 14\ m/s$, and thus the speed of the front vehicle changes in the interval $[0, 28\ m/s]$ (i.e., $[0, 100.8\ km/h]$). We set the limit speed of the controlled vehicle to be $32\ m/s$ (i.e., $115.2\ km/h$). The initial distance between the two vehicles is $F(0) = 5m$ and the initial speed of the controlled vehicle is 0. Thus, the controlled vehicle is initially at a safe state. The accelerating and braking rates of the two vehicles are both constant $a = b = 2\ m/s^2$.

In order to evaluate the performance and the quality of the controllers, we measure both the speed changes of the controlled vehicle and the relative distance between the two vehicles, reflecting the occupancy of the road. The smaller the distance is, the higher the road occupancy is. We perform experimental evaluations in two settings.

- In **Setting 1**, we consider that the free distance ahead is equal to the relative distance between the two vehciles, that is we ignore the speed of the front vehicle. This corresponds to a strict safety policy that avoids collision even when the front vehicle suddenly stops e.g. in case of accident.
- In **Setting 2**, we consider that the free distance is the relative distance increased by the braking distance of the front vehicle.

Fig. 10. Simulation environment in Carla

In both settings, we perform the evaluations with respect to three parameters: the period T_f of v_f, the number of speed levels n of the controlled vehicle and the period T of sensing the free distance. For experimental purposes, the safe accelerating and braking distances for eight speed levels $v[8] = \{4, 8, 12, 16, 20, 24, 28, 32\}$ are pre-computed as shown in Table.1 and configured in the implementations. The distances are obtained for constant accelerating and braking rates $a = b = 2\ m/s^2$.

Table 1. Safe accelerating and braking distances for the eight speed levels

speed level (m/s)	Accelerating distance (m)	Braking distance (m)	Distance for A/B policy
v_1 = 4	$A(v_0, v_1)$ = 4	$B(v_1)$ = 4	$D(v_0, v_1)$ = 8
v_2 = 8	$A(v_1, v_2)$ = 12	$B(v_2)$ = 16	$D(v_1, v_2)$ = 28
v_3 = 12	$A(v_2, v_3)$ = 20	$B(v_3)$ = 36	$D(v_2, v_3)$ = 56
v_4 = 16	$A(v_3, v_4)$ = 28	$B(v_4)$ = 64	$D(v_3, v_4)$ = 92
v_5 = 20	$A(v_4, v_5)$ = 36	$B(v_5)$ = 100	$D(v_4, v_5)$ = 136
v_6 = 24	$A(v_5, v_6)$ = 44	$B(v_6)$ = 144	$D(v_5, v_6)$ = 188
v_7 = 28	$A(v_6, v_7)$ = 52	$B(v_7)$ = 196	$D(v_6, v_7)$ = 248
v_8 = 32	$A(v_7, v_8)$ = 60	$B(v_8)$ = 256	$D(v_7, v_8)$ = 316

Setting 1: First, we evaluate how T_f affects the performance for two different values $T_f = 10\ s$ and $T_f = 30\ s$. We assume that the environment updates the free distance variable with period $T = 0.02\ s$, and that the asynchronous controller estimates this variable with period $\triangle t = 0.005\ s$ (as shown in the controller in Fig.9).

Fig.11 compares the simulation results for the synchronous (left part) and the asynchronous controller (right part). The top two figures compare the dynamics of the free distances. For both controllers, the relative distance is periodic with period T_f in the steady regime. It decreases for increasing period T_f. For instance, it is around $57.27\ m$ for $T_f = 10\ s$, while for $T_f = 30\ s$ it is $20.11\ m$. In fact, for slower speed changes, the controller has more time to adjust the movement of the controlled vehicle and can better utilize the available distance.

