2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Cooperative Collision Avoidance by Sharing Vehicular Subsystem Data

Chaitanya Yavvari, Zoran Duric, Duminda Wijesekera

George Mason University, Fairfax, VA
Email: {cyavvari,zduric,dwijesek}@gmu.edu

***Abstract*— Vehicular subsystems such as Anti-lock braking (ABS), Traction Control System (TCS) and Electronic Stability Program (ESP) use low-level sensory data to maintain the vehicle's stability. We show that sharing that information with neighboring vehicles could prevent potential collisions. Towards achieving this goal, we propose extensions to Dedicated Short Range Communication (DSRC)'s Basic Safety Message (BSM) set and propose a controller which can use the transmitted information to avoid collisions with vehicles experiencing instabilities. We show the utility of our methodology by using 3 common lane change scenarios under varying road surface conditions. Finally, we demonstrate some limitations of our methodology using similar scenarios.**

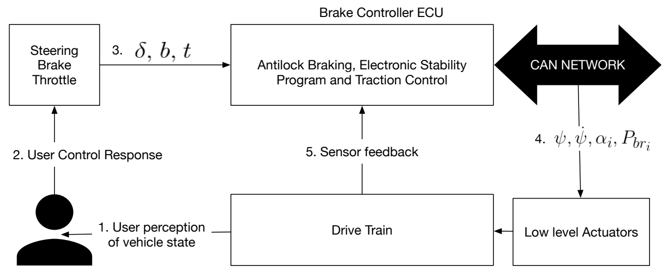

Fig. 1: The control loop model of vehicular subsystems

I. INTRODUCTION

The United States National Highway Traffic Safety Administration (NHTSA) released a notice of proposed rule making (NPRM) [1] for phasing in the implementation of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications starting in 2021, that identified six safety applications for crash avoidance. They are Forward Collision Warning, Emergency Electronic Brake Lights, Lane Change Warnings, Intersection Movement Assistance, *Do Not Pass* Notice Enforcement and Loss of Vehicular Control Warnings [2]. In order to enable these safety features, NHTSA proposed using the Dedicated Short Range Communication (DSRC) message set developed by the Society of Automotive Engineers (SAE) [3]. Most of the envisaged safety applications are supported by the Basic Safety Messages (BSM) of DSRC which is transmitted at a frequency of 10Hz. In addition to information standardized by the SAE [3], most modern vehicles use multiple control systems to monitor stability parameters and automatically adjust appropriate actuators to maintain vehicles stability in less than favorable road conditions. Examples of such systems are (1) Antilock Braking Systems (ABS) that mitigates wheel locking, slippage and sliding, (2) Traction Control Systems (TCS) that control excessive wheel spinning and (3) Electronic Stability Programs (ESP) that control lateral acceleration to limit yaw rates. These controllers continuously act to prevent potential instabilities at rates above the capabilities of human drivers (for example ABS systems monitor wheel locking 400 times per second and adjust brakes to prevent excessive locking). Thus these controllers are aware of vehicle's pending instabilities. Because instabilities of a vehicle can affect the safety of surrounding vehicles, if neighboring vehicles are alerted about potential instabilities, their automated safety controls may be able to act in order to avoid impending dangers, thereby going beyond Emergency Electronic Brake Lights (EEBL). In order to do so, we propose extending BSMs and illustrate a controller design that could take such messages in making safety maneuvers as shown in items #4 and #5 of Fig. 1.

We show the utility of our proposed extensions using three scenarios of *(1) aggressive neighboring vehicle, (2) loss of control at neighboring vehicles (3) multiple vehicles attempting lane change into same area*. We also show the limits of proposed cooperative accident avoidance using a fourth scenario. In order to evaluate the impact of our framework on the selected driving scenarios, we developed a fine grained simulation framework based on vehicle dynamics. The rest of the paper is organized as follows. Sec. II describes the background and related work. Sec. III describes the proposed control framework. Sec. IV describes our simulation framework. Sec. V provides our results.

II. BACKGROUND AND RELATED WORK

A. Basic Safety Messages (BSM)

BSM messages include static configuration attributes and dynamic operational attributes shown in Table I.

TABLE I: Basic Safety Message Content

Field	Description
$a_x, a_y, a_z, \dot{\psi}$	Lateral, Longitudinal, Vertical acceleration and Yaw rate
v	Speed
x, y	Position in (longitude,latitude)
θ	Heading angle
l_v, w_v	Vehicle length and width

Hobert et al. [4] enhanced ETSI standard for co-operative autonomous driving with V2V for co-operative services such

TABLE II: Proposed Extensions to Basic Safety Message

Field	Description
λ_i	Wheel slip at wheel i, $i = 0 \ldots 3$
σ_i	Slip angle at wheel i, $i = 0 \ldots 3$
ψ_d	Desired yaw rate computed by the stability control ECU

as convoy control, sensing support, lane changing and intersection control, and illustrated them using several scenarios.

Nilsson et al. [5] focused on ego-centric autonomous optimal lane changing strategies using point mass motion models and continuous constraint satisfaction. In contrast, we use combinations of suboptimal control actions (e.g aggressive overtaking, sudden braking etc.) by one or more vehicles in environments with low friction road surface to show the value of sharing low level control information in attaining cooperative safety. Additionally, our work uses V2V communication and is complimentary to existing work.

