Asynchronous Vehicle Control System Based on Integrated Driver Support Algorithm

Damian Petrecki
Department of Computer Science and Management
Wrocław University of Science and Technology
Wrocław, Poland
e-mail: damian.petrecki@pwr.edu.pl

***Abstract* — The paper describes a proposition of a driver support system composed of multiple independent processes producing discrete outputs and consuming continuous inputs, with a shared interpolate process. The research rejects multiple controllers handling different areas with the same actuators in favor of single but both multi-criteria and asynchronous decision-making system. This way, a decision-making problem has been limited, and a big data processing and control conflict hazard has been eliminated, keeping high vehicle performance, and lowering physical system complexity. This is an absolutely novel solution, very different than existing multi-domain in-vehicle controllers, due to new tasks division and processes synchronization approach. The results obtained during the simulation-based experiments show a very promising safety and comfortable ride.**

Keywords – integrated driver support system; multi-domain controller; continuous-time control; asynchronous algorithm, Integrated Driver Support Algorithm.

I. INTRODUCTION

A modern car can be controlled by a driver together with multiple control systems. Some of them just support the driver and some can take over entire control – at all times or in specific situations. It is getting increasingly critical for the automotive industry, customers and even governments and other lawmakers. It can be seen even in automotive marketing actions – new safety systems and drivers' assistance systems are boosted over new engines, better comfort, or practicality.

The drawback is that all these systems are created and implemented separately, often by different companies at different times, and simply not designed to work together. Therefore, a lot of devices are duplicated, multiple controllers with kilometers of cables are used, and control conflict hazard has to be maintained.

A proposed solution to this problem is to integrate all such systems into a single one, with one controller and a novel algorithm to rule all in-vehicle actuators.

The paper describes the algorithm – a new way to decompose the driver support problem, not into stability control, slipping wheels issue, extreme situation handling, etc., but into data acquisition, trajectory calculation, and control execution, which provides comparable results: a safe and comfortable ride. The algorithm has been called Integrated Driver Support Algorithm. The solution is literally a heuristic algorithm performing the vehicle control task, basing on driver's reference input and awareness of the surroundings, exclusively producing actuator signals for all actuators in the systems. This way, a vehicle equipped with the proposed solution can use a single, centralized computer system that eliminates multi-system interferences. Moreover, the vehicle can be easily maintained, including over-the-air software-based tuning, updating, and introducing new features without adding new physical sensors and controllers or modifying existing ones. What is essential, there is no ability to bypass the system by a driver, so it cannot be called a typical decision support system.

The paper is an extended version of [1], with a new, novel surrounding analysis method, and also supplemented by latest tests results, more descriptive explanations of previous ones and at last – a new, established algorithm name.

The paper structure is as follows. Section II shows a classic approach and currently popular research topics in the automotive area. The algorithm is described in Section III, which is followed by the presentation of the conducted simulation and its results in Section IV. The results of the simulations are evaluated in Section V. Comments on further work given in Section VI complete the paper.

II. STATE OF THE ART

It is hard to point-out a modern and adequately justified driver support system. There are well-known standard safety systems, like Anti-Lock Braking System (ABS) [2] or Electronic Stability Control (ESC) [3], but the mainstream is developed under non-public licenses or even as companies' secrets.

Nevertheless, we can observe modern vehicles' behavior and reach a conclusion how such systems work. Let us consider a case study, a widespread situation, well known from everyday driving – a driver wants to launch rapidly with front wheels turned, like when entering the flow of traffic. A modern car equipped with typical safety systems would involve a lot of these to influence the same parameter – wheels' speed. Engine Management System (EMS) [4] uses the engine to raise it, Acceleration Slip Regulation (ASR) [5] reduces it, active differential differentiates it, ESC applies brakes to avoid slipping, and ABS limits this brake action. This description omits other safety systems also able to use brakes when triggered, e.g., some surrounding aware collision prevention systems.

There is also a common issue for modern, existing solutions, briefly mentioned in the introduction – a hardware duplication. Modern vehicles are equipped with multiple sensors that measure the same parameters or areas, like a camera for traffic signs recognition, another one for lane assist system [6], another for pedestrian avoidance system, and another intended to control headlights – all directed ahead of the vehicle. With each sensor, there is separate wiring and, of course, a processing unit. This way, the complexity of the system raises, along with the cost, mass, failure probability and maintenance difficulty.

On the other hand, still, the most popular, related scientific topics are vision and perception [7][8], traffic models [9][10], or accident preventions [11][12]. Such papers are made to improve existing systems with better performance, lower cost, or extra features.

As it has to be mentioned, there is another, significant direction in the current automotive-related research – autonomous driving [13-15]. The aim of autonomous vehicles is to replace the driver by a surrounding and traffic-aware, intelligent algorithm or algorithms. To achieve it, a lot of different safety features were introduced – lane support assists, active cruise control, GPS and map-assisted localization mechanisms, traffic sign recognition system, pedestrian tracking system, etc. A lot of new papers are being produced around this issue and its different dimensions, problems, and perspectives. According to statistic research [16], drivers cause the vast majority of all accidents, so this approach is justified and has a lot of advantages. It is worth to mention that the solution proposed in this paper can be a potential, very convenient base for an autonomous driving system, but is not designed to be the base of it from the beginning.

There is also a new approach to create multi-domain controllers to handle several issues together. We can safely assume that modern stability control systems, even the companies-secret ones, are developed as single controllers that can control an engine and each brake separately, and realize ESC, ABS, ASR and another similar systems' functions within the same decision-making processes.

