

A Novel Safe Merging Algorithm for Connected Vehicles Using NetLogo

Mehmet Bozuyula¹, Abdullah T. Tola¹, Yetis Sazi Murat²

¹*Department of Electrical and Electronics Engineering, Pamukkale University,
20070, Denizli, Turkey*

²*Department of Civil Engineering, Pamukkale University,
20070, Denizli, Turkey
attola@pau.edu.tr*

Abstract—In this paper, we present two new algorithms for MetroCar System (MCS) vehicles at a merging zone. First algorithm states the regulations of vehicles in the same road such as control the following vehicle's to prevent collision with the frontier vehicle. Second algorithm is for merging section of roads in order to prevent any collision at merging point. The proposed algorithms are verified and validated by using simulation results of the NetLogo simulation software. The simulation results show that the proposed algorithms guarantees that no collision occur at the merging zone.

Index Terms—Autonomous vehicles; Intelligent vehicles; Cruise control.

I. INTRODUCTION

Traffic congestion is one of the main problems in many countries. This problem causes driver distractions and tiredness. These adverse conditions can negatively affect safe driving. Although driver's reaction to various disturbances can cause blockage at intersections and merging section of highways, which is the main sources of traffic congestion [1], [2]. Connected and automated vehicles on merging zone make a contribution to improve safety and reduce traffic congestion. The conventional approach to solve this problem is ramp metering [3]–[5] and in this approach the inflow from the secondary road is limited by traffic light stop, so it is possible to prevent breakdown and to maximize the flow on the main road.

Studies about merge problem classified as centralized and decentralized approaches by Rios-Torres *et al.* [6]. Using the central approach, the traffic flow is controlled from a single center whereas using decentralized approach, all vehicles communicate with each other and some local coordinators to ensure road organization. Reservation method [7]–[10], intersection coordination [11], [12] and intelligent systems [13], [14] are subtitles of the central approach.

Since the decentralized approach is preferred in this paper, let us focus on this approach further. Some of researchers use the concept of a virtual vehicle e.g. Uno *et al.* suggested a merging control algorithm to avoid congestion for vehicles on a ramp [15]. Letter *et al.*

presented a longitudinal freeway merging control algorithm for maximizing the average travel speed of fully automated connected vehicles [16]. Lu *et al.* [17] developed a longitudinal control algorithm to securely merging vehicles based on wireless communication. Kamal *et al.* [18] used model predictive control (MPC) to solve a multi objective optimization problem for safety. Makarem *et al.* [19] used MPC to solve the decentralized problem where each vehicle defines its own constraints by using some valuable information from other vehicles. Milanec *et al.* [20] used fuzzy logic to design a control system that allows a fully automated vehicle to allow the vehicle of incoming in the conflict in zone or cross if collision risk is not present. Lu *et al.* [21] proposed an automated merging control method based on automated highway system (AHS) based on reference speed path for the merging zone vehicle.

A nonlinear optimization model is developed by Xie *et al.* [22] in order to improve traffic safety and increase traffic flow at freeway ramps. This model uses the second-by-second accelerations of all vehicles as the decision parameter and tries to maximize the total speed of all vehicles. It also ensures that when a vehicle arrives at the merging point, the distance between the vehicle and adjacent vehicles are greater than a minimum value to guarantee safety. Wang *et al.* [5] proposed an on-ramp merging control algorithm for Internet-connected vehicles. On the basis of vehicular operation characteristics during the merging process analysis, a cooperative driving algorithm based on Internet of vehicles was designed to achieve ramp merging without collision.

In recent years, many papers are published to solve merging control problem at multi-way intersections [23]–[31].

In this paper we will focus on both lateral and longitudinal control in decentral approaches by new algorithms. Unlike former studies that we have developed new algorithms will be implemented on MCS which is a recently proposed system to overcome traffic problems in cities. MCS combines the advantages of personal vehicles and the metro transportation system [32].

