

Article

Simulated Sensor Based Strategies for Obstacle Avoidance Using Velocity Profiling for Autonomous Vehicle FURBOT

Khayyam Masood ^{*,†}, Rezia Molfino  and Matteo Zoppi 

PMAR Robotics, DIME, University of Genova, 16126 Genova, Italy; molfino@dimec.unige.it (R.M.); zoppi@dimec.unige.it (M.Z.)

* Correspondence: khayyam.masood@edu.unige.it; Tel.: +39-3205304054

† Current address: Università di Genova, PMAR Robotics Group, DIME—Dipartimento di Ingegneria Meccanica, Energetica, Gestionale e dei Trasporti, Via all'Opera Pia, 15/a, 16145 Genova, Italy.

Received: 21 April 2020; Accepted: 22 May 2020; Published: 26 May 2020



Abstract: Freight Urban Robotic Vehicle (FURBOT) is an autonomous vehicle designed to transport last mile freight to designated urban stations. It is a slow vehicle designed to tackle urban environment with complete autonomy. A slow vehicle may have slightly different strategies for avoiding obstacles. Unlike on a highway, it has to deal with pedestrians, traffic lights and slower vehicles while maintaining smoothness in its drive. To tackle obstacle avoidance for this vehicle, sensor feedback based strategies have been formulated for smooth drive and obstacle avoidance. A full mathematical model for the vehicle is formulated and simulated in MATLAB environment. The mathematical model uses velocity control for obstacle avoidance without steering control. The obstacle avoidance is attained through velocity control and strategies are formulated with velocity profiling. Innovative techniques are formulated in creating the simulated sensory feed-backs of the environment. Using these feed-backs, correct velocity profiling is autonomously created for giving velocity profile input to the velocity controller. Proximity measurements are assumed to be available for the vehicle in its given range of drive. Novelty is attained by manipulating velocity profile without prior knowledge of the environment. Four different type of obstacles are modeled for simulated environment of the vehicle. These obstacles are randomly placed in the path of the vehicle and autonomous velocity profiling is verified in simulated environment. The simulated results obtained show satisfactory velocity profiling for controller input. The current technique helps to tune the existing controller and in designing of a better velocity controller for the autonomous vehicle and bridges the gap between sensor feed-back and controller input. Moreover, accurate input profiling creates less strain on the system and brings smoothness in drive for an overall safer environment.

Keywords: velocity profiling; sensor feedback; autonomous drive; obstacle avoidance; hurdle detection; FURBOT

1. Introduction

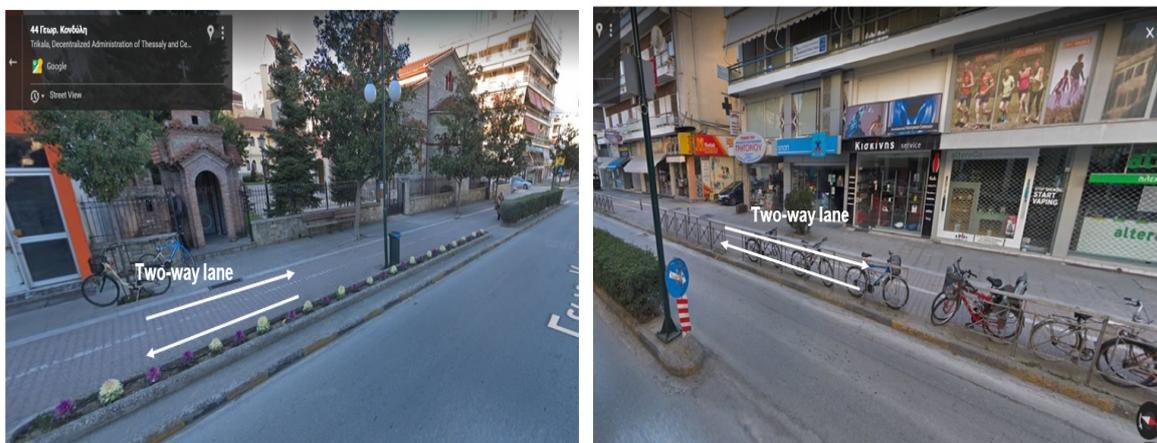
Work on autonomous vehicle and specially for obstacle avoidance is being studied in depth in recent times due to high demand of autonomous vehicles. Within literature, obstacle avoidance for autonomous vehicles mainly fall in the category of path planning and steering control [1–8]. Very few of the articles have mentioned obstacle avoidance taking into account the velocity adjustment to the obstacle [2,4,9]. Steering control for hurdle avoidance is a good technique specially for relatively high speed vehicles or for the vehicles which require to avoid obstacles on highway [1,3,6] but for slow moving vehicles, where there is usually only one designated lane, this approach is not viable.

From the research on autonomous vehicle it is clear that the controller choice for autonomous vehicles is model predictive controller or MPC [2,3,5–11]. This is because MPC has the ability to anticipate future events and can take control actions accordingly, unlike PID controllers which do not have this predictive ability. Many researchers used MPC for obstacle avoidance with steering control [2,3,5–8], however there is an exception in reference [4] where MPC is used to not only avoid obstacles through steering control, but also for control velocity of the vehicle in case of unexpected maneuver of the vehicle ahead. Although MPC is famous choice for many for control of autonomous vehicles [10]; many other types of control algorithms are also used for velocity and steering control. Among them sliding mode controller [12] and fuzzy logic controller [13,14] are highlighted. Similarly other control theories are also used for autonomous vehicles depending on choice and situation of user.

The outlook of the past decade in autonomous vehicles given in [10] also shows that it is difficult to seamlessly integrate motion prediction with vehicle control. Not only that, MPC has reported issues with steering control [10]. So the use of PID for our vehicle seems as a good enough option only if we can predict our motion computationally without the help of MPC. The study on autonomous vehicles in the last decade also shows that there has been very little work done on unexpected hurdles or tackling of usual urban obstacles, e.g., traffic lights.

Much of the autonomy of autonomous vehicles revolves around cruise control [1,3,15]. This is very essential for vehicles driving at high speed or at highway. Reference [6] used additional constraints for high-speed vehicles in the sense of toppling of the vehicle. The relatively less constrained autonomous vehicles, are not only dangerous for themselves, but are also dangerous for the environment. Another interesting work which is done recently is to maintain center-line i.e., center-line control [3,9,15]. This is another very key aspect for autonomous driving specially in cruise control. Lane discipline is one of the primary requirements for autonomous vehicles. Experimental obstacle avoidance is rarely found in literature for a full autonomous ground vehicle (AGV) however, in [16] obstacle avoidance with experimental results was achieved with AGV using steering control.