The next step is to integrate in-vehicle IT infrastructure and to create a single system to rule all actuators in a vehicle with knowledge coming from all sensor. One of the solutions which is under development now, is AUTOSAR [17]. It is a hardware-software solution intended to compose

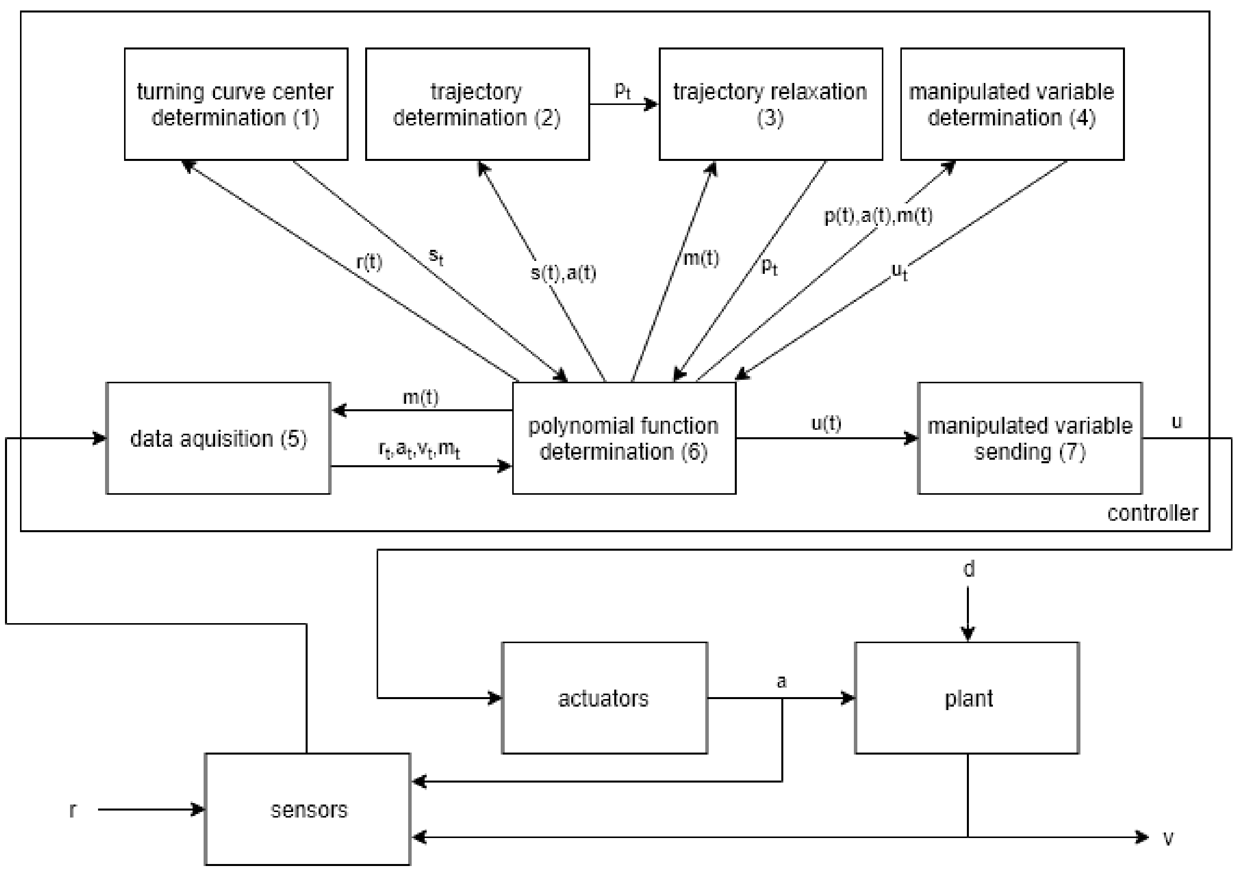

Figure 1. Control system schema

TABLE I. SYMBOLS USED IN FIGURE 1

symbol	description
r	reference input form an user
a	actuators state
d	external distortions
v	behavior measured via sensors (cameras, radars, accelerometers, etc.)
m	surrounding map – list of measured objects with its position and classification and parametrization results
u	control vector
s	center of turn
p	vehicle position p_t/trajectory $p(t)$

multiple control systems into a single one. This project notable expands the scope of a currently used, popular in-vehicle computer network, called CAN [18] or FlexRay [19], which is the only integration point now.

III. ALGORITHM GENERAL DESCRIPTION

This research presents a different perspective to driver support systems – it is an integrated system, and the main goal is not developing a better perception system, a more precise model, or a smarter autonomous driver-replacement, but presenting a new way to compose different, existing solutions to achieve high performance (in a way of vehicle safety and comfort) while lowering the computing power and system physical complexity at the same time.

The idea of algorithm is shown in Figure 1 with symbols explained in Table 1. When modeling using the black-box method and ignoring the controller's structure, the presented solution seems to be very similar to typical control systems. It reads reference input r from the user (mainly steering wheel and pedals positions), the vehicle behavior in its surrounding (using cameras and radars) and vehicle-related data (using accelerometers, thermometers, position sensors, etc.) v and also actuators state readers a and produces a control vector u consisting of all actuators manipulated variables: engine, linkage system, suspension, driveline, brakes. The only difference is the lack of time connection between inputs and the output.

When modelling using the white-box method, it can be seen that the algorithm refuses to use a popular algorithm chain architecture, based on iteratively producing new output data by calculating new input data with existing state data.

The algorithm consists of several processes instead. Each of them can be scheduled (triggered by time) or started by data incoming from a sensor. The outputs of all processes are discrete values (in one or more dimensions) shown as sequence elements with bottom index t, e.g., s_t. Input data (both starting processes and read during them) come from continuous-time functions stored in analytical, polynomial forms, shown as functions with t-argument, e.g., $s(t)$. It means a single, distinguished process (referred in Figure 1 as (6)) is introduced to build continuous-time functions from discrete sequences, which allows data interpolation and extrapolation. This way all data can be read between real measurements or calculations even after last ones without loosing accuracy. The process uses polynomial curve fitter method [20], accepts discrete values and timestamps, and produces a vector of polynomial coefficients. This way, a very specific storage is introduced that stores discrete variables and provides analytical functions as its output.

A single-dimensional data acquisition process is proposed (5). It reads and stores input values from input devices r_t and in-vehicle sensors, it reads actuators' states a_t, like suspension status, accelerations, engine status, etc., and simple (non-matrix) measurements from v_t. Each variable is handled by a separated thread.