In this study, a novel merging algorithm is developed for MCS vehicles in order to increase the safety and to decrease the traffic congestion at merging zone. Moreover, it is guaranteed that no collision at merging zone. In this work,

we can control traffic flows with different vehicle speed whereas some researchers have assumed that speed of vehicles at merging zone are fixed [2]. Additionally, merge algorithm is developed for three different zones separately whereas only one algorithm and/or assuming one zone has proposed in most papers [5], [20], [33], [34]. The rest of this paper is organized as follows: The MetroCar System is explained in Section II. MCS infrastructure and MCS vehicles' attributes are given. In Section III, the vehicle tracking problem is discussed. We offer a new algorithm to solve this problem. The merging control algorithm is proposed to organize vehicles with various speeds in order to avoid a collision in Section IV. All proposed algorithms are tested using NetLogo simulation software. Some of the simulation results are given in Section V. Finally, concluding remarks are reported in Section VI.

II. METROCAR SYSTEM

MetroCar System (MCS) is a new transportation system that combines the metro, personal cars and automated cars. Although personal cars provide flexibility, hygiene and security, they are caused traffic congestion and time wasting. On the other hand, although there are some problems about flexibility for using metro, there is no traffic congestion. We suggest a flexible, hygienic and secure transport system, which is named MCS combining these systems' advantages.

MCS is consisted of two levels, personal drive level and automated drive level. In the first level, user has to drive his/her car from home/work to the MCS station at grade-separated road. Throughout this level the vehicle is user controlled. Automated drive level is started when driver request to admission to MCS. If main system allows, it will take the vehicle's control from driver and automatic driving is performed. Finally, the driver takes the vehicle control again after the vehicle leaves the MCS station.

MCS vehicles are small sized, one seater, operated by only electricity. They communicate each other with DSRC when control is taken by the main system. We set fixed all fundamental attributes for a MetroCar vehicle. In this way, we can control the traffic flow easily. Mechanical, physical and hardware attributes for all of vehicles are assumed to have some standards. Furthermore, speed control is controlled by fixed deceleration and acceleration. MCS vehicles are less costly than automated and personal vehicles because they use special grade-separated roads.



Fig. 1. A sample MCS infrastructure.

MCS offers a good solution for traffic problems; however, new infrastructure and grade-separated narrow roads have to be built for MCS. These roads must be isolated from pedestrians, animals and other external factors. MCS roads, which can be made with fixed intervals of lane change, fixed entry and exit points. Furthermore, there is no signaling system in these roads. An illustration is given in Fig. 1.

III. VEHICLE TRACKING

In this section, a new vehicle tracking algorithm is developed before we propose our novel merging algorithm. It is assumed that all connected vehicles have the same technical specifications, e.g. size, maximum speed, etc. All vehicles can communicate with each other if necessary; moreover, each vehicle propagates its position and speed information to others every τ time. Speed value of a vehicle can be changed by the system with two acceleration constants, positively or negatively in order to increase or decrease speed respectively. Deceleration will be used for negative acceleration from now on. Since a discrete time simulation platform is used in this work, let us define that each vehicle has speed and acceleration/deceleration rate as follows:

$$v[n] = \frac{x[n+1] - x[n]}{\tau}, \quad (1)$$

$$a = \frac{v[n+1] - v[n]}{\tau}, \quad (2)$$

where v is speed, x is the position, a is acceleration/deceleration rate, and τ is time step. Using (1) and (2), each vehicle's speed and position can easily be derived for every time. Assuming the vehicle only increase or decrease the speed, for a n^{th} time step, e.g. time is equal to $n\tau$, speed and position of a vehicle can be given as follows:

$$v[n] = v[0] + n.a.\tau, \quad (3)$$

$$x[n+1] = x[0] + \tau \cdot \sum_{k=0}^n v[k]. \quad (4)$$

Since the speed of a vehicle can be determined by (3) for each time step using only initial speed value, (4) can be rewritten in terms of initial and final speed values of a vehicle after some manipulation as following

$$x[n+1] = x[0] + \frac{v^2[n] - v^2[0]}{2 \cdot a} + \frac{\tau}{2} \cdot (v[n] + v[0]). \quad (5)$$

If we know initial and current speed values and initial position, we can determine the current position of a vehicle using (5). Let us define Δx as position difference for both difference of initial and final position of a vehicle or difference of current positions of two vehicles. For a single lane road, all vehicles have to update their speed for a safe tracking according to some rules. The key point for safe tracking, Δx between successive vehicles has to be greater than Δx_{\min} for all times. Δx_{\min} is defined as minimum following headway for successive vehicles so that if the leading vehicle makes its speed lower or it stops, the following one has to change its speed accordingly in order

to prevent any collision.