Freight Urban Robotic vehicle is an autonomous vehicle which is required to operate autonomously in urban environment for last mile freight delivery. It has to achieve this goal for the European H2020 project SHOW (SHared automation Operating models for Worldwide adoption) in 2021 in Trikala/Greece. In the demonstration, it needs to operate in bicycle lane with no lane change option available. Figure 1 shows the demonstration area for FURBOT in Trikala for SHOW project. This work is an attempt to achieve the goals of automation of FURBOT set by project SHOW. The velocity controller for FURBOT requires true classification of velocity profile in order for correct automation. Since the vehicle is required to perform in urban environment at low speeds, it is imperative that it will encounter hurdles within its path. Thus, the need for obstacle avoidance is generated.



(a) Demonstration area first angle

(b) Demonstration area second angle

Figure 1. Demonstration area for Freight Urban Robotic Vehicle (FURBOT) in Trikala/Greece (Source street view, google maps [17]).

Very few articles solve the issue of obstacle avoidance in urban setting. Reference [8] is one of the few studies which conducted obstacle avoidance in urban setting. However, only vehicle avoidance was observed. Other obstacles which vehicles may encounter were not observed in this study.

For slow moving vehicles, obstacle avoidance is usually limited to autonomous parking problem [18,19]. This is the usual setting where vehicle speeds are slow and speed control is also necessary in addition to steering control. However, this problem is very limited in path driven of the vehicle yet it has positive practical applications in autonomous drive. Using mathematical models for trajectory generation is also noted work in autonomous vehicles [11,16] yet is one of the initial steps for autonomous driving modules.

INRIA team RITS (Robotics and Intelligent Transportation Systems) have been working on autonomous vehicles for more than 15 years and is working on similar type of slow vehicles for obstacle avoidance. They have published on pedestrian detection [20,21] and pedestrian avoidance [22]. The approach used for pedestrian hurdle avoidance is the same which is used in this research i.e., slowing down of vehicle until the hurdle goes away from the path thus avoiding the hurdle. Their research is also a benchmark for this research. Velocity profiling for obstacle avoidance or navigation adjustment due to obstacles is still a novel concept and is being under study. Apart from INRIA's work in [22] for pedestrian obstacle avoidance; another use of velocity profiling for cruise control behind a slow moving object is shown in [23]. Velocity profiling is not just essential for avoiding obstacles, but it is also necessary to ensure smooth drive and drive comfort [24].

Our problem revolves around FURBOT which has to tackle obstacles and adjust itself while maintaining its lane. The best option to do so is not through steering control due to lane limitation, but through velocity control. Moreover, it has to tackle other type of hurdles within its path except moving vehicles e.g., traffic lights. Little work is found in literature where traffic signal is treated as an obstacle for autonomous vehicles. The usual work found on traffic lights is usually on its recognition system [25]. Obstacle avoidance for a slow moving autonomous ground vehicle which has its dedicated lane is not found in literature. Thus the need of our work is generated. This is emphasized for the vehicles which has to navigate in dense environment to be prepared for all types of obstacles, whether it is a slow vehicle, traffic signal or human obstruction; the vehicle in such urban environment needs to be able to adjust according to all such obstacles. In Addition to above, FURBOT has to manage obstacle avoidance by adjusting its speed as it has to operate in single lane. Thus, autonomous velocity profiling is very essential for its drive. Keeping lane restriction in mind and goal of avoiding obstacles, the best option is to cater with autonomous velocity profiling. This need sets the tone for technical requirement for our work that is carried out in this research.

2. Lateral and Longitudinal Equation of Motion

A summarized mathematical model that is used is presented using Newton-Euler equations of motion for forces along x-axis, y-axis and moment along z-axis and are given by Equations (1)–(3).

$$\dot{v}_x = \frac{\sum F_x}{M_v} = \frac{F_t + F_d + F_g + F_r}{M_v} \quad (1)$$

$$\dot{v}_y = \frac{1}{M_v v_x} (-a_1 C_{af} + a_2 C_{ar}) r - \frac{1}{M_v v_x} (C_{af} + C_{ar}) v_y + \frac{1}{M_v} C_{af} \delta - r v_x \quad (2)$$

$$\dot{r} = \frac{1}{I_z v_x} (-a_1^2 C_{af} - a_2^2 C_{ar}) r - \frac{1}{I_z v_x} (a_1 C_{af} - a_2 C_{ar}) v_y + \frac{1}{I_z} a_1 C_{af} \delta \quad (3)$$

In above equations; M_v is mass of vehicle, F_t is the traction force, F_d is drag, F_g is gravitational and F_r is the rolling friction force acting on the vehicle. Similarly, I_z is moment of inertia of vehicle and C_{af}/C_{ar} are the cornering stiffness of front/rear wheels respectively. δ is the steering angle and a_1/a_2 are the distances of the rear/front wheels from the CG of the vehicle. The detailed mathematical custom made model for FURBOT is previously published in [26]. This model is used as a platform for

modelling the obstacles in the same MATLAB/SimuLink environment and implementing the obstacle avoidance techniques presented in this research.

3. Obstacles Characteristics and Modelling

Obstacle characterization is very vital for autonomous control of the vehicle. Four obstacles of different nature are studied and modelled for velocity profiling for FURBOT. The mathematical modelling and approach to the obstacles for simulation are discussed in their respective section below. The four different nature of obstacles modelled are

- Traffic signal
- Human/Stationary object
- Speed breaker
- Slow vehicle interaction

3.1. Traffic Signal

Creating the mathematical signal for identification of traffic signal and the color change posed three main hurdles; Signalling out current color of the light, approaching yellow light after green or red light and finally approach velocity to the light wrt current traffic light color. For signalling out the current color of the traffic light, each color detection is equated with a numeric value termed as t_i ; 0 for green, 1 for yellow and 2 for red. With the help of three pulse generators, a desired output for traffic signal was generated. The signal attributes chosen were 60 s for red, 10 s for yellow and 40 s for green. The numeric signal out for the traffic light is shown in Figure 2.