B. Stability Control Systems

Two measures of a moving vehicle's stability are the lateral slip angle (σ) of the vehicle and longitudinal slip (λ) of the wheels. The acceptable ranges of these vary for each vehicle model based on the vehicle's build. To maintain σ and λ within stable range, a moving vehicle controls many of its operational parameters such as pitch, roll, yaw, etc using multiple ECUs such as ABS, TCS and ESP. Vehicle responses such as pitch, roll, yaw are determined using the tire-terrain interaction which in turn is dependent on a coefficient of friction μ (ratio of the normal force at the tire contact to the sum of the lateral and longitudinal forces). All wheels of the vehicle should rotate without slipping or sliding by maintaining a sufficiently high μ. Usually, μ is around 0.9 on dry surfaces [6] but can exceed 1.0 for highly deformed tires. Vehicles are designed to accommodate a *bounded amount* of non-ideal stability measures with the support of control systems such as ABS, TCS and ESP with a control flow as shown in Fig. 1. The control loop consists of sensors on a vehicle gathering data transmitted to the respective ECUs over the Controller Area Network (CAN) network and the ECUs using control algorithms to asses stability and issue corrective actions via actuators. Now we provide basic mathematical formulation of vehicle's stability measures.

$$\sigma = \arctan \frac{v_y}{|v_x|} \tag{1}$$

$$\lambda = \frac{\omega r - v}{v} \tag{2}$$

where v_y, v_x and ω are the lateral, longitudinal and angular velocities of a wheel and r is the radius of the wheel. The ideal values of σ are close to 0 in straight driving maneuvers and small non-zero values in steering maneuvers. When σ increases, the influence of steering angle (δ) on the vehicle's yaw reduces loosing controllability. Also, when the slip increases to 1, the wheel locks and the vehicle begins to slide.

a) Anti-Lock Braking System (ABS): ABS distributes braking pressure during hard braking in order to avoid wheel locking and subsequent sliding of the vehicle by monitoring the longitudinal wheel slips at all the wheels. A μ-slip curve [7] characterizes the relationship between λ and μ of road surface. Ideally a μ-slip curve can be determined for each road surface condition and is useful for determining the reference slip ratio used by ABS. A *typical* road surface with $\mu = 0.9$ has $\lambda_o = 0.2$. Without ABS, when the longitudinal speed controller (driver actuation) chooses a non-zero braking value, all the wheels receive equal braking force as shown in Eq. (3) where $i = \{fl, fr, rl, rr\}$ represent the 4 wheels of the vehicle. If engaged ABS distributes the braking pressure given in Eq. (4).

$$\tau_{B_{fl}} = \tau_{B_{fr}} = \tau_{B_{rl}} = \tau_{B_{rr}} = b \times \tau_{wheel_{max}} \tag{3}$$

$$\tau_{B_i} = \begin{cases} b \times \tau_{wheel_{max}}, & \text{if } |\lambda_i| \leq \lambda_o \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

When engaged, ABS modulates brake pressure in order to control slip λ so that the wheel does not lock and thereby the effective brake force is maximized, typically programmed to happen in control cycles where the states of pressure hold and release are maintained for a small hold time t_h.

b) Traction Control System (TCS): TCS, engaged in situations where ABS cannot be engaged because the driver has not applied brakes. It monitors and prevents excessive slip by distributing brake force in situations such as sudden acceleration on a μ-split road surface. μ-split results in having different values for μ on different regions of the road, leading to vehicles going into high spin on the lower μ valued parts of the road. TCS, typically integrated with ABS minimize wheel slip by applying differential brake pressure as part of ABS by utilizing a limited-slip differential that distributes the torque between the two wheels of a driven axle to counter the slip on one of them, and finally by requesting the engine-ECU to reduce engine output and thereby reducing the torque on the driven axle. A limited-slip differential controls the power distribution between left and right wheels of a driven axle and thereby minimizes the effect of wheels spinning at different speeds. When both wheels are spinning at the same speed, both receive equal proportion of the axle torque τ_{in}. $\tau_{in} = \tau_{wheel1} + \tau_{wheel2}$ where $\tau_{wheel1} = \tau_{wheel2}$. Otherwise (suppose wheel2 is spinning excessively), then the limited-slip differential takes effect and modulates the torque distribution.

$$\tau_{in} = \tau_{wheel1} + \tau_{wheel2} \tag{5}$$

$$\tau_{wheel1} = \frac{\tau_{in}}{2} + \tau_{differential} \tag{6}$$

$$\tau_{wheel2} = \frac{\tau_{in}}{2} - \tau_{differential} \tag{7}$$

A torque bias ratio TBR determines how the value of $\tau_{differential}$ changes dynamically with respect to τ_{in}. Traction control system also request to reduce the engine output torque, where the amount of reduction depends on the vehicle's stability factor f and the engine characteristics such as maximum torque τ_{max}, torque while idling τ_{idle}, maximum rpm $\omega_{engine_{max}}$ [8].

Algorithm 1 The Control Algorithm

Definitions:

t , br : Throttle and Braking input in range [0,1], st : Steering input in range [-1,1]

L : Look ahead distance for speed guidance

$e_{v\_current}$,$e_{\theta\_current}$: Current error of speed controller, steering controller

vehicle state at time t :

m_t:$< x_t, y_t, v_t, a_y, a_x, \theta_t, \dot{\psi}_t, \ddot{\psi}_t, \dot{\psi}_{t_{desired}}, \lambda_{1:4}, \sigma_{1:4}, l_v, w_v >$

$BSMMap$ = $Map < i, List < m_i >>$ ▷ time ordered list of messages from each neighbor vehicle i

1: **procedure** Control Iteration
2: **while** $time \leq T_{end}$ **do** ▷ while current time less than simulation end
3: **Default actuation based on Steering and Speed PID controllers**
4: $V_{desired} \leftarrow V_{safe}$
5: $speedout \leftarrow speedPID(V_{desired\_controller}, V_{current,}, e_{v\_current})$
6: $steeringout \leftarrow steeringPID(path, e_{\theta\_current})$
7: **if** $speedout \geq 0$ **then**
8: $\big[t, br\big] \leftarrow \big[clamp(speedout, 0, 1), 0\big]$
9: **else** $\big[t, br\big] \leftarrow \big[0, clamp(speedout, 0, -1)\big]$
10: **end if**
11: $st \leftarrow clamp(steeringout, -1, 1)$ ▷ now default actuation inputs are computed
12: **Situational awareness update**
13: **for** $i = 1; i < n; i++$ **do**
14: sa_i = Get Neighbor Info((m_i,t_recent))
15: $ttc_{ego,i}$ QuarticTTC($state_{ego}, m_{i,t_{recent}}$)
16: **if** $ttc_{ego,i} < 2$ **then**
17: Evasive Action($state, sa_i$)
18: **end if**
19: **end for**
20: $br[1:4]$=stability correction($state, br$)
21: ▷ corrects braking torque at individual wheels
22: **end while**
23: **end procedure**