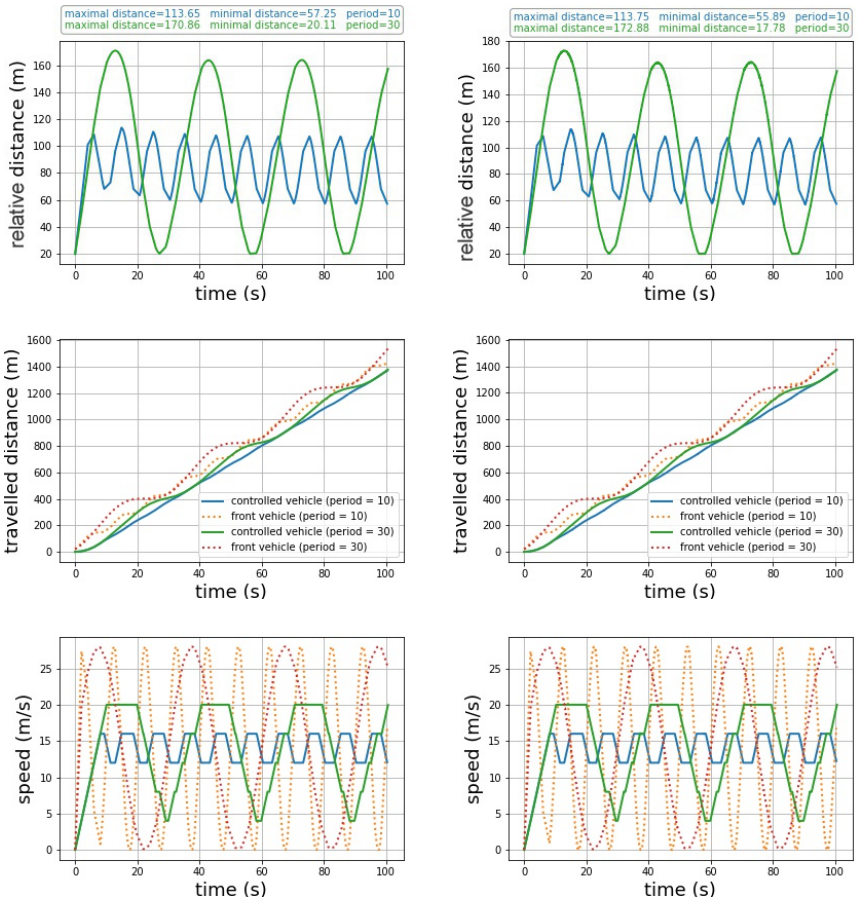

Fig. 11. Simulation results for the synchronous (left part) and the asynchronous controller (right part) for $T_f \in \{10\ s, 30\ s\}$ and $n = 8$ speed levels and sensing period $T = 0.02\ s$.