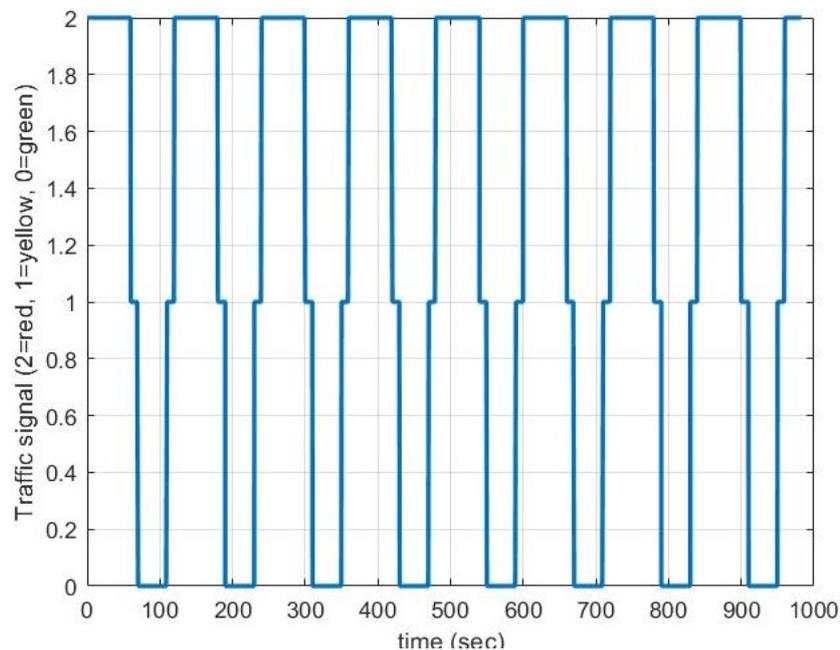


Figure 2. Traffic signal output.

An unfamiliar problem arose while modelling the vehicle velocity profile when approaching traffic light. The issue was how to approach yellow light when it is turning from red to yellow or from green to yellow. The idea is to stop before the signal in case light is turning yellow after green and to keep driving if it is turning from red to yellow or switch to cruise if the car is stopped at signal. An intuitive approach is used to tackle this issue; since numeric values were assigned to colors of signal, a derivative coupled with rate limiter of traffic light numeric value wrt time could define whether the signal is turning red or turning green after yellow signal. Using this ideology, an intuitive conditional equation for reference velocity $V_{ref_{sig}}$ as given by Equation (4) is formulated.

$$V_{ref_{sig}} = \begin{cases} v_{ref_{red}} & \text{if } \frac{d}{dt}t_{lr} + t_l - 1 \geq 0 \\ v_{ref_{green}} & \text{if } \frac{d}{dt}t_{lr} + t_l - 1 < 0 \end{cases} : t_{lr} = \begin{cases} R_t \Delta t + t_l(i - 1), & \text{if } \frac{d}{dt}t_{lr} \geq R_t \\ t_l, & \text{if } R_t > \frac{d}{dt}t_{lr} > -R_t \\ -R_t \Delta t + t_l(i - 1), & \text{if } -R_t \geq \frac{d}{dt}t_{lr} \end{cases} \quad (4)$$

where R_t is the rate limiter valued at 0.05 and $v_{ref_{red}}/v_{ref_{green}}$ are the reference velocities for red and green light respectively. Using this equation, we can stop the vehicle if the value is greater than 0 or keep cruising if it is less than 0 keeping in view the yellow traffic light signal condition. Figure 3 shows vehicle approach to the signal. The start of the signal corresponds to the signal capturing by sensors of the traffic light. Figure 3a reflects the sensor output of the traffic light detection whereas Figure 3b outputs the conditional derivative value of the signal sensor.

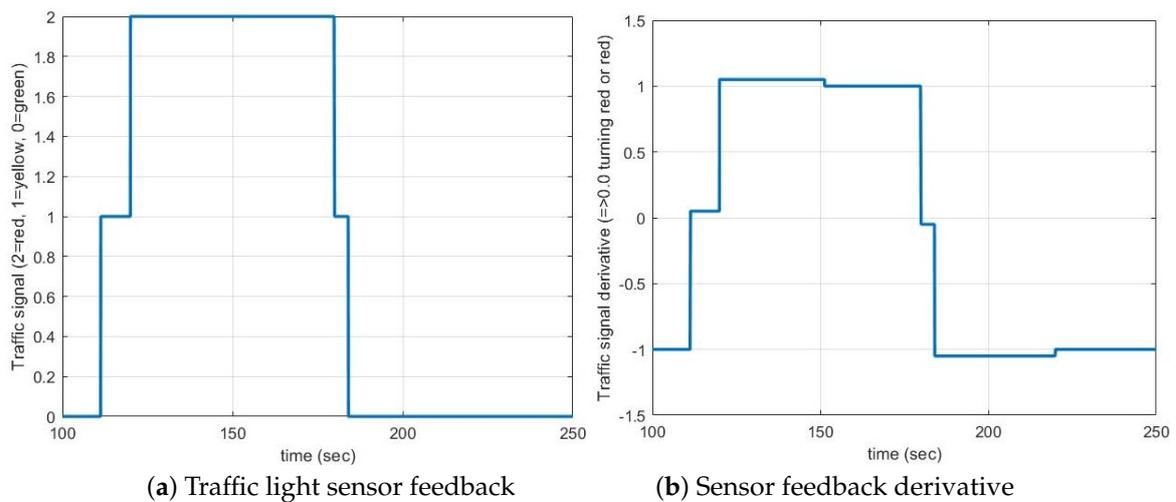


Figure 3. Traffic light sensor and derivative feedback.

The reference velocity profiling for traffic signal is kept as to steadily decrease the velocity of vehicle to zero on approach to the traffic light. This is achieved by considering current velocity of vehicle and distance to the light as given in Equation (5) where $V_{ref_{sig}}$ is reference velocity due to traffic light, d_{obs} is remaining distance to the obstacle, d_{band} is stop criteria for vehicle, d_{init} is initial detection of the obstacle and V_{ref} is desired reference velocity of the vehicle i.e., 20 km/h in our case. An example velocity profile using this formula is thus shown in Figure 4. The criteria for stopping the vehicle is to stop 2 m before the traffic light i.e., d_{band} . The same formula is used for velocity profiling in other obstacles as well except speed bump where simple switch between the velocity profiles is used.

$$V_{ref_{sig}} = \frac{d_{obs} - d_{band}}{d_{init}} \times V_{ref} \quad (5)$$

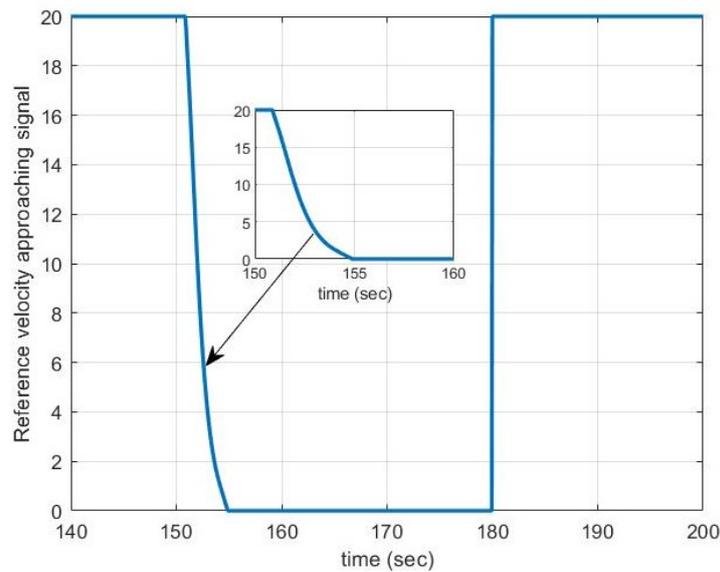


Figure 4. Traffic signal output.