Algorithm 2 Evasion

1: **procedure** Evasive Planning Algorithm($state, sa_i$)
2: a_{max} =

$$\max_{v_t - dv \leq v \leq u_t + dv} a_x \ni (v, a_x) \in P$$

▷ retrieve maximum possible acceleration for current operating speed bin from precomputed Profile

3: **if** $sa_i = LK$ **then** ▷ lane keeping
4: **if** $a_{req} \leq a_{max_{vx}}$ **then**
5: $t, br \leftarrow [1, 0]$
6: **else**
7: $t, br \leftarrow [0, 1]$
8: **end if**
9: **end if**
10: **if** $sa_i = LC$ **then** ▷ lane changing
11: **if** $ego = lanechanging$ **then**
12: ego lane change abort
13: **else if** $a_{req} \leq a_{max_{vx}}$ **then**
14: $t, br \leftarrow [1, 0]$
15: **else**
16: $t, br \leftarrow [0, 1]$
17: **end if**
18: **end if**
19: **if** $sa_i = LOC$ **then** ▷ loss of control
20: $path_{other}$ = GetFuturePoints($m_{i,t}$,dt,n) ▷ project BSM state to get future path of other vehicle
21: **if** intersect($path, path_{other}$) **then**
22: $br \leftarrow 1$
23: **end if**
24: **end if**
25: **end procedure**

III. The Proposed Controller

Our objective is to show how to use surrounding vehicle's subsystem data in providing safer navigation in spite of unexpected maneuvers of surrounding vehicles. To this end, we created a control algorithm that utilizes our extended BSM messages containing such information by extending a path following controller.

Algorithm 1 shows the sequence of control action computations in each iteration. Lines 3-11 show the default computation of the PID based steering and speed controllers of the path following controller. The output at this stage are the triplet of st, t, br in the ranges $[-1, 1]$,$[0, 1]$ and $[0, 1]$ respectively. In order to dynamically react to other vehicle's BSM data, an ego vehicle maintains an accurate situational awareness of surrounding vehicles and makes decisions to take necessary evasive safety action. Lines 12-19 show the update of situational awareness based on received BSMs and the consequent planning of evasive actions is shown in Algorithm 2. This evasive ability is illustrated with scenarios in Sec. V. As shown in line 15, after receipt of each situational awareness update, the time to collision between ego vehicle and neighbor is recomputed and evasive action is planned if $ttc_{ego,other} < 2sec$. Time to Collision (TTC) is the lowest time in which a pair of vehicles will collide if they continue their current dynamic state (heading, velocity and acceleration). In our work, it is computed using the circular buffer method proposed by Hou et al. [9] that models every vehicle as a circular object with diameter being the vehicle's length. Using this approach, TTC can be computed by solving a quartic polynomial equation (Eq. 8) formed from the positions ($[x_1, y_1]$ and $[x_2, y_2]$), velocities ($[v_{x_1}, v_{y_1}]$ and $[v_{x_2}, v_{y_2}]$) and accelerations in lateral and longitudinal directions ($[a_{x_1}, a_{y_1}]$ and $[a_{x_2}, a_{y_2}]$).

$$\begin{aligned}
&[(x_1 + v_{x_1} \times TTC_{1-2} + \frac{a_{x_1} \times TTC_{1-2}^2}{2}) \\
&-(x_2 + v_{x_2} \times TTC_{1-2} + \frac{a_{x_2} \times TTC_{1-2}^2}{2})]^2 \\
&+[(y_1 + v_{y_1} \times TTC_{1-2} + \frac{a_{y_1} \times TTC_{1-2}^2}{2}) \\
&-(y_2 + v_{y_2} \times TTC_{1-2} + \frac{a_{y_2} \times TTC_{1-2}^2}{2})]^2 \\
&= (r_1 + r_2)^2 \qquad (8)
\end{aligned}$$

We filter the trivial low TTC values computed in case of vehicles safely maneuvering on adjacent lanes with impending overlapping buffers. The controller is aware of lanes by using the center line as reference and can compute lane change trajectories dynamically based on current vehicle speed, precomputed handling limits and the target point constraints based on received BSMs. To plan lane changes, it uses a pair of quintic polynomials [10] to plan the lateral and longitudinal offsets of the trajectory between current position and targeted position. A quintic polynomial trajectory planner allows for specifying constraints on velocities, accelerations and position for both initial and final vehicle configurations. After maneuver initiation, the lane change maneuver is tracked for progress and re-evaluated when necessary to accommodate dynamic constraints. With this approach, feasibility of a maneuver in progress is re-evaluated dynamically at a frequency of 10 Hz. This is how maneuver abort is achieved(Algorithm. 2, lines 11-12). For other evasive actions which are braking or accelerating in a host lane, the decisions are made based on precomputed stopping distances and maximum possible accelerations for each current speed. In line 2 of Algorithm 2, a_{max} refers to the maximum acceleration possible for current operating speed and is gathered from a precomputed profile P. Similarly, single lane change maneuvers with combinations of speed, longitudinal offsets that are precomputed for the ego vehicle model (Table III) in order to create lookup a table for ego vehicle limits, although these values may be computable dynamically. Lines 20-21 refer to the transformation of default uniform braking torque into differential torques by ABS and ESP as described in Sec. IV-B.