The second part of process (5) handles the surrounding data v_t and is the most complex one. This is a complex part of the data acquisition process. This is the only case when the matrix data (distances of radars or bitmaps from cameras) must be handled. The process is triggered for each input from each signal separately. The result is a 3D model of the vehicle surrounding consisting of a set of classified objects, in the form of objects' shapes (3D line segments), class and positions, so data size is significantly decreased. Due to long processing time, the output of this process is stored with the input data appearance timestamps.

The surrounding analysis process is the biggest challenge related to the research. It needs a separate algorithm that accepts various formats of input data arriving at unpredictable time (cameras, radars) with, optionally, the already known 3D surrounding model $v(t)$ to update the model as its output. The Long Short-Term Memory (LSTM) [21] neural networks were considered as the most promising way to solve this problem, but it turned out that a very typical neural network used in some unusual form is a better solution.

Therefore, TensorFlow [22] platform is used to face this issue, but instead of using neural networks to classify objects, they are used to localize objects in the vehicle surrounding. Each network is trained to return a proper position of an object of a single class (like a specific model of vehicle, or a tree of a known shape) or zero when no object of this class is found. It means that each known class requires a separately trained network which results in a very long training process. The process was automated using scripts in the simulation environment during the research.

When the algorithm is being started, all known networks have to be run to analyze the vehicle surrounding, but after that, the existing surrounding model is used to limit the number of networks, using a list of already found or expected objects.

The process of the next type calculates the desired trajectory using the surrounding knowledge, vehicle-geometry model, and input data. It is split into several sub-processes, without any time-synchronization:

- The first sub-process referred to (1) calculates the center of the turning curve (if any) in the vehicle-centered coordinate system, using speed, steering wheel position, and vehicle geometry.
- Sub-process (2) calculates the desired vehicle positions p_t in the future, which means the desired vehicle trajectory.
- Sub-process (3) uses a genetic relaxation algorithm [23] and the surrounding knowledge v_t to improve the trajectory to avoid accidents, lowering external objects hit possibility. Please note this process can change the trajectory in any way, e.g., by decreasing speed or changing the turn, and its behavior is unpredictable. This is the only process that reads its input directly from the other process, not the storage.

The next process (4) uses the trajectory $p(t)$ to calculate control values for all executors u_t. For example, it calculates each wheel speed and turn, followed by the determination of engine power, braking force, linkage system, and differential parameters. This process bases on a mathematical model of the vehicle. It cannot be assumed that the model is utterly reliable and precise, but as it was realized during experiments, inaccuracy does not affect the evaluation of the algorithm. The model ignores the vehicle's body stiffness, uses Pacejka "magic formula" tire model and simple stiffness/dumping suspension model and it is still good enough for purposes of the algorithm. Moreover, using a more complex model, better reflecting a real vehicle behavior, impacts negatively on computation power usage but does not significantly improve overall algorithm evaluation results.

Calculated manipulated variables are being sent to the vehicle by own sender processes (7) (one process per variable), which read data from storage, not from the processes that actually generated them. Please note when using such algorithm architecture, there is no guarantee that output data is calculated using lastly collected input data. The delay can be relatively big (up to over a dozen measurements), and there is no mathematical proof that it does not affect the overall results. However, during the experiments, such a negative impact was not observed even during the worst-case scenarios, with rapid, unpredictable condition changes.

Please also note that processes (2) and (3) are shown in Figure 1 as two distinct parts, because each of them has its own interface and can be replaced by any other algorithm that fulfills it. On the other hand, it is a single process that uses the storage at its input and output and produces the trajectory $p(t)$ in two steps. In the same way, process (5) is shown as a single one because it has a single interface, but it effectively consists of two internal sub-processes to handle different formats of incoming data.