3.2. Human/Stationary Object

Unlike in airborne vehicles, road vehicles can encounter sudden unknown obstacles e.g., humans or animals. These types of obstacles require sudden and immediate braking or if the object is farther away from the vehicle, a quick decrease in the overall velocity of the vehicle. This need is further explained in reference [22]. This aspect of driving phenomenon is taken into consideration in order to model such obstacles for better controlling the vehicle.

The stationary object/obstacle/human detection was randomly placed between 2~20 m ahead of the vehicle. The duration of the object in the navigation path was randomly fixed between 2~30 s. If the object moved away before the arrival of the vehicle, the vehicle continues with its reference velocity otherwise it adjusts its velocity to be zero 2 m ahead of the obstacle. This helps in gradually decreasing the velocity of the vehicle in case the object is away from the vehicle or to apply sharp and sudden brakes if it is in the vicinity of the obstacle. The reference velocity profile adjustment for a single such obstacle detection is shown in Figure 5.

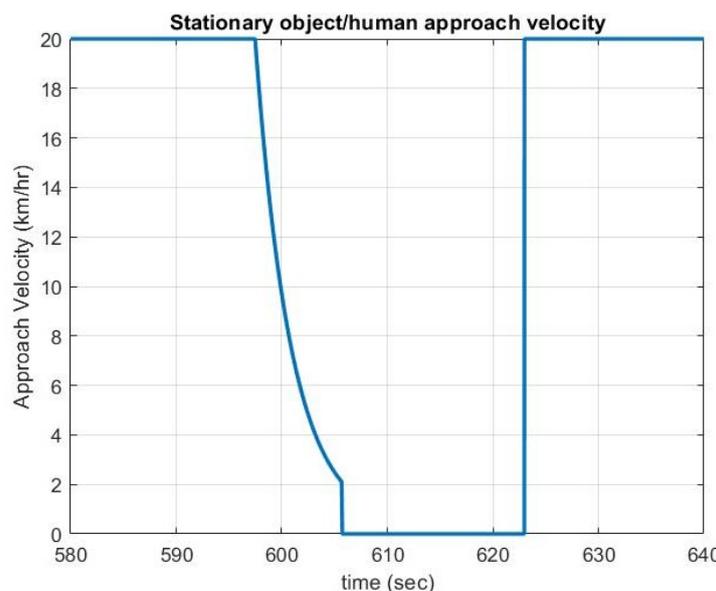


Figure 5. Velocity profiling for stationary object.

From observing the Figure 5 we see how the velocity is gradually adjusted to keep the object safe and to avoid a collision. During a single 5 km patch of simulation, five stationary objects with random times in-front of the vehicle are simulated. The placement of such hurdles was random in the 5 km patch. Additionally, the timing in-front of the vehicle of the obstacles and their distance immediate to the vehicle was also random between the limits mentioned above.

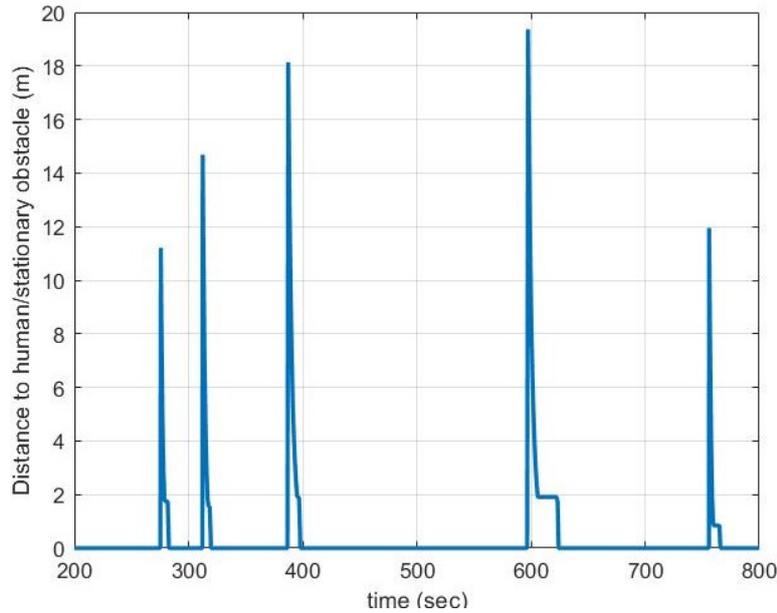
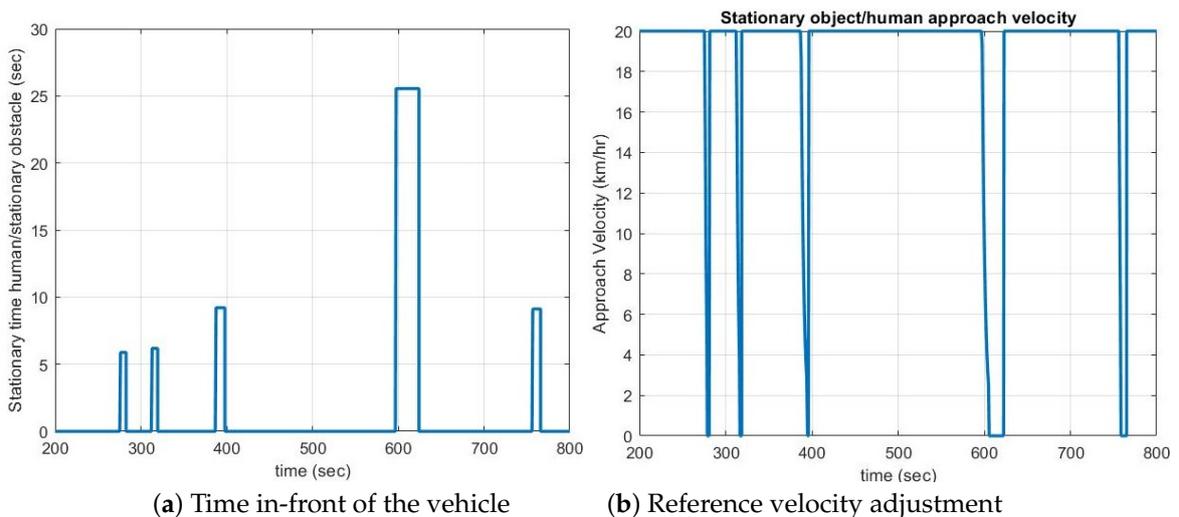


Figure 6. Vehicle distance to obstacle.