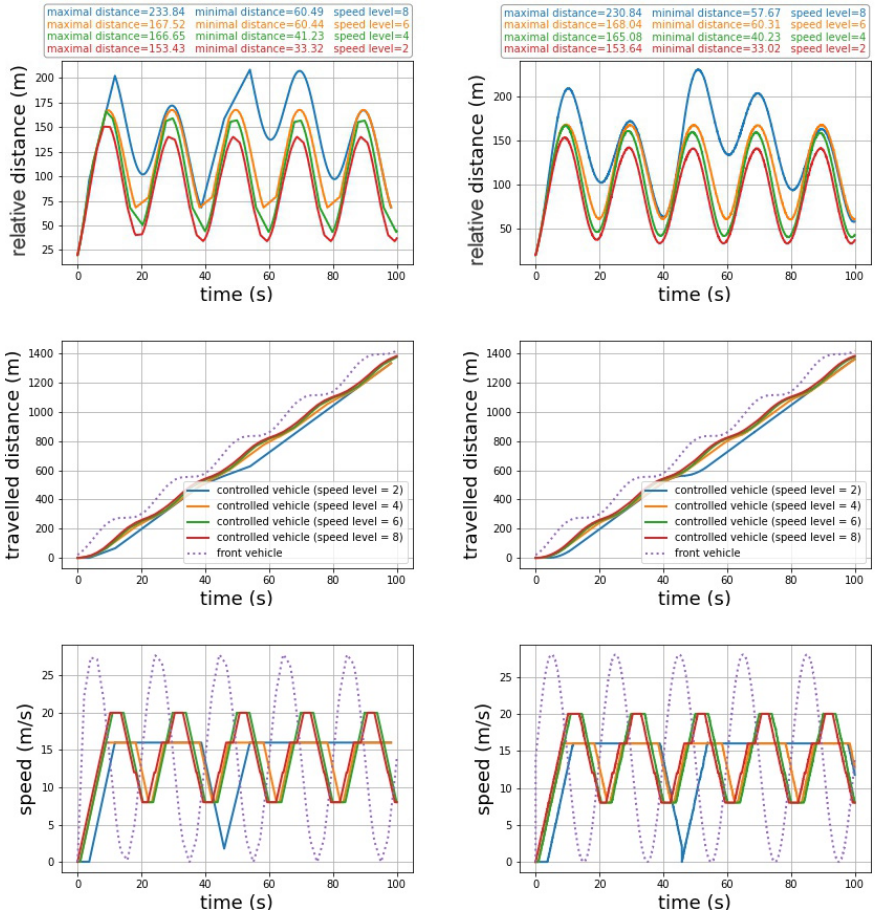

Fig. 12. Simulation results for the synchronous (left part) and the asynchronous (right part) controller for four different speed levels ($n \in \{2, 4, 6, 8\}$) and $T_f = 20\ s, T = 0.02\ s$.

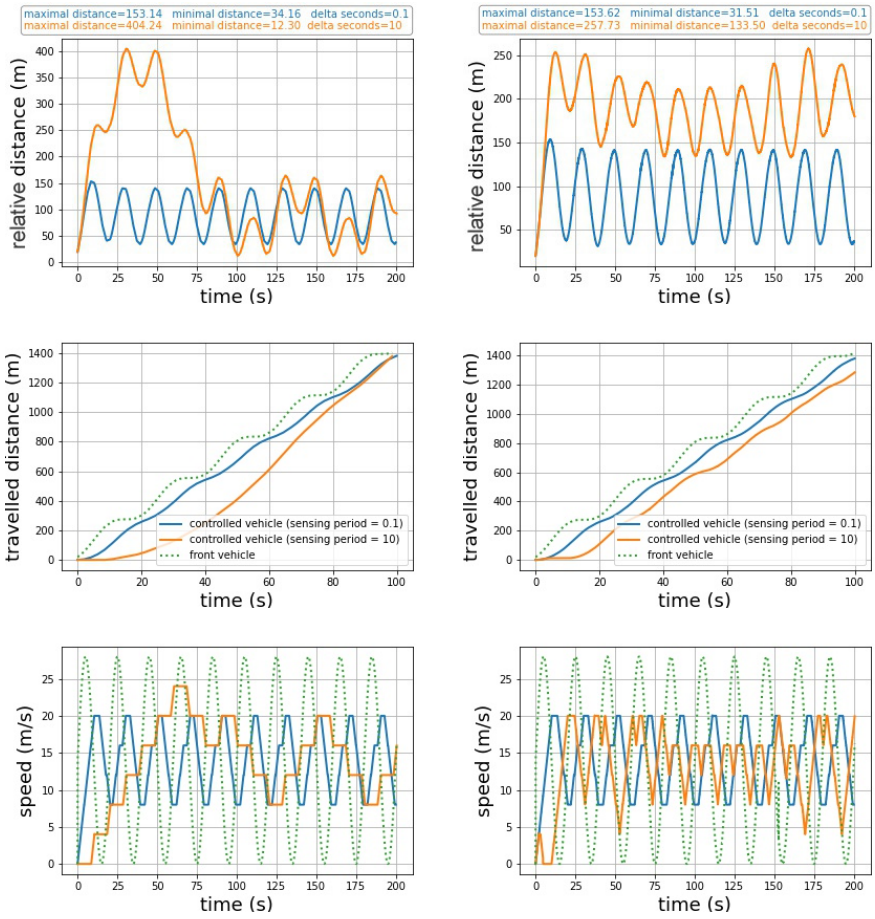

Fig. 13. Simulation results for the synchronous (left part) and the asynchronous (right part) controller for two different sensing periods ($T \in \{0.1\ s, 10\ s\}$) and $T_f = 20\ s$, $n = 8$.