Algorithm 3 Steering Control Action

procedure STEERING PID($path$,$e_{\theta\_current}$)
Sentinel $\overline{S}$, Target $\overline{T}$ ▷ sentinel position , target position
Vehicle COG position $\overline{P}$ ▷ vehicle position
$\overline{e} = \overline{T} - \overline{S}$ ▷ error in global frame
$sign = \mathrm{sgn}\left(\{\{\overline{S}-\overline{V}\} \times \{\overline{T}-\overline{V}\}\} \cdot \langle 0,0,1\rangle \right)$ ▷ sign of error in vehicle frame, +1 or -1
$e_{\theta\_current} = sign \times |\overline{e}|$
$e_d = \frac{e}{\Delta t}$ ▷ compute derivative error
e_i +=$\frac{(e+e_i)\times\Delta t}{2}$ ▷ compute integral error
$e_{\theta\_current} = e$ ▷ update current error
$out = K_{P_\theta} \times e + K_{I_\theta} \times e_i + K_{D_\theta} \times e_d$ ▷ compute PID output
return out
end procedure

Algorithm 4 Speed Control Action

procedure SPEED PID(V_{target}, $V_{current}$, $e_{v\_current}$)
$e = v_{target} - v_{current}$ ▷ compute speed error
$e_d = \frac{e}{\Delta t}$ ▷ compute derivative error
e_i +=$\frac{(e+e_i)\times\Delta t}{2}$ ▷ compute integral error
$e_{v\_current} = e$ ▷ update current error
$out = K_{P_V} \times e + K_{I_V} \times e_i + K_{D_V} \times e_d$ ▷ compute PID output
return out
end procedure

IV. SIMULATION FRAMEWORK

A. The Vehicle Dynamics Framework

As mentioned, we simulated three lane changing scenarios to demonstrate the improved safety provided by our controller. In order to simulate the detailed vehicle dynamics at the level necessary for implementing subsystems, we extended an open source physics based simulator Chrono-Vehicle [11]. The Chrono library enables parameterizing static vehicle configurations such as mass, moments of inertia, length, width, maximum engine torque etc. We extended the *Generic vehicle* model to incorporate the safety related subsystems as described in Sec. IV-B. All vehicles in our experiments are front wheel driven. Algorithms 3 and 4 together form the default path following controller provided by Chrono Vehicle library [11]. Our subsystem implementation builds on it.

TABLE III: Vehicle model parameters

Parameter	Values
length and width l_v, w_v	3.42,1.54 m
weight W_v	1200 kg
moments of inertia M_x, M_y, M_z	222,944,1053 kgm^2
maximum engine torque τ_{max}	365 Nm
maximum braking torque τ_{max}	4000 Nm
tire cornering stiffness c_α	89250N/RAD
driveline	front-wheel drive

B. Implementing Vehicle Safety Control Subsystems

a) Traction Control System: We used a torque bias ratio of TBR=3:1 implying that the torque at the driven axle can be differentially distributed to both wheels and the distribution ratio is limited to 3:1. Also, the TCS-ECU can request a 20% output reduction to the Engine control ECU during a high slip. i.e ($\tau_{in} = \tau_{in} \times 0.8$).

b) Electronic Stability Control and Anti-lock braking System: Chrono's default braking model called simple brake model is implemented in such a way that the driver input in within the range [0,1] and acts as a multiplier for the supplied brake torque applied equally to all the wheels during the simulation time-step. In our implementation, we modified it to perform differential braking by supplying four separate coefficients $b_{fl}, b_{fr}, b_{rl}, b_{rr} \in [0,1]$ that are determined by the combined ABS and ESP system computations. The desired yaw rate in each step is computed as follows [12]:

$$\dot{\psi}_d = \frac{v_x \times \delta}{L + \frac{m a_x^2 (l_r - l_f)}{2 C_\alpha L}} \quad (9)$$

where v_x is current longitudinal speed, δ is steering angle, m is mass, C_α is cornering stiffness of the tires, L is distance between two axles, l_f is the distance between center of gravity and front axle, l_r is distance between center of gravity and rear axle. In an over-steer condition, $|\dot{\psi}_d| < |\dot{\psi}|$ where as in an under-steer condition $|\dot{\psi}_d| > |\dot{\psi}|$. In case of under steering, the inner wheels are braked whereas in case of over steering the outer wheels are braked to achieve the desired lateral weight transfer to maintain the yaw-rate closer to the desired yaw-rate. In our implementation, differential braking for yaw-rate control is achieved by using a value of 0.5 for the chosen inner or outer wheels. The required differential braking for slip control by ABS is achieved by using a value between 0 and 0.5 for the slipping wheels. For each wheel the two coefficients combine to determine the applied braking coefficients in the range [0,1]. Fig. 2 illustrates vehicle dynamic response without (left) and with (right) the safety control systems in action. We ran 2 simulations of the vehicle with a speed of 25m/s with and without the safety control systems enabled, attempting to traverse a so called ISO-double lane change [13] path - a standard course used in testing the stability of real vehicles. The yaw, yaw-rate and desired yaw-rate of the vehicle are plotted in the top section of Fig. 2. The middle and lower sections show the slip and slip angle responses at the 4 individual wheels of the vehicle. The figure shows that both the longitudinal slip values and slip angles are under control when the stability subsystem is enabled.

V. Evaluation

A. Usage Scenarios

We evaluated 3 scenarios with and without deploying our controller. Specifically, we illustrate how imminent collision can be avoided by taking evasive action based on the contextual information provided by the extended BSM. The extended BSM provides context by supplying λ_i , σ_i and ψ_d (Table II) that are used with other BSM attributes listed in Table I. We divided the behavioral motion of neighbor vehicle into 3 classes - lane keeping (LK), lane changing (LC) and loss of control (LOC). Each vehicle maintains a situational awareness map of it's neighbors, based on received BSMs. The default behavior of each vehicle is LK and it is revised after each new BSM is received as described in Algorithm 5. For each neighbor, car_i a value of TTC_{i-ego} is computed as per the method described in III. In a LC or LK scenario, if the value of TTC_{i-ego} is less than 2 seconds an evasive action is planned by the ego vehicle. In a LOC situation, additional computation is required to plan the evasive action based on the BSM as illustrated in Scenario 2. Each of the following scenarios is designed to run up to 10 seconds and fails if a collision occurs. The scenarios are crafted for a collision to occurs sooner. It is worth noting that in all figures, the contact positions of the 4 wheels are plotted for each car with front wheels in a lighter color to show vehicle orientation.