Figure 6 shows the detected object in-front of the vehicle. It is noted that the condition for stopping 2 m before the obstacle was not always met. In the last hurdle, the distance reduced to 1 m which was due to the fact that the car was on a slope to the approach. However, in all the cases, the vehicle stopped before colliding with the stationary hurdle. The approach velocity and the time of the obstacle in-front of the vehicle are plotted in Figure 7.



(a) Time in-front of the vehicle

(b) Reference velocity adjustment

Figure 7. Stationary obstacle collision avoidance.

3.3. Speed Breaker

The velocity adjustment in case of speed breaker obstacle was kept simple as the velocity drop is relatively small and there is no behavioral change in the obstacle. On detection of speed bump,

the velocity of FURBOT is adjusted to 10 km/h. The overall velocity adjustment on the detection of speed breaker is shown in Figure 8.

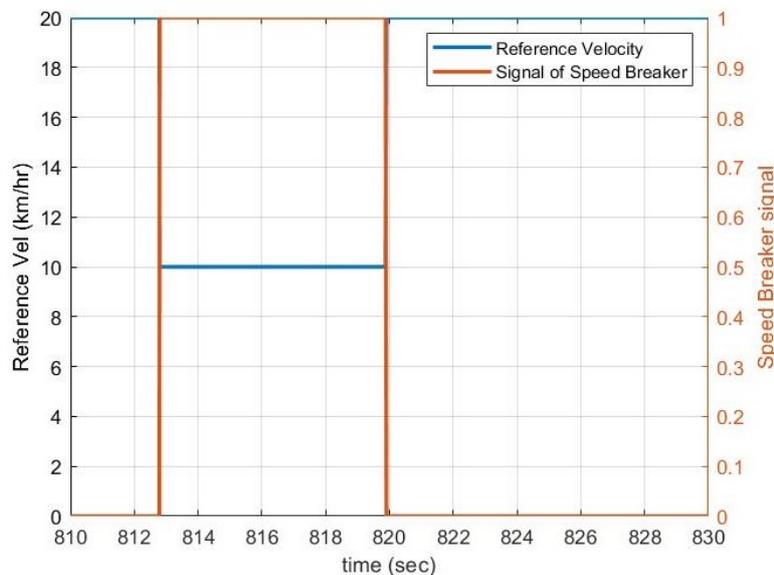


Figure 8. Speed breaker hurdle avoidance.

3.4. Slow Vehicle Interaction

One of the biggest hurdles/obstacles which are faced by driving vehicles is the encounter of slow vehicles and thus adjusting the velocity of the vehicle to that of the slow vehicle in-front. For this purpose, a slow vehicle is modelled in the path of the vehicle which comes randomly in its 5 km journey. The number of slow vehicles, their velocity profiles and any additional behaviors could be adjusted in the simulation. However, to test the controller, there is set to be a single slow vehicle which has a continuous velocity profile of 10 km/h. A random signal is generated for simulation purposes which shows that the vehicle has now moved away from the path of the vehicle. This signal is randomly generated between 5~60 s of encountering the vehicle. After the slow vehicle signal disappears, the vehicle continues with its default reference velocity profile.

Figure 9 shows the overall interaction of FURBOT with the slow vehicle. The velocity profile of the vehicle is gradually adjusted with time as to minimize the distance between the two vehicles and for not applying sudden brakes. The reference velocity adjustment takes the same approach of decreasing velocity gradually with the distance to the obstacle as was similar with the other obstacles. In Figure 9 the blue line represents the current position of the slow vehicle wrt time, the dotted red line shows the reference velocity of the FURBOT and the orange line shows the distance to the vehicle. The detection of slow vehicle is set to be 20 m away. We can see that the reference velocity starts to gradually decrease at 20 m markup and attains the velocity of the vehicle 2 m before it, thus avoiding the collision. As soon as the signal of the vehicle moving away from the path of the vehicle is received, the vehicle reference velocity is restored to its original value.

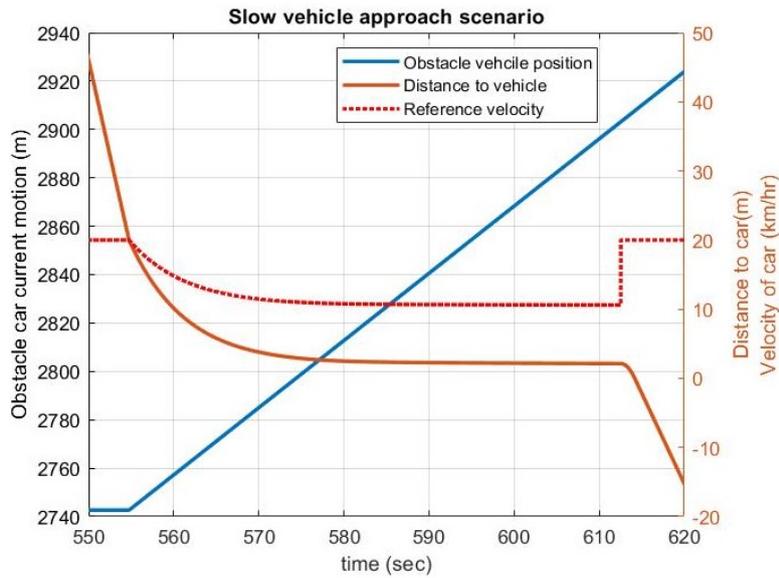


Figure 9. Slow vehicle hurdle avoidance.

4. Simulation Platform and Results

All the obstacles mentioned were modelled and simulated in a single run for a 5 km drive with varying road elevation profile chosen arbitrarily from an expected logistic path of the vehicle. The logistic path and the elevation profile chosen are available in reference [26]. The obstacles present in the simulation are given in Table 1. Furthermore, the whole obstacle avoidance technique is further explained in the flowchart in Figure 10. One additional condition from flowchart information is also placed which is given in Equation (6) in case of multiple obstacle detection.

$$V_{ref} = \min(V_{ref_{ped}}, V_{ref_{sig}}, V_{ref_{car}}, V_{ref_{bump}}) \tag{6}$$

where $V_{ref_{ped}}$ is the reference velocity generated due to pedestrian, $V_{ref_{car}}$ is reference velocity from moving car obstacle and $V_{ref_{bump}}$ is reference velocity from speed bump obstacle. The above condition is additionally placed in case multiple obstacles are detected in the presence of each other.