Note that the results are similar for the two controllers when period T_f changes. A minor difference is that the minimal relative distances are smaller for the asynchronous controller: when $T_f = 30\ s$, the minimal relative distance of $20.11\ m$ for the synchronous controller reduces to $17.78\ m$ for the asynchronous controller. The bottom figures compare the speed changes of the controlled vehicle (solid lines) in response to the speed changes of the front vehicle (dotted lines). Similarly, the speed of the controlled vehicle gets closer to v_f for increasing period T_f. For both controllers, the maximal speeds increase from $16\ m/s$ to $20\ m/s$ when the period increases from $10\ s$ to $30\ s$.

In the second set of experiments, we evaluate how the number of speed levels n affects the performance of the two controllers for four different values of $n = 2$, $n = 4$, $n = 6$ and $n = 8$. The period of the front vehicle's speed function is $T_f = 20\ s$ and the sensing period is $T = 0.02\ s$.

The results for the synchronous controller are shown in the left part of Fig.12. We can see that for decreasing number of speed levels, the relative distance increases. For instance, when $n = 8$, the minimal relative distance is $33.32\ m$, which becomes 60.49 when $n = 2$. Thus, occupancy deteriorates when less speed levels are used. This result simply confirms a consequence of Theorem 3. The bottom figure compares the speed changes of the controlled vehicle in response to the speed changes of the front vehicle. The result similarly shows that the maximal speed of the controlled vehicle increases from $16\ m/s$ to $20\ m/s$ with the number of speed levels. However, beyond a certain number, the performance improvement is negligible: e.g., the speed curves for $n = 6$ and $n = 8$ are almost identical.

For the asynchronous controller, similar results are shown in the right part of Fig.12. For decreasing number of speed levels, the relative distance increases. The minimal relative distance changes from 33.02 to $57.61\ m$ when the speed level reduces from 8 to 2. These benchmarks show that varying the number of speed levels does not result in significant performance differences between synchronous and asynchronous controller. However, the relative distances for the considered speed levels are slightly smaller for the asynchronous controller.

In the third set of experiments, we evaluate how the period T of sensing the free distance ahead affects the performance for $T = 0.1\ s$ and $T = 10\ s$ with $n = 8$ and $T_f = 20\ s$. The results for the synchronous controller are shown in the left part of Fig.13. Note the transient behavior when the period T increases. In the speed diagram the controlled vehicle accelerates from 0 to the highest speed $24\ m/s$ because the largest relative distance is reached only in the transient phase. Furthermore, for increasing period, the range of the relative distance increases. When $T = 0.1\ s$, the relative distance changes in the interval $[34.16, 153.14]\ m$, which becomes $[12.30, 161.60]\ m$ when $T = 10\ s$ as the uncertainty about the free distance increases with the period.

For the asynchronous controller, the results are shown in the right part of Fig.13. Note that period of sensing has significant impact on the performance of the asynchronous controller. For increasing period, the relative distance considerably increases. Furthermore, compared to the synchronous controller, there is no obvious oscillation during the transient phase. This is because the asynchronous controller computes an estimation of the free distance at each local time step. As a result, the controlled ve-

hicle will not accelerate to the highest speed 24 m/s, which is the possible for the synchronous controller.