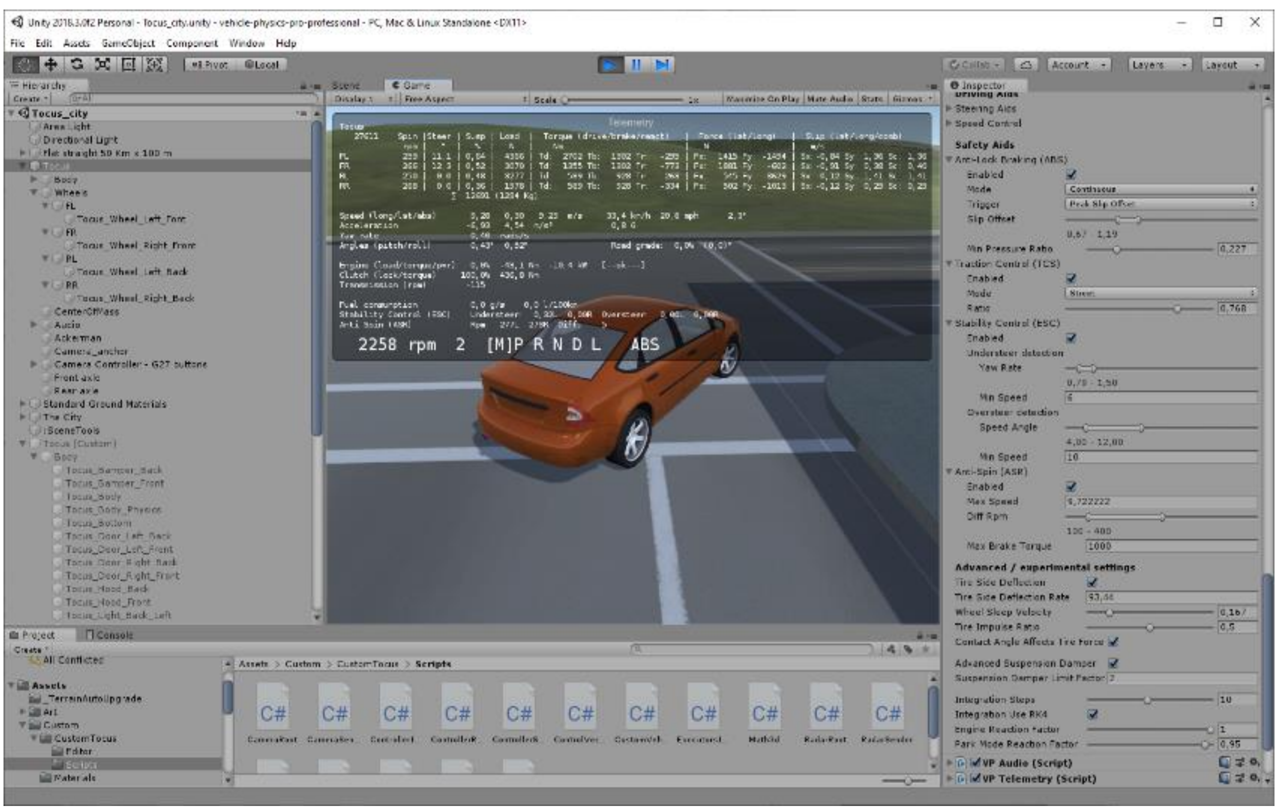

Figure 2. Simulator view (90-degree test)

TABLE II. MOOSE TEST RESULTS

enter speed	reference solution			proposed solution		
	A	*B*	*C*	*A*	*B*	*C*
80	1.6	-4	170	0.1	2	91
80	1.8	-7	172	0.1	-1	84
80	1.7	-8	168	0.1	0	83
80	1.6	-5	192	0.2	1	86
100	1.6	-6	99	0.1	2	100
100	1.6	-6	97	0.2	11	87
100	1.7	-10	78	0.1	-2	81
100	1.7	-6	89	0.1	0	82
120	1.8	-9	145	0.2	1	92
120	1.5	-9	150	0.2	4	91
120	1.7	-2	154	0.2	4	97
120	2	-4	140	0.2	-1	95
140	failed			0.1	2	110
140	1.6	-40	180	0.4	2	101
140	2.1	-36	165	0.2	4	105
140	1.9	-38	79	0.3	1	87
160	1.9	-48	145	0.2	-2	89
160	1.8	-60	138	0.3	1	90
160	2.2	-40	139	0.3	2	98
160	failed			0.3	2	115
180	failed			0.4	2	97
180	2.1	-68	165	0.3	2	95
180	failed			0.3	1	96
180	failed			0.2	1	97
200	2	-20	66	0.2	2	94
200	Failed			0.3	1	119
200	1.9	-30	81	0.4	-6	126
200	failed			0.3	-1	83

The most important idea behind the algorithm is the absolute lack of time synchronization between its input and output. Each process is being run separately and uses data extrapolated or interpolated from other processes, no matter the age of the source values or the last polynomial calculation time.

The second most important idea is to allow replacing processes with similar ones, that use the same interfaces. For example, this way the interpolation algorithm can be replaced by trigonometric one or TensorFlow can be replaced by some computer vision-based algorithm, without affecting the algorithm architecture, even with simple software upgrade of a vehicle.

TABLE III. 90-DEGREE TURN TEST RESULTS

enter speed	reference solution			proposed solution		
	A	*B*	*C*	*A*	*B*	*C*
10	0.2	2	253	0.2	-1	76
10	0.1	1	268	0.2	1	99
10	0.4	0	342	0.1	2	108
10	0.4	-2	268	0.2	0	104
20	0.8	1	372	0.2	-1	76
20	0.7	2	371	0.2	2	108
20	0.9	0	365	0.3	1	104
20	1	-1	312	0.2	0	98
30	1.2	2	290	0.2	2	96
30	1.2	1	246	0.2	1	94
30	1.1	-1	256	0.2	2	88
30	1.3	3	267	0.1	3	98
40	1.5	-10	160	0.2	-3	198
40	1.6	-8	381	0.2	-4	197
40	1.5	-12	271	0.3	-4	178
40	failed			0.4	3	174
50	1.9	-28	450	0.3	3	149
50	failed			0.2	-24	324
50	failed			0.3	-23	354
50	failed			failed		
60	failed			0.4	-38	450
60	failed			0.4	-39	450
60	failed			failed		
60	failed			0.4	-42	450

The third main idea is to not focus on well-known automotive-related issues, like preventing wheel slipping or speedway lane recognition, but to use computer science knowledge, a decision making algorithm and a robotic-like approach to monitor and improve a driver expertise and intuition.

It must be noted that all functions calculated by the algorithm can be used by external processes, not related to vehicle control, like headlight control, climate control, comfort features, etc., but it is not part of this research.

IV. CURRENT RESULTS

The presented solution has been tested in different scenarios, and the current results are presented.

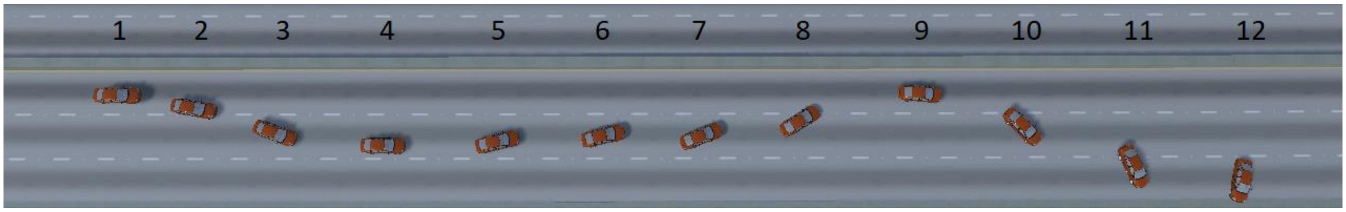

Figure 3. Moose test visualization for the reference vehicle and enter speed 140km/h (test failed)

A. *Simulation environment*

All experiments are conducted in Unity3D [24] environment with Vehicle Physics Pro (VPP) [25]. Unity3D is responsible for communication with an operating system driver of Logitech G29 steering wheel [26], rendering visual interpretation of simulated rides (shown in Figure 2 and Figure 3), and fundamental Newton physics. VPP is responsible for the vehicle simulation, including dependencies between in-vehicle physical subsystems, tires and suspension behavior, and standard active safety systems. VPP is also a source of the reference vehicle used, with all safety systems already included and configured. VPP is, in general, a pre-compiled library, so a lot of its mechanisms are unknown. Its realism is not verified, just considered to be sufficient to compare two vehicles in the same conditions. The validity of results obtained on the basis of a simplified environment for a corresponding real-world environment is the principal assumption of the research. The experiments' results are read from the telemetry panel provided by VPP (Figure 2) and from controller application. All data are stored during the tests in text files and analyzed offline.