Figure 2 shows a vehicle tracking to another one by Δx_1 and Δx_2 respectively. Δx_1 is the difference between the initial positions of vehicles whereas Δx_2 is the difference between final positions of vehicles. If the leading one makes lower its speed therefore the following one also has to decrease its speed. The worst case of this scenario is the leading one's stop. In this case, Δx_2 has to be greater or equal to $\Delta x_{min} = \delta + x_{size}$ where x_{size} is fixed vehicle size and δ is minimal safe tracking distance. Assume that both vehicles stop for final case. Since vehicles speed of each one will be zero for the final condition, (5) can be rewritten for each vehicle as follows:

$$x_A[n+1] = x_A[0] + \frac{v_A^2[0]}{2 \times a_{dec}} + \frac{v_A[0]}{2/\tau}, \quad (6)$$

$$x_B[n+1] = x_B[0] + \frac{v_B^2[0]}{2 \times a_{dec}} + \frac{v_B[0]}{2/\tau}. \quad (7)$$

For a safe traffic flow, the next equation must be provided as explained above

$$x_A[n+1] + \Delta x_{min} \leq x_B[n+1]. \quad (8)$$

Therefore, the following conditions of (9) have to be ensured for all time for a safe tracking.

$$\frac{v_A^2[0] - v_B^2[0]}{2 \times a_{dec}} + \frac{v_A[0] - v_B[0]}{2/\tau} - x_B[0] + x_A[0] \geq \Delta x_{min}. \quad (9)$$

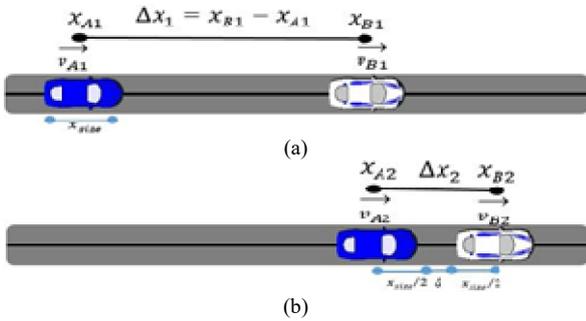


Fig. 2. Tracking vehicles by another one at an initial time (a) and at final time (b).

Algorithm. 1. Tracking Algorithm
 1. **begin**
 2. **for all of vehicles as A do**
 3. **find B** which nearest forward vehicle on same road
 4. **start** to communicate between A and B
 5. **if** position difference between A and B is less than Δx_{min}
 6. **decrease** speed of A
 7. **update** position of A
 8. **else if** speed of B is bigger than of A
 9. **increase** speed of A
 10. **update** position of A
 11. **else if** Equation (9) is false
 12. **decrease** speed of A
 13. **update** position of A
 14. **else**
 15. **increase** speed of A
 16. **update** position of A
 17. **end if**
 18. **end for**
 19. **finish**

In summary, we can control the safe vehicles tracking with the following algorithm according to (9). We can update each one of vehicle's speed with tracking algorithm.

IV. MERGING CONTROL

A novel vehicle merging tracking algorithm is proposed in this section. We consider a main and a secondary parallel road are merging; therefore, all vehicles at the secondary road have to merge to the main road vehicles safely in order to avoid any collisions. Secondary road is assumed long enough so that a vehicle can increase its speed at most maximum allowed value if necessary before merging point. For that reason, average speed values of both vehicles are very similar around the merging point. Vehicles at the secondary road keep on seeking for a gap to enter into the main road. The main problem is whether main road vehicle or secondary road vehicle has to accelerate or decelerate its speed in order to ensure a suitable gap between two vehicles. A sample road configuration is given as Fig. 3. As seen from the figure, there are three zones for merging roads; namely, self-zone, control zone and critical zone.



Fig. 3. A secondary road is merging to a main road.

At the self-zone, vehicles at both roads have to obey only tracking algorithm. It means, there is only communication among the vehicles at the same road.