a) 2-lane, 2 car, aggressive double lane change scenario: In this scenario, two cars are travel in parallel unidirectional adjacent lanes. Car 1 (red) is initially at $(x_{cog1}, y_{cog1}) = (35, 0)$ and $v_{x_1} = 15m/s$ and Car 2 (green) is initially at $(x_{cog2}, y_{cog2}) = (0, 0)$ and $v_{x_2} = 21m/s$. In our scenario, Car 2 that is initially behind Car 1 initiates an

Algorithm 5 Determining neighbor state

Definitions:
Message at time t :
m_t:$< x_t, y_t, v_t, a_y, a_x, \theta_t, \dot{\psi}_t, \ddot{\psi}_t, \dot{\psi}_{t_{desired}}, \lambda_{1:4}, \sigma_{1:4}, l_v, w_v >$

function GetNeighborInfo(m_t,dt,n)

    if $\lambda_i > (\lambda_{threshold} = 0.2)$ **then**
        return LOC ▷ Mark as loss of control
    end if
    if ($|\dot{\psi}|, |\dot{\psi}_{desired}| \geq 5°/sec$) & ($a_y > a_{y-roadreference}$) **then**
        return LC ▷ Mark as lane changing
    end if
    return LK ▷ Mark as lane keeping by default
end function

aggressive double lane change maneuver to pass Car 1. In the first run without collision avoidance, a collision happens at time $t = 6.22sec$. With additional BSM broadcasts and proposed controller, Car 1 detects a second lane change at time $t = 5.302$ when it computes TTC_{2-1} = 1.2 secs. This triggers a braking action by Car 1 (Alg. 2, Lines: 10-18) until Car 2 completes lane change and $vx_2 - vx_1 > 0$. Fig. 3 shows the scenario run with and without cooperative collision avoidance.

b) 3-lane, multi car, μ-split road induced accident:: In this scenario three cars travel in parallel unidirectional adjacent lanes. Car 1 (red) is initially at $(x_{cog1}, y_{cog1}) = (35, 0)$ and $v_{x_1} = 25m/s$. Car 2 (green) is initially at $(x_{cog2}, y_{cog2}) = (0, 3.58)$ and $v_{x_2} = 20m/s$. Car 3 (blue) is initially at $(x_{cog3}, y_{cog3}) = (0, -3.58)$ and $v_{x_3} = 20m/s$. Car 1 encounters a μ-split from x=80 to x=100 (shaded gray in Fig. 4, first graphic). At this point, Car 1 starts skidding and transmits the BSM. A hard braking event causes the vehicle to skid. Cars 2 and 3 receive BSM data of the skid and project future points based on the most recent BSM. The extra BSM data $\lambda_i = -0.6, \psi_d = 0.4rad/sec, \sigma_{1,2} = (0.29, 0.32)rad$ indicates that Car 1 is not in control (Alg. 5) and that its skid path coincides with the planned path of Car 3. At time $t = 2.902$ sec, the value of TTC_{3-1} is $1.77sec < TTC_{threshold} = 2sec$ and the estimated trajectory of Car 1 from $BSM_{1,t=2.902}$ sec (Fig. 4 middle graphic, colored *red*) overlaps the sled-position estimation of Car 3. Upon this estimation (Alg. 6) Car 3 applies brake as an evasive action(Alg. 2, Lines 19-24) to avoid the collision. In Fig. 4, the middle graphic shows the actual trajectory and estimated future points from each of the BSMs sent by the skidding vehicle at the corresponding message time stamps for this example. The third graphic of Fig. 4 shows the resulting trajectories of the cars with the system enabled. Notice that Car 3 has come to a stop before reaching the skid path of Car 1.

c) Three-lane Multi-Car Simultaneous Two-car Lane Changes: In this scenario, three cars travel on 3 unidirectional adjacent lanes similar to Scenario 2. At time $t = 3$sec, a right lane change is initiated by Car 2 and a left lane change is initiated by Car 3. This causes a collision at time t=6.09 sec between cars 3 and 1 if collision avoidance constraints are not continually re-evaluated during the maneuver. With