B. *Test and reference vehicles*

The reference vehicle is built using VPP components only. It has an active suspension, 4-wheel steering, automatic gearbox, 4-wheel drive with active differential, and following active safety systems: ABS, Traction Control System, ESC, ASR. Most of its implementation is hidden and unknown but is calibrated using built-in configuration

TABLE IV. MOOSE WITH OBSTACLE TEST RESULTS

enter speed	reference solution			proposed solution		
	A	*B*	*C*	*A*	*B*	*C*
120	1.4	-4	145	0.2	0	87
120	1.5	-5	155	0.2	0	87
120	1.7	-6	149	0.3	4	89
120	1.9	-8	141	0.2	-2	89
140	1.6	-29	180	0.3	-1	98
140	2.0	-32	180	0.2	-3	99
140	failed			0.1	4	100
140	1.9	-32	81	0.2	5	89
160	2.0	-65	145	0.2	-1	110
160	1.9	-54	154	0.2	8	98
160	1.9	-53	163	0.3	4	95
160	failed			0.4	-4	110
180	failed			0.4	-5	101
180	2.2	-66	153	0.3	2	104
180	failed			0.3	-5	97
180	failed			0.3	-5	109
200	failed			0.2	-6	94
200	failed			0.5	6	111
200	failed			0.4	-6	132
200	failed			0.4	3	87

TABLE V. CRASH TEST RESULTS

enter speed	reference solution	proposed solution
60	avoided	avoided
60	avoided	avoided
60	avoided	avoided
60	avoided	avoided
80	wall	avoided
80	avoided	avoided
80	avoided	avoided
80	wall	avoided
100	avoided	avoided
100	wall	avoided
100	wall	avoided
100	following	avoided
120	wall + following	wall + following
120	wall + following	avoided
120	wall + following	avoided
120	following	avoided
140	other	other
140	other	other + following
140	other + following	other + following
160	other + following + wall	other
160	other + wall + following	other + following
160	wall + following	other
160	wall + following	other + wall

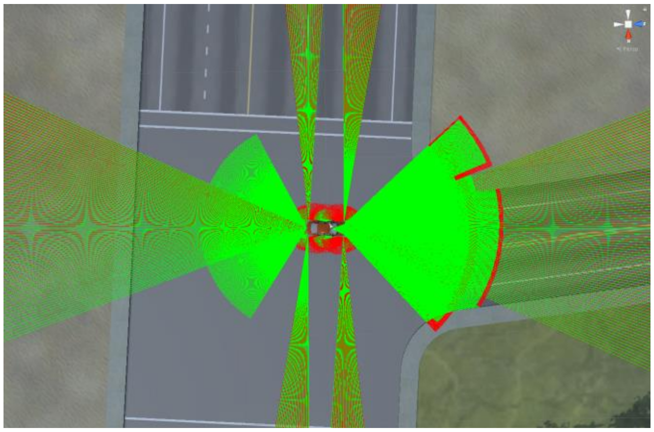

Figure 4. Lidar coverage (red lines mean no-hit rays)

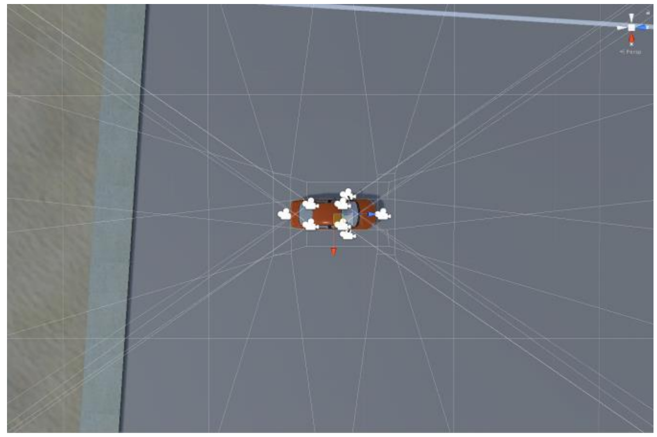

Figure 5. Camera coverage

panels, visible on the right part of Figure 2. Some systems, like active differential, were not available with VPP so have been implemented manually for this research.

The test vehicle has the same 3D model, physical parameters (weight 1200kg and equal weight distribution per wheel, wheels localizations, engine power/torque curves, etc.) and abilities (4-wheel drive, 4-wheel steering, controllable transmission, differential, and suspension). So, it is the same vehicle equipped with a different control system.

The difference is that in the test vehicle, the input from a driver is not sent to the vehicle itself but transferred to an external application implementing the proposed algorithm. All in-vehicle and surrounding-related sensors data are handled in the same way. The application sends back control variables for each actuator separately in separate threads. In this vehicle, there is no other driver support system implemented.

C. *Performance results*

All experiments are conducted using i7-7700k 4.2Ghz processor, 16GB RAM, SSD hard drive, and Windows 10 64bit operating system. Both simulation (Unity3D) and control (external application) are performed on the same machine because its performance is sufficient for current test scenarios. RAM usage never exceeds 10GB, and CPU load is always below 20% when simulation framerate 40fps is preserved.

The situation changes when all external sensors (21 LIDAR's and 8 cameras – see Figure 4 and Figure 5) are running. Then the simulation occupies about 4GB more RAM (which is still irrelevant), but exhausts all CPU abilities, reducing simulation framerate to 10-15fps (depending on a scenario). There are two ways to face this issue – to split simulation execution and control calculation to different computers using standard TCP/IP and UDP/IP network stacks or to disable several sensors simply. For the purpose of this paper, the second option is chosen, and all sensors directed backward are disabled.