On the other hand, possibly merging vehicles at main and secondary roads start to communicate with each other at the control zone in order to decide the priority. At this zone, secondary road vehicles are master whereas the main road vehicles are the slave since secondary road vehicles are assumed to have more priority than main road vehicles. A vehicle at the secondary road seeks a vehicle at the main road to be a companion before the critical zone. Assume that a vehicle at the secondary road has a position of x_s , i.e. Δx_s is left to the merging point. Assuming to keep its speed, it can be at the merging point at t_s time. This vehicle controls the main road vehicles if there is any vehicle at the position of $x_s \pm \Delta x$ which can be at the merging point at $t_s \pm \Delta t$. Here, Δt is the minimum time gap between two vehicles' passing time at merging point safely without any collision. If there is so vehicle, let assume that it has a position of x_M , i.e. Δx_M is left to the merging point. Assuming to keep its speed, it can be at the merging point at t_M time. If both conditions are assured, they are set to be the companion at this time step. Companion vehicles are lined up by increasing or decreasing their speed before the critical zone so that secondary vehicle can merge to the main road before or after the companion vehicle safely. Companion vehicles have to obey the following merging algorithm (Fig. 4).

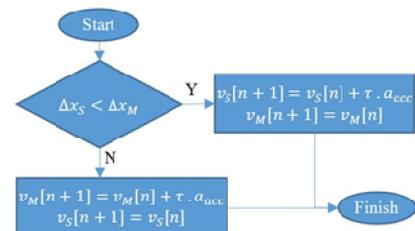


Fig. 4. Flow chart of merging algorithm.

Note that, all other vehicles at this zone have to obey the tracking algorithm.

Finally, at the critical zone, both roads are assumed as a virtual single road. Since companion vehicles are lined up before the critical zone, it is guaranteed that there is no collision at the merging point. They can be assumed as flowing on the same road. All vehicles have to follow only the tracking algorithm in this zone assuming virtually to be on the same road.

V. SIMULATION RESULTS

Proposed algorithm is tested using NetLogo which is an agent-based programming language and integrated modeling environment. A platform of Fig. 3 is designed. The length of the main and secondary road is set to 140 m and 120 m, respectively. The control and critical zone length are set to 60 m and 15 m respectively. Acceleration rate is considered as 2,5 m/s² whereas deceleration rate is 10 m/s². Minimal tracking distance δ is 2 m and size of each vehicle is regarded as 2 m.

Headway of vehicles entering the secondary road is considered to be fit to the normal distribution as given (11)

$$f(x) = \frac{e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}{\sigma\sqrt{2\pi}}, \quad (11)$$

where $-\infty < x, \mu < \infty, \sigma^2 > 0$, μ is the mean and σ is the standard deviation.

The developed tracking algorithm and the merging algorithm is run at NetLogo simulation program. The algorithm is tested with different μ and σ to verify the safety of the proposed algorithm. In this paper, μ is set to 3000 and σ is set to 1000 and simulation tests are performed. For example, leading vehicle forced to be stopped, hopefully the following vehicle is also able to stop safely. It is observed that the following vehicle may adjust (increase or decrease) its' speed to keep safe tracking distance with leading vehicle. Speed and position graphics for this simulation are depicted in Fig. 5. As you seen from the figure, the rear vehicle increased its speed until to point where it can stop safely then it decreased its speed. Finally, it stopped to keep safe margin between its position and the front car. All of the simulation results shows that our novel algorithm is definitely safe and there has never been happened any collision.

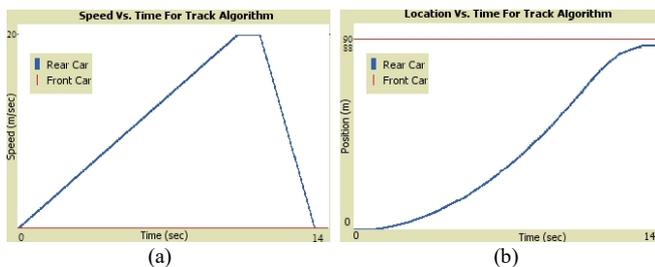


Fig. 5. Tracking algorithm verification by: a) speed and; b) position.

The merging algorithm has also been tested for different situations. As shown in Fig. 6, vehicles which have been flowing on secondary and main roads are started to communicate with each other. The vehicle on the secondary road can be easily merged to the main road at the final

position. Additionally, this figure shows that distance between vehicles on the main road is increased by the time to create a safe gap for the merging car. Furthermore, speed and position changes of these vehicles are given in Fig. 7. As seen from the figure, the position difference between the vehicles is increased by the time and the distance between the vehicles is as much as the distance Δx_{min} at the final point. In addition, all of vehicles are started the simulation with the maximum speed. The front vehicle is continued at the maximum speed and the middle and rear vehicles are decreased their speed by the time. However, it is observed that the rear vehicle reduces the speed more than the middle one. It is also seen that the speed change of the vehicles is variable, up and down basis. This can adversely affect engine performance and driving comfort. Future work will suggest solutions to this situation.