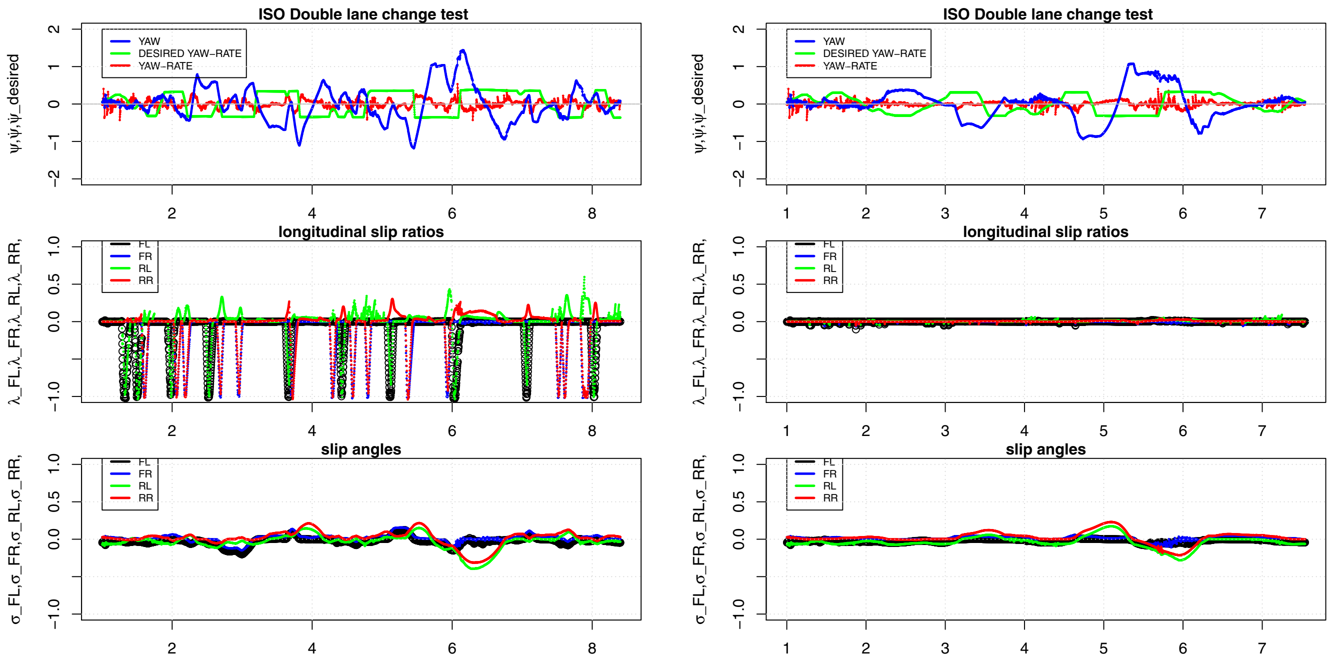

Fig. 2: Vehicle dynamic response while negotiating an ISO Double lane change maneuver with and without the stability system.

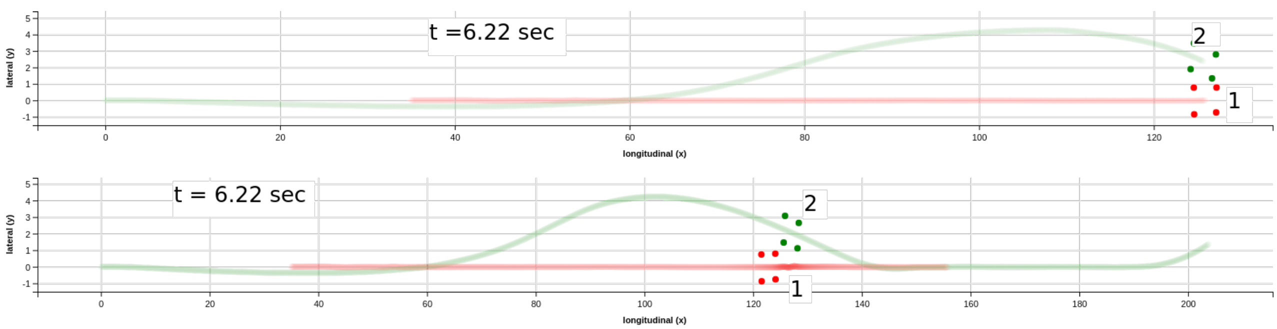

Fig. 3: Scenario 1

our controller, at the instant of imminent collision detection (t=5.2sec, $TTC_{3-2} < 2$), Car 2 and Car 3 simultaneously evaluate the stability and maneuver progress of the ego and other vehicle. The controller in Car 3 aborts the initiated lane change (Alg. 1,Lines 16-20 and Alg. 2,Lines 11-12) because of a built in the policy that the slower lane car has lower priority.

d) An Unsatisfiable Safety Constraint: As an extension to Scenario 3, we created Scenario 4 where the lead and lag vehicles in the three lanes as shown in Fig. 6. A total of 7 cars are initialized with positions and velocities as marked in the top graphic of Fig. 6. The center graphic shows the collision of Cars 4 and 5 with out the evasive action. The lower graphic shows the trajectories of cars with the system deployed. At $t = 3$ sec, Car 4 and Car 5 initiate a lane change into the center lane (lane 2). At $t = 3.2$ sec, an impending collision is detected at cars 4 and 5 and a lane change abort is triggered at car 5. However, soon after the LCM abort, Car 3's following distance constraint (Alg. 1, Lines 16-19) is violated for both pairs of (Car 3 & Car 5) , (Car 5 & Car 7). A braking action is triggered at both Car 5 and Car 7. This scenario exemplifies a situation where collision is unavoidable if either braking action at Car 3 (lead vehicle on lane) or acceleration at Car 7 occur. In the presented example Car 3 which is the leading vehicle of Car 5 brakes to slow down at $t = 3$ sec and Car 5 which rejoins lane 3 after aborting lane change does not have enough stopping distance leading to collision (lower graphic of Fig. 6).

B. Discussion

Safety controllers in a vehicle are the first to detect potential instabilities of individual vehicles. They use that information to initiate evasive maneuvers in order to maintain or regain stability of a vehicle. We showed how this information can also be used by neighboring vehicles to avoid accidents. We also showed a pseudo-code of a controller that can use this data to avoid accidents which would otherwise happen. We also showed the limits of our technique. How-