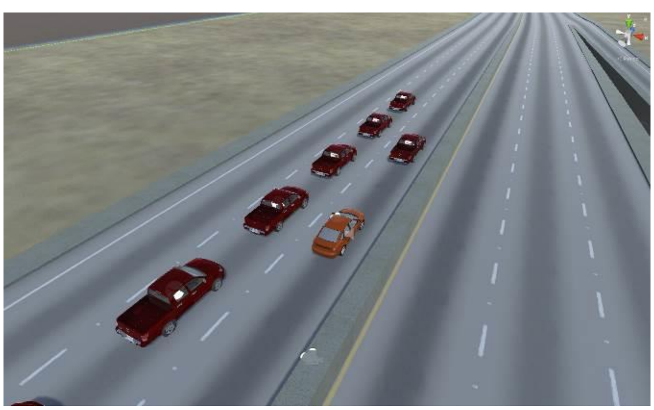

Figure 6. Experiment 4 scenario

D. *Dynamic experiments*

Five different experiments have been performed. All test rides have been conducted 4 times, each with the same driver and the same conditions. In Tests 1-3, vehicle roll angle A (degree), speed change B (km/h), and maximum steering wheel angle C (degree) have been evaluated. Lower roll angle means better comfort. Lower loss of speed means higher safety (shorter maneuver time) and also better efficiency (energy loss). Lower steering wheel rotation angle is considered as sportier and even safer behavior, allowing the driver to turn faster when holding the steering wheel with both hands all the time. During Test 4, both vehicles should avoid collisions and stay on the road, so the evaluation is descriptive. Possible values are: "avoided" (no crash), "wall" (crashed into a wall), "following" (crashed into the following vehicle), "other" (crashed into a vehicle on the other line) or a combination of them. In the last test, the most important evaluation parameter is minimum D and maximum E tires slip (m/s). Lower slip means better handling and safer ride. The test is passed when car fits the 4.3m lanes during the entire test.

1) *Moose test*

The test scenario requires a rapid change a lane and return to the original one on a straight road with velocity in range 80-200km/h (changing by 20km/h).

Results of the test are shown in Table II and Figure 3.

TABLE VI. LONG TURN TEST RESULTS

enter speed	reference solution		proposed solution	
	D	*E*	*D*	*E*
20	0.01	0.23	0.07	0.07
20	0.02	0.25	0.06	0.07
20	0.01	0.22	0.05	0.07
20	0.01	0.23	0.06	0.06
30	0.06	0.4	0.1	0.11
30	0.05	0.42	0.09	0.11
30	0.06	0.41	0.09	0.12
30	0.07	0.4	0.11	0.13
40	0.18	0.72	0.27	0.31
40	0.2	0.7	0.22	0.25
40	0.19	0.7	0.25	0.3
40	0.17	0.71	0.27	0.3
50	0.22	1.81	0.2	0.21
50	0.21	1.9	0.23	0.26
50	0.24	1.86	0.25	0.29
50	0.24	1.78	0.21	0.24
60	0.25	2.59	0.26	0.37
60	0.25	2.72	0.25	0.28
60	0.27	2.58	0.25	0.27
60	0.26	2.58	0.26	0,29
70	0.26	2.42	0.21	0.25
70	0.25	2.44	0.22	0.25
70	0.26	2.53	0.21	0.26
70	0.26	2.51	0.25	0.28

2) 90-degree turn

The test scenario assumes the turn right on a 90-degree intersection with velocity in range 10-60km/h (changing by 10km/h).

Results are shown in Table III.

3) Moose test with an obstacle

During this test, the vehicle should avoid a collision. The obstacle of a known class appears from the right side (like a vehicle coming in from a side road) when riding with velocity in range 120-200km/h (changing by 20km/h), on a two-lane road. No incoming traffic is taken into consideration. The aim of the test is to avoid a collision and return to the lane. The test is passed when a collision is avoided, and the vehicle fits in two 4.3m lanes during the entire test ride.

Experiment results are shown in Table IV.

4) Crash test

During the test, a vehicle rides behind another one on the right lane of a two-lane road. The left and right lane are crowded and walled, respectively (Figure 6). All vehicles start each test located in the middle of the proper lanes. The vehicle in front of the test one stops rapidly (with infinitive decelerating force and traction). At this moment, both vehicles (the test one and the one in front of it) are separated from each other by a distance value 2 times smaller than velocity, with unit conversion from velocity (km/h) to distance (m). It means 25m gap for 50km/h, 50m gap for 100km/h, etc. Tested velocity is in range 60-160km/h (changing by 20km/h).

The aim of the test is to avoid a collision or reduce an impact when possible. Results are given in Table V.

5) Long turn

Now a vehicle rides around a circle with a constant radius of 20m and speed in range 10-70km/h (changing by 10km/h). In this test, the capability of separately controlling all-wheel speed is shown.

Experiment results are shown in Table VI and Figure 7.

V. CONCLUSIONS

When analyzing results, some considerations arise. The reference vehicle has failed in 29% and the tested one in 4% of all 1st and 2nd tests' trials. This is the main proof of improved safety in the presented solution. Secondly, the maximum roll of the test vehicle is limited to less than 0.5 degrees no matter of conditions, for all trials. The reason is that the anti-roll bars work pro-actively, reacting to turn and speed, not to the roll itself. This behavior proves that the comfort of the ride improved.

The next thing to notice is that the maximum steering wheel rotation angle in the test vehicle is significantly lower and fits into 90 degrees for most cases. The cause is the steering wheel ratio being adjustable in an extensive range, due to the lack of physical connection (even simulated one). The function converting wheel angle and speed to the position of the center of a turn is adjusted to lower the minimum turn radius at high speed when rapid turns are impossible anyway due to vehicle momentum.