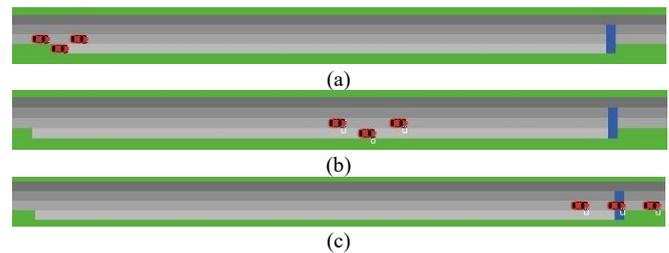


Fig. 6. Situation of merge algorithm: a) start; b) second; c) final positions.

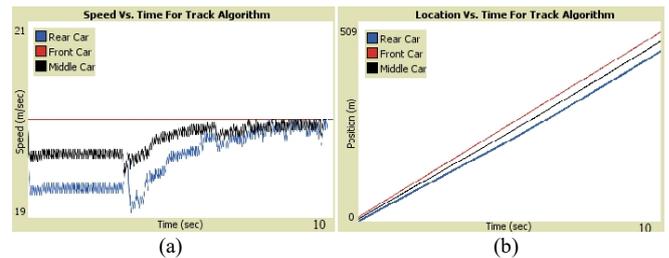


Fig. 7. Merge algorithm verification about: a) speed and; b) position.

As mentioned above, our proposed algorithm works with different speed vehicles at merging zone, i.e. vehicles do not have to be the same speed. In order to verify this property, some simulations are performed. Position versus time for a vehicle is plotted and given in Fig. 8.

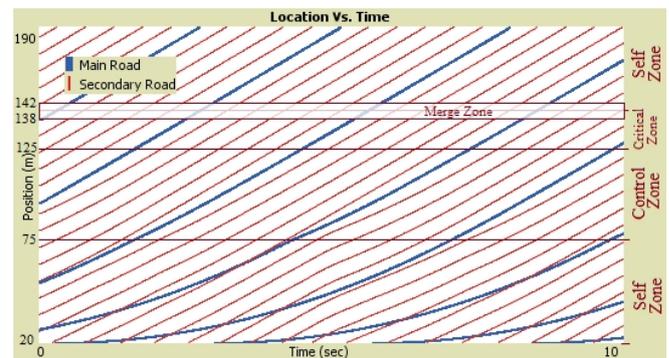


Fig. 8. Distance of vehicles traveled in merging coordination.

As seen from the figure, position of vehicles on the main road is plotted as blue whereas secondary road's one is plotted as red. Vehicles obey the proposed algorithm to pass the merging point safely. Please note that, leading vehicle at the self or control zone can be following vehicle at the critical zone since they adjust their speed to obey algorithm rules. Since companion vehicles are lined up before the

critical zone, only one vehicle at a specific time is crossing the merging.

VI. CONCLUSIONS

In this paper, we developed a tracking and merging algorithms for MCS vehicles in order to ensure safe and efficient flow at merging zone. We design a main and a secondary road to test our algorithm with NetLogo simulation platform. The efficiency of the proposed system was validated through simulation. By using the proposed algorithms, traffic flow can be easily controlled with different vehicles speed. Headway of vehicles entering the secondary road is forced to be as the normal distribution. The platform is tested with mean as 3000 and standard deviation as 1000. All simulation results agree that the proposed algorithm ensure for safe merging for MCS vehicles. Moreover, it is guaranteed that no collision at anywhere including merging zone. These results are encouraged us for further studies about MCS algorithms.