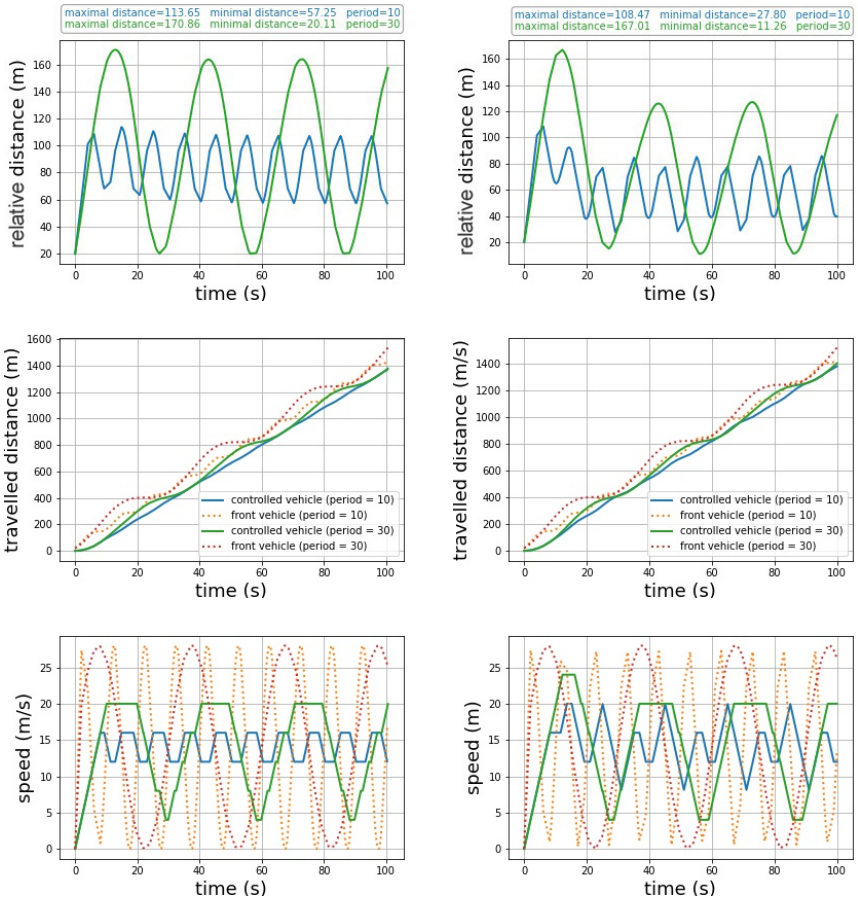

Fig. 14. Simulation results for the synchronous controller with and without considering the braking distance of the front vehicle (right and left part, respectively) for $T_f \in \{10\ s, 30\ s\}$.

Setting 2: in the subsequent experiments, we evaluate how performance changes when we take into account in the evaluation of the free distance the braking distance of the front vehicle travelling with speed $v_f(t) = v_{f0} + v_{f0} * sin(\omega * t)$. At time t its braking distance is $v_f(t)^2/2{*}b_f$ for a constant braking rate b_f. We take $b_f = 5m/s^2$ for experimental purposes. Then the corresponding safe accelerating and braking distances are $D_i' = D_i - v_f(t)^2/2 * b_f$ and $B_i' = B_i - v_f(t)^2/2 * b_f$.

In Fig.14, Fig.15 and Fig.16, we compare the performance of the synchronous controller with and without considering the braking distance of the front vehicle (right and left part, respectively). In Fig.14 we compare the results for $T_f \in \{10\ s, 30\ s\}$ with