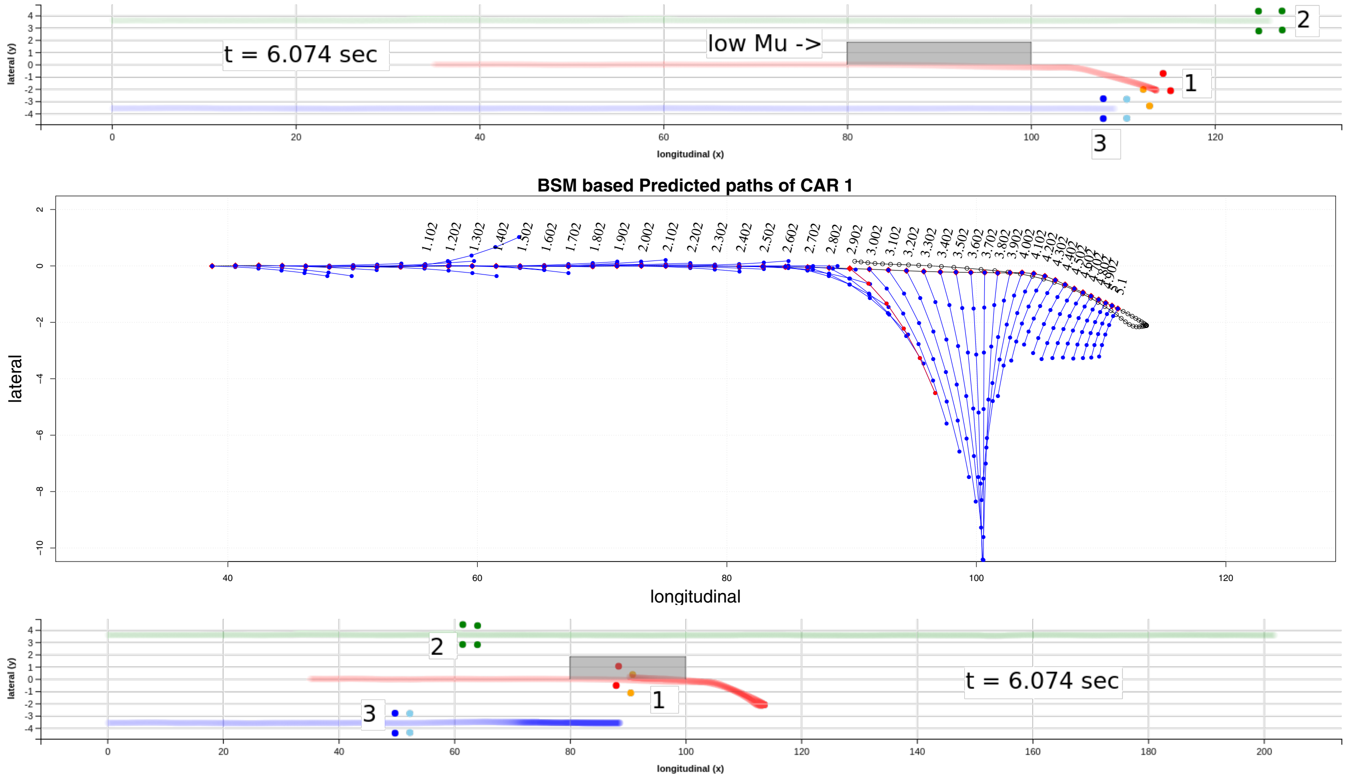

Fig. 4: Scenario 2

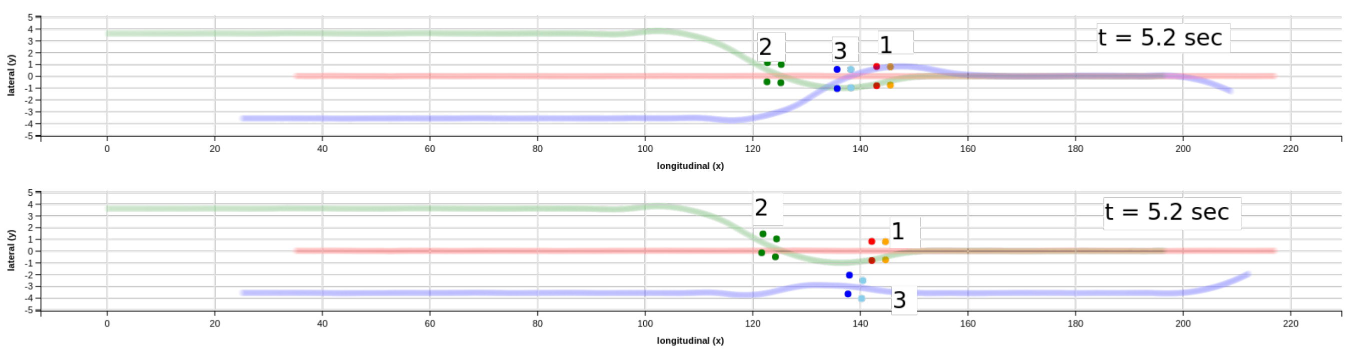

Fig. 5: Scenario 3

ever, this may also expose proprietary information of the control algorithms. For example, OEMs often do not release message codes and formats for ECU's communicating sensor and control data over the CAN Bus [14]. Sharing some of the sensor data as suggested in this work entails a possibility of discovering proprietary control algorithms as well as privacy risk for the operator. We showed how automated controllers can use DSRC messages in ensuring safety in different driving scenarios. We did so by developing a high fidelity vehicular dynamics based simulation framework in order to evaluate the utility of additional information.

VI. Conclusions

In this paper, we proposed to transmit low-level vehicle state information and modify control algorithms in recipient vehicles to use transmitted information for accident avoidance. We demonstrated the utility of our proposal by simulating practical lane changing scenarios. We also presented an abstraction of proprietary ESP-ECU data which the receiving vehicles can use to plan their maneuvers. This partly alleviates disclosing proprietary message codes that may result in privacy breaches.

References

[1] "U.s. dot advances deployment of connected vehicle technology to prevent hundreds of thousands of crashes — nhtsa," https://www.nhtsa.gov/press-releases/us-dot-advances-deployment-connected-vehicle-technology-prevent-hundreds-thousands, (Accessed on 01/03/2017).

[2] "Vehicle to vehicle communication technology fact sheet," https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/v2v_-fact_sheet_101414_v2a.pdf, (Accessed on 01/05/2018).