REFERENCES

- [1] C. Lee, Y. Lim, D. Kim, K. Sohng, "Improved vehicle detection algorithm in heavy traffic for intelligent vehicle", *Elektronika ir Elektrotechnika*, vol. 20, no. 9, pp. 54–58, 2014. DOI: 10.5755/j01.eee.20.9.3734.
- [2] J. Rios-Torres, A. A. Malikopoulos, P. Pisu, "Online optimal control of connected vehicles for efficient traffic flow at merging roads", in *Proc. 18th Int. Conf. Intell. Transp. Syst.*, Spain, 2015, pp. 2432–2437. DOI: 10.1109/ITSC.2015.392.
- [3] J. Rios-Torres, A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps", *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, 2017. DOI: 10.1109/TITS.2016.2600504.
- [4] D. Baselt, F. Knorr, B. Scheuermann, M. Schreckenber, M. Mauve, "Merging lanes-fairness through communication", *Vehicular Communications*, vol. 1, no. 2, pp. 97–104, 2014. DOI: 10.1016/j.vehcom.2014.05.005.
- [5] Y. Wang, E. Wenjuan, W. Tang, D. Tian, G. Lu, G. Yu, "Automated on-ramp merging control algorithm based on internet-connected vehicles", *IET Intell. Transp. Syst.*, vol. 7, no. 4, pp. 371–379, 2013. DOI: 10.1049/iet-its.2011.0228.
- [6] J. Rios-Torres, A. A. Malikopoulos, "Automated and cooperative vehicle merging at highway on-ramps", *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 4, pp. 780–789, 2017. DOI: 10.1109/TITS.2016.2587582.
- [7] K. Dresner, P. Stone, "A multiagent approach to autonomous intersection management", *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, 2008. DOI: 10.1613/jair.2502.
- [8] S. Huang, A. W. Sadek, Y. Zhao, "Assessing the mobility and environmental benefits of reservation-based intelligent intersections using an integrated simulator", *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1201–1214, 2012. DOI: 10.1109/TITS.2012.2186442.
- [9] T. Au, P. Stone, "Motion planning algorithms for autonomous intersection management", *AAAI 2010 Work. Bridg. Gap Between Task Motion Plan.*, 2010, pp. 2–9.
- [10] K. Dresner, P. Stone, "Multiagent traffic management: a reservation-based intersection control mechanism", in *Proc. Third Int. Jt. Conf. Auton. Agents Multiagent Syst. (AAMAS 2004)*, 2004, pp. 530–537.
- [11] Q. Jin, G. Wu, K. Boriboonsomsin, M. Barth, "Platoon-based multiagent intersection management for connected vehicle", in *16th Int. IEEE Conf. Intell. Transp. Syst. Proceedings (ITSC 2013)*, Netherlands, 2013, pp. 1462–1467. DOI: 10.1109/ITSC.2013.6728436.
- [12] C. Wuthishuwong, A. Traechtler, "Coordination of multiple autonomous intersections by using local neighborhood information", in *Int. Conf. Connect. Veh. and Expo (ICCVE 2013)*, USA, 2013, pp. 48–53. DOI: 10.1109/ICCVE.2013.6799768.
- [13] I. H. Zohdy, H. Rakha, "Game theory algorithm for intersection-based cooperative adaptive cruise control (CACC) systems", in *15th Int. IEEE Conf. Intell. Transp. Syst.*, USA, 2012, pp. 1097–1102. DOI: 10.1109/ITSC.2012.6338644.
- [14] P. Kachroo, Z. Li, "Vehicle merging control design for an automated highway system", in *Proc. Conf. Intell. Transp. Syst.*, USA, 1997, pp. 224–229. DOI: 10.1109/ITSC.1997.660479.
- [15] A. Uno, T. Sakaguchi, S. Tsugawa, "A merging control algorithm based on inter-vehicle communication", in *Proc. IEEE/IEEE/JSAI Int. Conf. Intell. Transp. Syst.*, Japan, 1999, pp. 783–787. DOI: 10.1109/ITSC.1999.821160.
- [16] C. Letter, L. Elefteriadou, "Efficient control of fully automated connected vehicles at freeway merge segments", *Transp. Res. Part C*, vol. 80, pp. 190–205, 2017. DOI: 10.1016/j.trc.2017.04.015.
- [17] X. Lu, K. J. Hedrick, "Longitudinal control algorithm for automated vehicle merging", in *Proc. 39th IEEE Conf. Decis. Control.*, Australia, 2000, pp. 450–455. DOI: 10.1109/CDC.2000.912805.