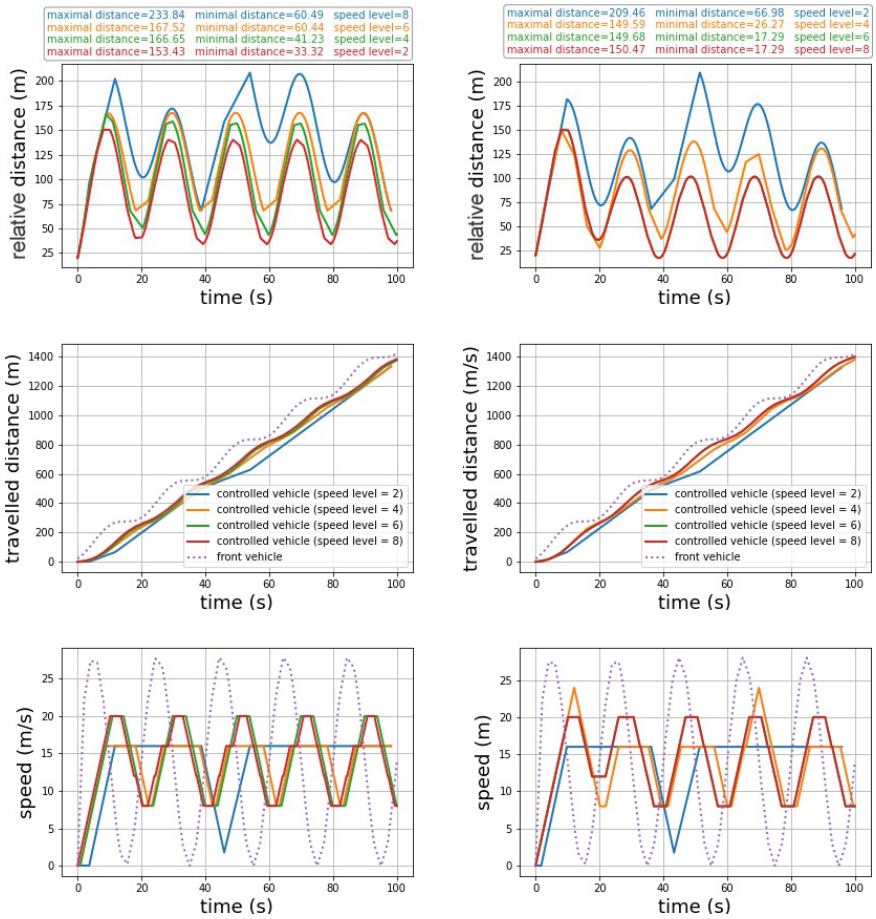

Fig. 15. Simulation results for the synchronous controller with and without considering the braking distance of the front vehicle (right and left part, respectively) for four different speed levels $n \in \{2, 4, 6, 8\}$.

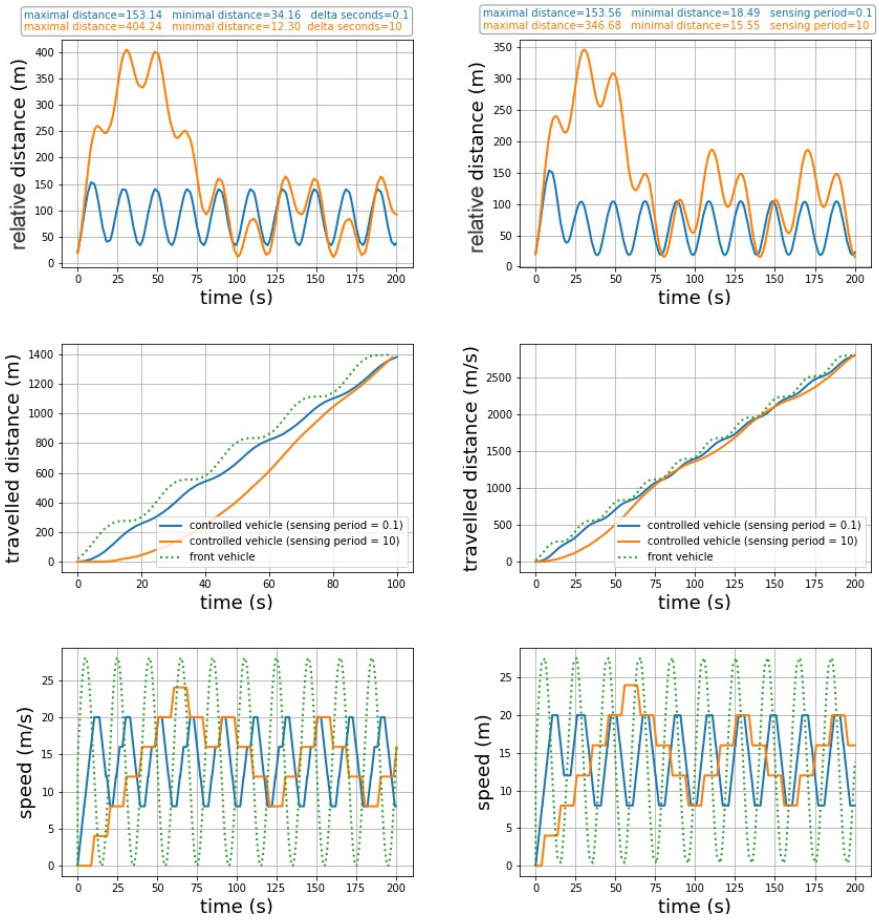

Fig. 16. Simulation results for the synchronous controller with and without considering the braking distance of the front vehicle (right and left part, respectively) for two sensing periods $T \in \{0.1\ s, 10\ s\}$.

sensing period $T = 0.02\ s$ and speed level $n = 8$. We can see that when taking into account the braking distance of the front vehicle, the relative distance between the two vehicles becomes much smaller. For instance, the minimal relative distance decreases from $20.11\ m$ for $T_f = 30\ s$ to $11.26\ m$. Furthermore, the distance decreases when the period T_f increases as shown in the top figures in Fig.14. The performance improvement can also be observed from the speed diagrams shown in the bottom of Fig.14. Finally, the speed range of the controlled vehicle becomes larger and the maximal speed increases from $16\ m/s$ to $20\ m/s$ for $T_f = 10\ s$.