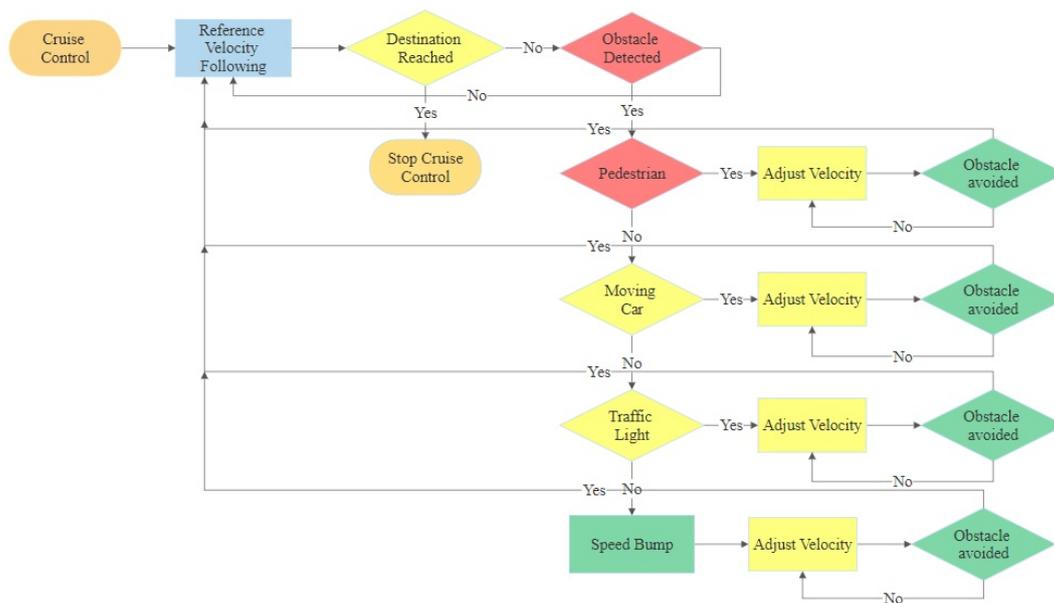


Figure 10. Flowchart of obstacle avoidance algorithm.

Table 1. Type of obstacles present in simulated drive.

Obstacle Type	Number of Obstacles	Time In-Front of Vehicle (s)	Distance In-Front of Vehicle (m)
Traffic light	1	Dependent on light	From 20
Stationary/Human	5	2~30	2~20
Speed Breaker	1	NA	From 20
Slow vehicle	1	5~60	From 20

In case two or more obstacles are encountered simultaneously within the path of the vehicle; the vehicle adjusts its reference velocity in accordance with Equation (6). Figure 11 is shown as an example where two obstacles i.e., moving car and stationary person coincides during the passage of vehicle. The vehicle first adjust its reference velocity wrt slow vehicle but when human obstacle is detected, the reference velocity is automatically adjusted to the lower reference velocity.

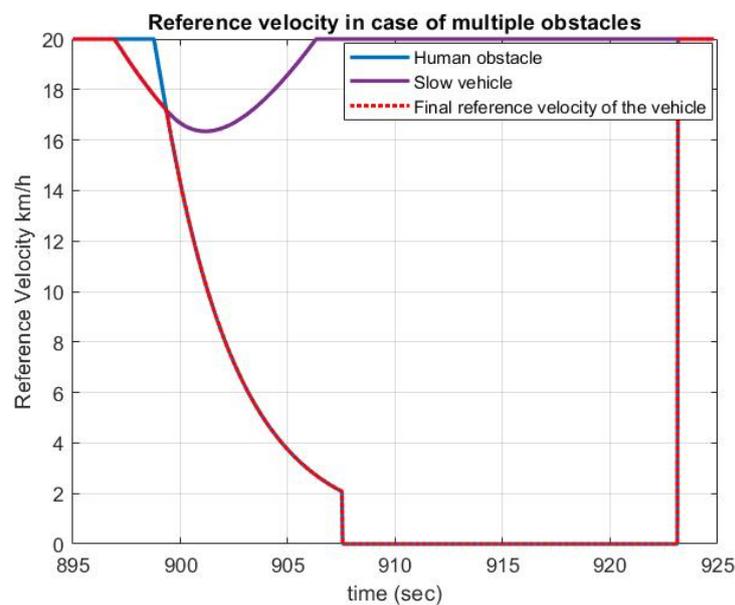


Figure 11. Reference velocity of the vehicle.

Two separate controllers are designed for velocity control of the vehicle; one for braking force and one for the traction force. This type of cascade controller is chosen deliberately as the requirements of acceleration and brakes are different for the vehicle. Additionally, when braking is applied, traction power is automatically shut down as in conventional vehicles. The idea of a low-level speed controller is taken from the idea introduced in Cycab [22] where the speed is regulated as a function of vehicle current speed and the target speed. The sampling time for the simulation is chosen as 1 ms and the reference velocity is updated at every sampling period. The controller for the vehicle had to be altered to adjust for obstacle avoidance. For braking control a PI controller was chosen whereas for traction force a PD controller was chosen. The traction force controller had an additional error amplification factor to adjust for power control. The overall controller configuration is given in Table 2.

Table 2. Velocity controllers.

Controller Type	Controller Values			
	Error Amplification	Proportional Gain	Integral Gain	Derivative Gain
Traction power control	500	20	0	1
Braking power control	1	550	6	0

The mathematical model for the vehicle was previously developed and behavior checked and verified in [26]. The reference velocity for the vehicle was set at 20 km/h unless the need to adjust reference to obstacles was required. The minimum reference velocity received either from the obstacles or in general is the final reference velocity of the vehicle in accordance with Equation (6). Figure 12 shows the reference velocity for the vehicle in the presence of aforementioned obstacles.