- [18] M. A. S. Kamal, J. Imura, A. Ohata, T. Hayakawa, K. Aihara, "Coordination of automated vehicles at a traffic-lightless intersection", in *Proc. 16th Int. IEEE Annu. Conf. Intell. Transp. Syst.*, Netherlands, 2013, pp. 922–927. DOI: 10.1109/ITSC.2013.6728350.
- [19] L. Makarem, D. Gillet, "Model predictive coordination of autonomous vehicles crossing intersections", in *Proc. 16th Int. IEEE Annu. Conf. Intell. Transp. Syst.*, Netherlands, 2013, pp. 1799–1804. DOI: 10.1109/ITSC.2013.6728489.
- [20] V. Milanés, J. Perez, E. Onieva, C. Gonzalez, "Controller for Urban intersections based on wireless communications and fuzzy logic", *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 1, pp. 243–248, 2010. DOI: 10.1109/TITS.2009.2036595.
- [21] X. Lu, H. Tan, S. Shladover, J. Hedrick, "Automated vehicle merging maneuver implementation for AHS", *Veh. Syst. Dyn.*, vol. 41, no. 2, pp. 85–107, 2004. DOI: 10.1076/vesd.41.2.85.26497.
- [22] Y. Xie, H. Zhang, N. Gartner, T. Arsava, "Collaborative merging behaviors and their impacts on freeway ramp operations under connected vehicle environment", *J. of Intelligent Transportation Systems*, vol. 21, no. 2, pp. 136–147, 2016. DOI: 10.1080/15472450.2016.1248288.
- [23] F. Zhu, S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment", *Transp. Res. Part C*, vol. 55, pp. 363–378, 2015. DOI: 10.1016/j.trc.2015.01.006.
- [24] Q. Zhu, C. Peng, J. Shi, P. Duan, Y. Bao, M. Xie, "Cooperative traffic light control based on semi-real-time processing", *J. Autom. Control Eng.*, vol. 3, no. 6, pp. 463–469, 2015. DOI: 10.12720/joace.3.6.463-469.
- [25] W. Cao, M. Mukai, T. Kawabe, H. Nishira, N. Fujiki, "Cooperative vehicle path generation during merging using model predictive control with real-time optimization", *Control Eng. Pract.*, vol. 34, pp. 98–105, 2015. DOI: 10.1016/j.conengprac.2014.10.005.
- [26] J. Lee, B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment", *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 81–90, 2012. DOI: 10.1109/TITS.2011.2178836.
- [27] W. Wu, J. Zhang, A. Luo, J. Cao, "Distributed mutual exclusion algorithms for intersection traffic control", *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 65–74, 2015. DOI: 10.1109/TPDS.2013.2297097.
- [28] Y. J. Zhang, A. A. Malikopoulos, C. G. Cassandras, "Optimal control and coordination of connected and automated vehicles at urban traffic intersections", in *Proc. Am. Control Conf.*, USA, 2016. DOI: 10.1109/ACC.2016.7526648.
- [29] H. Ramezani, R. F. Benekohal, "Optimized speed harmonization with connected vehicles for work zones", in *IEEE 18th Int. Conf. Intell. Transp. Syst.*, Spain, 2015, pp. 1081–1086. DOI: 10.1109/ITSC.2015.179.
- [30] J. Lee, B. Park, K. Malakorn, J. So, "Sustainability assessments of cooperative vehicle intersection control at an urban corridor", *Transp. Res. Part C*, vol. 32, pp. 193–206, 2013. DOI: 10.1016/j.trc.2012.09.004.
- [31] Z. Li, M. V. Chitturi, L. Yu, A. R. Bill, D. A. Noyce, "Sustainability effects of next-generation intersection control for autonomous vehicles", *Transport*, vol. 30, no. 3, pp. 342–352, 2015. DOI: 10.3846/16484142.2015.1080760.
- [32] A. T. Tola, M. Bozuyula, "A novel transportation system: metrocarr", *Electronics World*, submitted for publication.
- [33] S. Agarwal, P. Kachroo, S. Contreras, S. Sastry, "Feedback-coordinated ramp control of consecutive on-ramps using distributed modeling and godunov-based satisfiable allocation", *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2384–2392, 2015. DOI: 10.1109/TITS.2015.2398453.
- [34] J. Zheng, K. Suzuki, M. Fujita, "Predicting driver's lane-changing decisions using a neural network model", *Simul. Model. Pract. Theory*, vol. 42, pp. 73–83, 2014. DOI: 10.1016/j.simpat.2013.12.007.