Similar results are shown in Fig.15 and Fig.16, which compare the results for four different numbers of speed levels $n \in \{2, 4, 6, 8\}$ and two different sensing periods $T \in \{0.1\ s, 10\ s\}$, respectively. The relative distance reduces significantly when taking into account the braking distance of the front vehicle. For instance, the minimal free distance drops from $33.32\ m$ to $17.29\ m$ when $n = 8$. These results confirm that by taking into account the movements of the front vehicle, the controlled vehicle can move more aggressively and better utilize the free distance ahead.

6 Conclusions and future work

The paper presents a novel framework and approach for safe and efficient collision avoidance for self-driving vehicles. The framework is model-based and assumes that control policies are implemented as the application of acceleration, braking and constant speed commands. The presented algorithms do not make any assumption about the dynamics of the controlled vehicle except that there are functions giving the traveled distance when the speed of the vehicle changes by some quantity. Additionally, the assumptions about the vehicle's environment are minimal as it is described by a function $F(t)$ giving at any time the free available distance ahead.

The assumption that all vehicles move in the same direction as the controlled vehicle does not limit the generality of our approach. The same algorithm can be applied by adequately taking into account movements in the different directions in the estimation of the free headway distance. If for instance the distance between the controlled vehicle and the front vehicle moving in the opposite direction is F' at time t, then the free space ahead can be estimated as $F = F' - D(\triangle t)$, where $\triangle t$ is the time needed for the controlled vehicle to completely stop by braking from its current speed, and $D(\triangle t)$ is the maximal distance travelled by the front vehicle within time $\triangle t$.

The same algorithm can be adapted to two-dimensional movement. In that case, the function $F(t)$ can be defined as the maximal convex area containing the controlled vehicle and such that all the obstacles are outside this area. The distances $A(V, v)$ and $B(V, v)$ for initial and target speeds respectively are also replaced by adequately approximated convex areas so that the safety test boils down to area inclusion that can be efficiently decided.

The presented approach differs from others based on control theory or controller synthesis in that it guarantees at a high level of abstraction both safety and efficiency. We progressively relax the assumption about perfect real-time knowledge of the free space and provide solutions that are safe and relatively efficient even when the free space is sporadically updated. Furthermore, switching between a set of speed levels depending

on pre-computed conditions drastically reduces the computational complexity of the decision process. This also allows to reduce the control algorithm sensitivity to changes of the environment while keeping driving safe and robust. Experimental results show that it is easy to get implementations under minimal assumptions about the operational environment.

This work is part of a larger project on the design of safe and efficient autopilots for self driving cars. Future developments include the adaptation of this algorithm to two-dimension movement where the free space function provides areas around the controlled vehicle as well as the integration of our algorithms in autonomous car models of the Carla simulator.