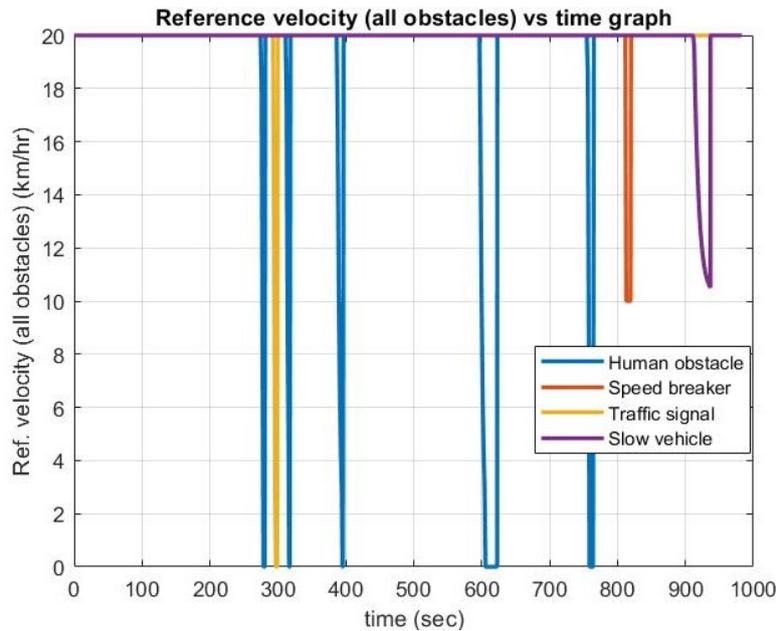


Figure 12. Reference velocity of the vehicle.

The braking control is only activated when there is positive difference between actual and reference velocities as shown in Figure 13. Moreover, the actual velocity of the vehicle and distance travelled by the vehicle are given in Figure 14. The bumps in the velocity profile when compared to reference velocity are due to the fact that there is a gradient profile of the road and the vehicle velocity changes due to weight shift of the vehicle. The simulation results displayed satisfactory acknowledgement of hurdles by the algorithm and the controller took correct preventive measures to avoid collision or adjusting to traffic light/speed breaker.

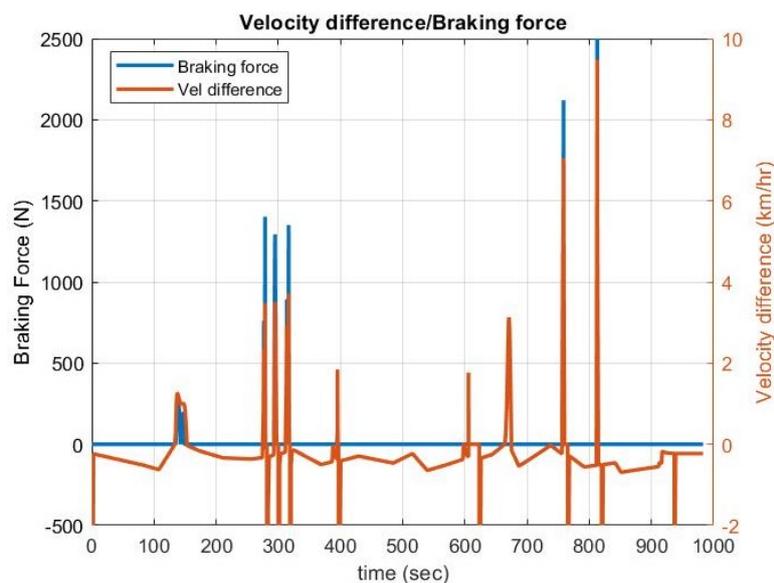


Figure 13. Braking force wrt velocity difference.

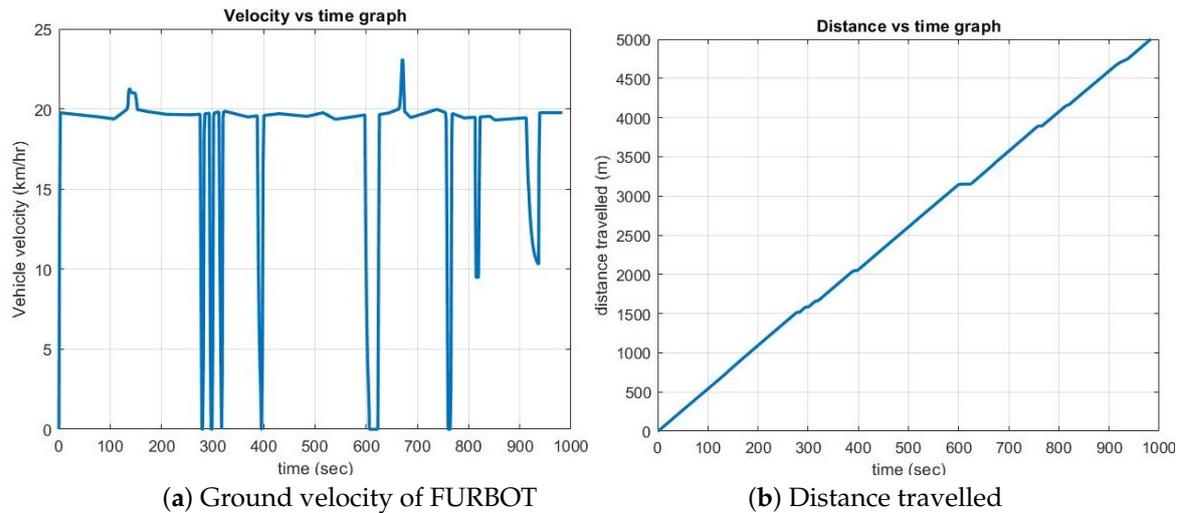


Figure 14. Stationary obstacle collision avoidance.

5. Conclusions

Obstacle avoidance for lane-restricted slow vehicles is studied and results are shared. Velocity profiling was formulated in this research and implemented in the simulated environment for the project FURBOT. The correct velocity profiling successfully helped in controlling the vehicle and avoiding the obstacles and collisions. The approach in itself is novel as it creates less strain on the controller with multiple inputs. All four hurdles are successfully modelled and avoided in random fashion within the course of drive for the vehicle FURBOT in the simulated environment. Additionally, this research will further help the autonomy of autonomous vehicles while they are commuting in a semi autonomous environment i.e., where manual and autonomous vehicles drive together. True velocity profiling in urban environment can also take away another driver based decision for manual cars. Furthermore, with this technique being implemented, we are one step closer to driving our autonomous vehicle FURBOT in urban environment setting for project SHOW.