The next observation is that the presented vehicle does not slow down during most of the tests, except the ones, when preserving speed is impossible, due to high vehicle inertia. The cause is that the driver does not press the brakes, so all trajectories are calculated for the same speed. Stability is preserved with an active differential that transfers proper speed to all wheels to avoid a slip, with the stiff connection between wheels and engine and without using brakes. This way, a maneuver can be finished faster, and the engine is never stalled by brakes, which also improves safety.

On the other hand, the reference vehicle uses brakes to preserve comparable stability which causes a significant loss of speed.

As it is seen in Figure 3, vehicle position 6, dangerous loss of control over a vehicle can happen even when the vehicle is equipped with ABS and ESC. No such thing has happened during all experiments for the test vehicle, for any

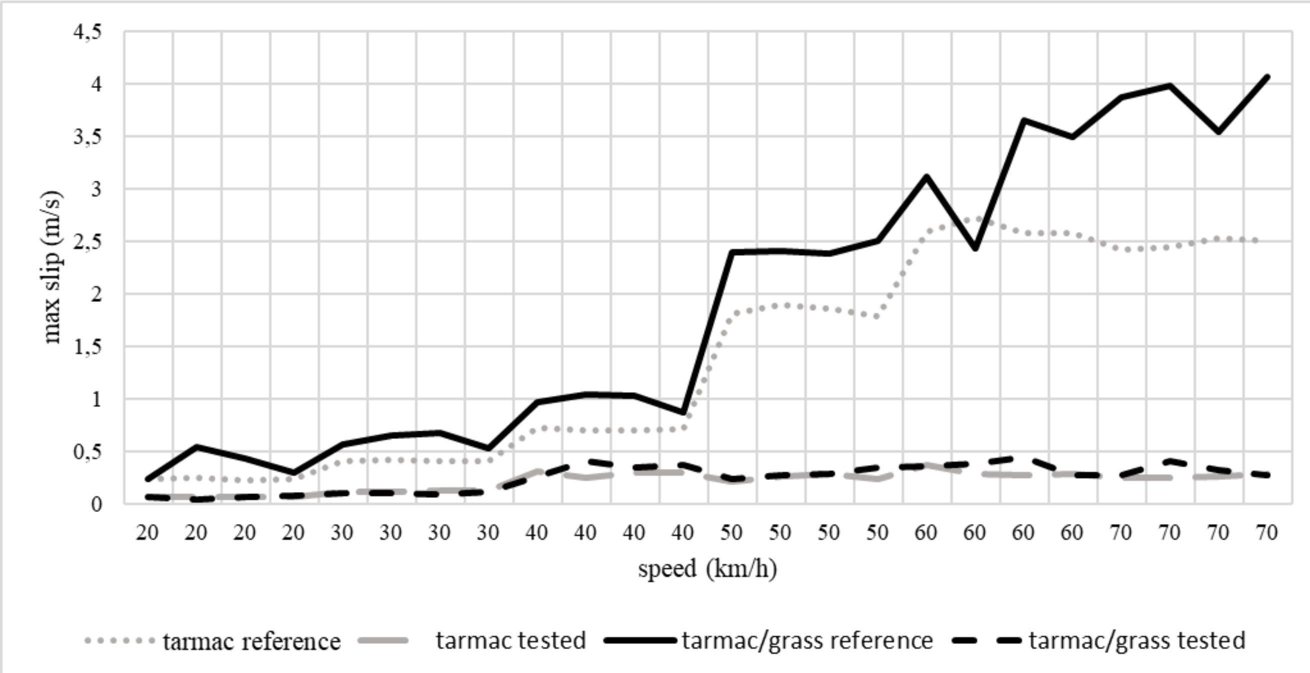

Figure 7. Slip test on different surfaces

scenario. It proves that without awareness of the surroundings, the reference vehicle follows driver's commands, even when the driver leads the vehicle to an accident.

Test 3 coincides, in general, with Test 1. It proves that awareness of the surroundings does not affect safety in a negative way. Neither hazardous nor dangerous situations were observed during the test vehicle's rides. What is important, the driver did not see any trajectory changes, even when it occurred. It proves seamless but safe trajectory changes calculated in the relaxation process.

Test 4 has delivered very interesting results. For all tests where a crash could be avoided, the test vehicle managed to avoid it. The driver reported that the car triggered braking before hitting the pedal or even without it. This behavior proves that awareness of the surroundings has been used to calculate and execute an accident-free trajectory. The action of the test vehicle after a first hit is the next interesting observation. In such situations, the entire algorithm is still working and trying to avoid another accident. The test has passed for the majority of repetitions. The mathematical model of the vehicle, that is not working well when vehicle inertia is disturbed after a hit, is a probable reason for the noticed failures of the test. A damage of a vehicle has not been comprised by the model. Almost all crashes up to 120km/h have been avoided by the test vehicle that is another advantage of the proposed algorithm. The vehicle tried to fit among the foregoing car and cars located on the second lane. The only failed result occurred when a driver has intentionally turned right during the trial.

On the other hand, the behavior of the reference vehicle has strictly depended on driver's actions during all rides. It has to be mentioned that the driver had also tried not to hit obstacles directly but to fit between them and in some tests (no. 16 and 18) had touched obstacles very gently even though high velocity. In similar situations, the test vehicle had crashed very hard each time when a collision was inevitable. This behavior is caused by a fallback mechanism implemented in the algorithm. When a collision probability is certain (trajectory relaxation cannot find any accident-free trajectory within a configured timeout), the unchanged path is returned from the process (3). It means that the vehicle follows the driver's trajectory without any changes and the driver may not be prepared for that.

In the last test, the minimum slip, occurring for inner wheels, is comparable for both vehicles, but active differential implementation offered by VPP is not as effective as the tested one. Moreover, in different conditions, when outer wheels travel on grass instead of tarmac, the differences between the reference and test vehicles are even bigger (Figure 3). The reason is, again, that tested standard safety systems react by breaking wheels already slipping, and the test one controls the behavior of the wheels proactively, calculating its speed before any slip occurs using mainly engine and differential, not brakes. This result also proves higher ride safety.