References

1. Althoff, M., Maierhofer, S., Pek, C.: Provably-correct and comfortable adaptive cruise control. IEEE Transactions on Intelligent Vehicles PP, 1–1 (05 2020)
2. Ames, A.D., Grizzle, J.W., Tabuada, P.: Control barrier function based quadratic programs with application to adaptive cruise control. In: 53rd IEEE Conference on Decision and Control. pp. 6271–6278. IEEE (2014)
3. Asplund, M., Manzoor, A., Bouroche, M., Clarke, S., Cahill, V.: A formal approach to autonomous vehicle coordination. In: International Symposium on Formal Methods. pp. 52–67. Springer (2012)
4. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. pp. 1–16 (2017)
5. Esterle, K., Aravantinos, V., Knoll, A.: From specifications to behavior: Maneuver verification in a semantic state space. arXiv preprint arXiv:1905.00708 (2019)
6. Hilscher, M., Linker, S., Olderog, E.R., Ravn, A.P.: An abstract model for proving safety of multi-lane traffic manoeuvres. In: International Conference on Formal Engineering Methods. pp. 404–419. Springer (2011)
7. Horowitz, R., Varaiya, P.: Control design of an automated highway system. Proceedings of the IEEE 88(7), 913–925 (2000)
8. Korssen, T., Dolk, V., van de Mortel-Fronczak, J., Reniers, M., Heemels, M.: Systematic model-based design and implementation of supervisors for advanced driver assistance systems. IEEE Transactions on Intelligent Transportation Systems 19(2), 533–544 (2017)
9. Krook, J., Svensson, L., Li, Y., Feng, L., Fabian, M.: Design and formal verification of a safe stop supervisor for an automated vehicle. In: 2019 International Conference on Robotics and Automation (ICRA), Palais des congres de Montreal, Montreal, Canada. pp. 5607–5613 (2019)
10. Li, L., Peng, X., Wang, F.Y., Cao, D., Li, L.: A situation-aware collision avoidance strategy for car-following. IEEE/CAA Journal of Automatica Sinica 5(5), 1012–1016 (2018)
11. Li, L., Lu, G., Wang, Y., Tian, D.: A rear-end collision avoidance system of connected vehicles. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 63–68. IEEE (2014)
12. Loos, S.M., Platzer, A., Nistor, L.: Adaptive cruise control: Hybrid, distributed, and now formally verified. In: International Symposium on Formal Methods. pp. 42–56. Springer (2011)
13. Lygeros, J., Godbole, D.N., Sastry, S.: Verified hybrid controllers for automated vehicles. IEEE transactions on automatic control 43(4), 522–539 (1998)

14. Milanés, V., Pérez, J., Godoy, J., Onieva, E.: A fuzzy aid rear-end collision warning/avoidance system. Expert Systems with Applications 39(10), 9097–9107 (2012)
15. Nilsson, P., Hussien, O., Balkan, A., Chen, Y., Ames, A.D., Grizzle, J.W., Ozay, N., Peng, H., Tabuada, P.: Correct-by-construction adaptive cruise control: Two approaches. IEEE Transactions on Control Systems Technology 24(4), 1294–1307 (2015)
16. Nistér, D., Lee, H.L., Ng, J., Wang, Y.: The safety force field. NVIDIA White Paper (2019)
17. Park, J., Kim, D., Yoon, Y., Kim, H., Yi, K.: Obstacle avoidance of autonomous vehicles based on model predictive control. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 223(12), 1499–1516 (2009)
18. Rizaldi, A., Immler, F., Schürmann, B., Althoff, M.: A formally verified motion planner for autonomous vehicles. In: Lahiri, S.K., Wang, C. (eds.) Automated Technology for Verification and Analysis. pp. 75–90. Springer International Publishing, Cham (2018)
19. Sadraddini, S., Sivaranjani, S., Gupta, V., Belta, C.: Provably safe cruise control of vehicular platoons. IEEE Control Systems Letters 1(2), 262–267 (2017)
20. Shalev-Shwartz, S., Shammah, S., Shashua, A.: On a formal model of safe and scalable self-driving cars. CoRR abs/1708.06374 (2017), http://arxiv.org/abs/1708.06374
21. Stursberg, O., Fehnker, A., Han, Z., Krogh, B.H.: Verification of a cruise control system using counterexample-guided search. Control Engineering Practice 12(10), 1269–1278 (2004)
22. Varaiya, P.: Smart cars on smart roads: problems of control. IEEE Transactions on automatic control 38(2), 195–207 (1993)
23. Wang, P., Gao, S., Li, L., Sun, B., Cheng, S.: Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm. Energies 12(12), 2342 (2019)
24. Zhang, X., Liniger, A., Borrelli, F.: Optimization-based collision avoidance. arXiv preprint arXiv:1711.03449 (2017)
25. Zhao, C., Xing, Y., Li, Z., Li, L., Wang, X., Wang, F.Y., Wu, X.: A right-of-way assignment strategy to ensure traffic safety and efficiency in lane change (2019)
26. Zita, A., Mohajerani, S., Fabian, M.: Application of formal verification to the lane change module of an autonomous vehicle. In: 2017 13th IEEE Conference on Automation Science and Engineering (CASE). pp. 932–937. IEEE (2017)