Author Contributions: Conceptualization, K.M.; methodology, K.M.; software, K.M.; validation, K.M. and M.Z.; formal analysis, K.M.; investigation, K.M.; resources, K.M.; data curation, K.M.; writing—original draft preparation, K.M.; writing—review and editing, K.M. and M.Z.; visualization, K.M.; supervision, R.M. and M.Z.; project administration, M.Z.; funding acquisition, R.M. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This paper is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 875530, project SHOW (SHared automation Operating models for Worldwide adoption).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

FURBOT Freight Urban Robotic Vehicle
 SHOW SHared automation Operating models for Worldwide adoption

References

1. Brown, M.; Funke, J.; Erlien, S.; Gerdes, J.C. Safe driving envelopes for path tracking in autonomous vehicles. *Control Eng. Pract.* **2017**, *61*, 307–316. [[CrossRef](#)]
2. Ji, J.; Khajepour, A.; Melek, W.W.; Huang, Y. Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control with Multiconstraints. *IEEE Trans. Veh. Technol.* **2017**, *66*, 952–964. [[CrossRef](#)]
3. Rosolia, U.; Bruyne, S.D.; Alleyne, A.G. Autonomous Vehicle Control: A Nonconvex Approach for Obstacle Avoidance. *IEEE Trans. Control Syst. Technol.* **2017**, *25*, 469–484. [[CrossRef](#)]
4. Magdici, S.; Althoff, M. Fail-safe motion planning of autonomous vehicles. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016. [[CrossRef](#)]
5. Abbas, M.A.; Milman, R.; Eklund, J.M. Obstacle avoidance in real time with Nonlinear Model Predictive Control of autonomous vehicles. In Proceedings of the 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), Toronto, ON, Canada, 5–8 May 2014. [[CrossRef](#)]
6. Liu, J.; Jayakumar, P.; Stein, J.L.; Ersal, T. Combined Speed and Steering Control in High-Speed Autonomous Ground Vehicles for Obstacle Avoidance Using Model Predictive Control. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8746–8763. [[CrossRef](#)]
7. Febbo, H.; Liu, J.; Jayakumar, P.; Stein, J.L.; Ersal, T. Moving obstacle avoidance for large, high-speed autonomous ground vehicles. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017. [[CrossRef](#)]
8. Schwarting, W.; Alonso-Mora, J.; Paull, L.; Karaman, S.; Rus, D. Safe Nonlinear Trajectory Generation for Parallel Autonomy With a Dynamic Vehicle Model. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 2994–3008. [[CrossRef](#)]
9. Altche, F.; Polack, P.; de la Fortelle, L. High-speed trajectory planning for autonomous vehicles using a simple dynamic model. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017. [[CrossRef](#)]
10. Berntorp, K. Path planning and integrated collision avoidance for autonomous vehicles. In Proceedings of the 2017 American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017. [[CrossRef](#)]
11. Li, B.; Du, H.; Li, W.; Zhang, B. Integrated trajectory planning and control for obstacle avoidance manoeuvre using non-linear vehicle MP algorithm. *IET Intell. Transp. Syst.* **2019**, *13*, 385–397. [[CrossRef](#)]
12. Hu, C.; Wang, R.; Yan, F. Integral Sliding Mode-Based Composite Nonlinear Feedback Control for Path Following of Four-Wheel Independently Actuated Autonomous Vehicles. *IEEE Trans. Transp. Electrification* **2016**, *2*, 221–230. [[CrossRef](#)]
13. Guo, J.; Hu, P.; Wang, R. Nonlinear Coordinated Steering and Braking Control of Vision-Based Autonomous Vehicles in Emergency Obstacle Avoidance. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3230–3240. [[CrossRef](#)]
14. Benloucif, A.; Nguyen, A.-T.; Sentouh, C.; Popieul, J.-C. Cooperative Trajectory Planning for Haptic Shared Control Between Driver and Automation in Highway Driving. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9846–9857. [[CrossRef](#)]
15. Plessen, M.G.; Bernardini, D.; Esen, H.; Bemporad, A. Spatial-Based Predictive Control and Geometric Corridor Planning for Adaptive Cruise Control Coupled With Obstacle Avoidance. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 38–50. [[CrossRef](#)]
16. Laghmara, H.; Boudali, M.-T.; Laurain, T.; Ledy, J.; Orjuela, R.; Lauffenburger, J.-P.; Basset, M. Obstacle Avoidance, Path Planning and Control for Autonomous Vehicles. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019. [[CrossRef](#)]
17. Google. “Streetview”, Digital Images, Google Maps. Photograph of 30 Geor. Kondili, Trikala, Greece, Taken in 2019. 2020. Available online: <http://maps.google.com> (accessed on 17 April 2020).
18. Zhang, X.; Liniger, A.; Sakai, A.; Borrelli, F. Autonomous Parking Using Optimization-Based Collision Avoidance. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 14–19 December 2018. [[CrossRef](#)]
19. Sientop, C.; Laukart, V.; Krastev, B.; Kasper, D.; Wedel, A.; Breuel, G.; Stachniss, C. Autonomous Parking Using Previous Paths. In *Advanced Microsystems for Automotive Applications 2015; Lecture Notes in Mobility*; Springer: Cham, Switzerland, 2015; pp. 3–14. [[CrossRef](#)]

20. Pop, D. Detection of Pedestrian Actions Based on Deep Learning Approach. *Studia Universitatis Babeş-Bolyai Informatica* **2019**, *64*, 5–13. [[CrossRef](#)]
21. Pop, D.O.; Rogozan, A.; Nashashibi, F.; Benshair, A. Pedestrian Recognition using Cross-Modality Learning in Convolutional Neural Networks. *IEEE Intell. Transp. Syst. Mag.* **2020**. [[CrossRef](#)]
22. Flores, C.; Merdrignac, P.; Charette, R.D.; Navas, F.; Milanés, V.; Nashashibi, F. A Cooperative Car-Following/Emergency Braking System With Prediction-Based Pedestrian Avoidance Capabilities. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 1837–1846. [[CrossRef](#)]
23. Jacobsen, S.E.T.; Gustafsson, A.; Vu, N.; Madhusudhana, S.; Hamednia, A.; Sharma, N.K.; Murgovski, N. Predictive cruise control behind a stationary or slow moving object. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019. [[CrossRef](#)]
24. Du, Y.; Liu, C.; Li, Y. Velocity Control Strategies to Improve Automated Vehicle Driving Comfort. *IEEE Intell. Transp. Syst. Mag.* **2018**, *10*, 8–18. [[CrossRef](#)]
25. Eickeler, S.; Valdenegro, M.; Werner, T.; Kieninger, M. Future Computer Vision Algorithms for Traffic Sign Recognition Systems. In *Advanced Microsystems for Automotive Applications 2015*; Lecture Notes in Mobility; Springer: Cham, Switzerland, 2015; pp. 69–77. [[CrossRef](#)]
26. Masood, K.; Zoppi, M.; Molino, R. Mathematical Modelling for Performance Evaluation using Velocity Control for Semi-Autonomous Vehicle. In Proceedings of the 15th International Conference on Soft Computing Models in Industrial and Environmental Applications, Seville, Spain, 13–15 May 2020.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).