VI. FUTURE WORK

Although the current results are promising, a lot of work is planned. For now, all processes are triggered by data or time. The event-based trigger (rapid condition change) is planned. Besides that, disruption analysis with fuzzy functions [27] usage will be introduced, to replace all continuous-time functions with fuzzy equivalents to improve the overall evaluation. The full assessment will be conducted with more test cases and more drivers to find more edge-case scenarios to improve. And lastly, all processes implementations of the method use very simple algorithms so far, but they are designed to be replaceable, so the best combination is to be found. The first element to replace is the trajectory relaxation's objective function, which should minimalize possible accident impact instead of just avoiding it.

To make the evaluation of the algorithm fairer and more unprejudiced, a surrounding aware, automatic braking system has to be introduced to the reference car.

Future experiments will be conducted using two computers with a direct network link.

Physical experiments with real vehicles are not planned so far. Very sophisticated, highly equipped vehicle (with active differential, suspension, etc.) with an open-access available to all in-vehicle actuators and sensors and also a large set of extra sensors are needed, which makes such experiments too expensive for the current stage of research. This kind of research is possible after the full simulation evaluation.

References

[1] D. Petrecki, “Asynchronous Vehicle Control System Basing on Analytical Continuous-Time Functions”, in: The Eighth International Conference on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR 2019), Rome, Italy, June-July 2019.

[2] J. Ernst, “Mercedes-Benz and the invention of the anti-lock braking system: ABS, ready for production in 1978”, in: Daimler Communications, July 2008.

[3] A. van Zanten, "Bosch ESP Systems: 5 Years of Experience" in: SAE Technical Paper 2000-01-1633, January 2000.

[4] T. Denton, “Automobile Electrical and Electronic Systems”, Chapter 10, Routledge, July 2007.

[5] D. Hoffman, "The Corvette Acceleration Slip Regulation (ASR) Application with Preloaded Limited Slip Differential," SAE Technical Paper 920642, February 1992.

[6] E. D. Dickmanns, “Dynamic Vision for Perception and Control of Motion”, chapter 7, Springer, June 2007.

[7] M. Knorr, “Self-Calibration of Multi-Camera Systems for Vehicle Surround Sensing”, KIT Scientific Publishing, December 2018.

[8] B. Ranft and C. Stiller, “The role of machine vision for intelligent vehicles”, in: IEEE Transactions on Intellifent Vehicles, Vol. 1, pp. 8-19, March 2016.

[9] M. Z. Liu and M. Sun, “Application of Multidimensional Data Model in the Traffic Accident Data Warehouse”, in: Applied Mechanics and Materials, Volumes 548-549, pp. 1857-1861, April 2014.

[10] D. Badura, “Prediction of Urban Traffic Flow Based on Generative Neural Network Model”, in: Management Perspective for Transport Telematics. TST 2018. Communications in Computer and Information Science, volume 897, pp. 3-17, September 2018.

[11] Z. H. Ren, K. Zhang, L. X. Xue, and Y. L. Gao, “Mandatory Lane Change Model and Time Delay under Traffic Emergency Incidents”, in: Applied Mechanics and Materials, Vols. 644-650, pp. 2627-2631, September 2014.

[12] H. Abut, J. Hansen, G. Schmidt, K. Takeda, and H. Ko, “Vehicle Systems and Driver Modelling”, De Gruyter, September 2017.

[13] T. Nitsch, “Sensor Systems and Communication Technologies in Autonomous Driving”, GRIN Verlag, April 2017.

[14] S. Liu, L. Li, J. Tang, S. Wu, and J. Gaudiot, “Creating Autonomous Vehicle Systems”, Morgan & Claypool Publishers, October 2017.

[15] D. Vaishnavi, E. Sundari, T.V. Sangeetha, S. Shrinidhi, and P. Saravanan, “Design and Development of Computational Intelligence for Enhanced Adaptive Cruise Control Using Arduino”, in: Applied Mechanics and Materials, Vol. 852, pp. 782-787, September 2016.

[16] N. Djurkovic, “Car Accident Statistics in The U.S. – 2020 Update”, Carsurance, January 28, 2020, Accessed on: February 4, 2020. [Online] Available: https://carsurance.net/blog/car-accident-statistics/

[17] T. Scharnhorst and G. Reichart, “Progress on the AUTOSAR Adaptive Platform for Intelligent Vehicles”, in: The Eighth International Conference on Advances in Vehicular Systems, Technologies and Applications (VEHICULAR 2019), Rome, Italy, June-July 2019.

[18] L. Renjun, L. Chu, L. Feng. “A design for automotive CAN bus monitoring system “. in: 2008 IEEE Vehicle Power and Propulsion Conference, Harbin, September 3-5 2008, pp. 1-5.

[19] R. Makowitz and C. Temple, “FlexRay-a communication network for automotive control systems”, in: 2006 IEEE International Workshop on Factory Communication Systems, IEEE, 2006. pp. 207-212

[20] P. G. Guest, “Numerical Methods of Curve Fitting”, Cambridge University Press, December 2012.

[21] A. Yenter, “A Multi-kernel Convolutional Neural Network with LSTM for Sentimental Analysis”, ProQuest, 2017.

[22] https://www.tensorflow.org/, last retrieved, May 2020

[23] C. Cheng, C. Liu, and C. Liu, “Unit Commitment by Lagrangian Relaxation and Genetic Algorithms”, in: IEEE Transactions on Power Systems, Vol. 15, pp. 707-714, May 2000.

[24] https://unity3d.com/, last retrieved May 2020.

[25] https://vehiclephysics.com/, last retrieved May 2020.

[26] https://www.logitechg.com/pl-pl/products/driving/drivingforce-racing-wheel.html, last retrieved May 2020.

[27] J. J. Buckley and E. Eslami, “Fuzzy Functions” in: An Introduction to Fuzzy Logic and Fuzzy Sets. Advances in Soft Computing, Physica, 2007.