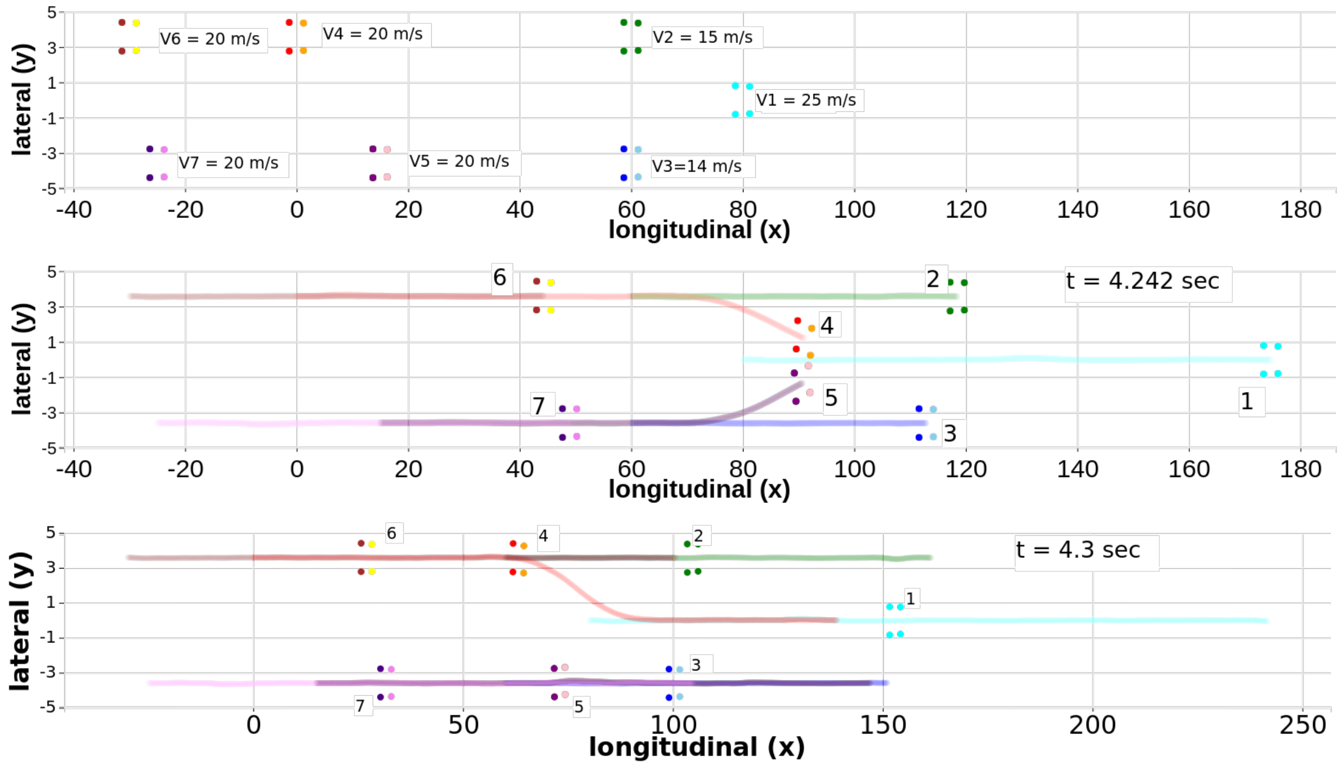

Fig. 6: Unavoidable collision: simultaneous driver events

Algorithm 6 Computing Future Points

Definitions:
Message at time t :
m_t:$< x_t, y_t, v_t, a_y, a_x, \theta_t, \dot{\psi}_t, \ddot{\psi}_t$,l_v,w_v$>$

function GetFuturePoints(m_t,dt,n)
  $FP_{t+dt} \leftarrow GetNext(m_t)$
  $Prev \leftarrow FP_{t+dt}$
  for $i = 1; i < n; i++$ **do**
    $FP_{t+i*dt} \leftarrow GetNext(Prev)$
    $Prev \leftarrow FP_{t+i*dt}$
  end for
  return FP ▷ Return computed set
end function
function GetNext(m_t,dt)
  $vx_{inertial} \leftarrow v_t * cos(\theta_t)$
  $vy_{inertial} \leftarrow v_t * sin(\theta_t)$
  $x_{t+dt_{pred}} \leftarrow x_t + vx_{inertial} * dt + .5 * ax_t * dt * dt$
  $y_{t+dt_{pred}} \leftarrow y_t + vy_{inertial} * dt + .5 * ay_t * dt * dt$
  $\theta_{t+dt_{pred}} \leftarrow \theta_t + \dot{\psi} * dt$
  $\dot{\psi}_{t+dt_{pred}} \leftarrow \dot{\psi}_t + \ddot{\psi} * dt$
  $m_{t+dt_{pred}} \leftarrow < x_{t+dt_{pred}}, y_{t+dt_{pred}}, v_{t+dt_{pred}}, ay_{t+dt_{pred}}, ax_{t+dt_{pred}}, \theta_{t+dt_{pred}}, \dot{\psi}_{t+dt_{pred}}, \ddot{\psi}_t, l_v, w_v >$
  return $m_{t+dt_{pred}}$ ▷ Return predicted next state
end function

[3] "J2735d: Dedicated short range communications (dsrc) message set dictionaryâĎć - sae international," http://standards.sae.org/j2735_201603, 3 2016, (Accessed on 12/27/2016).

[4] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enhancements of v2x communication in support of cooperative autonomous driving," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 64–70, 2015.

[5] J. Nilsson, M. BrÃďnnstrÃűm, E. Coelingh, and J. Fredriksson, "Lane change maneuvers for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1087–1096, May 2017.

[6] J. Lacombe, "Tire model for simulations of vehicle motion on high and low friction road surfaces," in *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, vol. 1, 2000, pp. 1025–1034 vol.1.

[7] R. Rudd, "Estimating the mu slip curve via extended kalman filtering," http://www.mathematica-journal.com/issue/v11i1/contents/MuSlipCurve/MuSlipCurve.pdf, (Accessed on 01/27/2018).

[8] A. Van zanten, "Control of horizontal vehicle motion," in *Road and off-road vehicle system dynamics handbook*. CRC Press, 2014, ch. 10.

[9] J. Hou, G. F. List, and X. Guo, "New algorithms for computing the time-to-collision in freeway traffic simulation models," *Computational intelligence and neuroscience*, vol. 2014, p. 57, 2014.

[10] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[11] "Project chrono - an open-source physics engine," https://projectchrono.org/, December 2016, (Accessed on 01/03/2017).

[12] *Electronic Stability Control*. Boston, MA: Springer US, 2006, pp. 221–256.

[13] ISO3888-1:1999, "Passenger cars–test track for a severe lane-change manoeuvre part 1: double lane-change," 1999.

[14] S. Freiberger, M. Albrecht, and J. Käufl, "Reverse engineering technologies for remanufacturing of automotive systems communicating via can bus," *Journal of Remanufacturing*, vol. 1, no. 1, p. 6, Dec 2011. [Online]. Available: https://doi.org/10.1186/2210-4690-1-6