

Vehicle-to-Everything (V2X) Communication: A Roadside Unit for Adaptive Intersection Control of Autonomous Electric Vehicles

Michael Evans^{1,*}, Marcial Machado², Rickey Johnson³, Anna Vadella⁴, Luis Escamilla⁵,
Beñat Froemming-Aldanondo⁶, Tatiana Rastoskueva⁷, Milan Jostes⁸, Devson Butani⁸,
Ryan Kaddis⁸, Chan-Jin Chung⁸, and Joshua Siegel⁹

Abstract—Recent advances in autonomous vehicle technologies and cellular network speeds motivate developments in vehicle-to-everything (V2X) communications. Enhanced road safety features and improved fuel efficiency are some of the motivations behind V2X for future transportation systems. Adaptive intersection control systems have considerable potential to achieve these goals by minimizing idle times and predicting short-term future traffic conditions. Integrating V2X into traffic management systems introduces the infrastructure necessary to make roads safer for all users and initiates the shift towards more intelligent and connected cities. To demonstrate our solution, we implement both a simulated and real-world representation of a 4-way intersection and crosswalk scenario with 2 self-driving electric vehicles, a roadside unit (RSU), and traffic light. Our architecture minimizes fuel consumption through intersections by reducing acceleration and braking by up to 75.35%. We implement a cost-effective solution to intelligent and connected intersection control to serve as a proof-of-concept model suitable as the basis for continued research and development. Code for this project is available at <https://github.com/MMachado05/REU-2024>.

I. INTRODUCTION

Crosswalks oriented along street intersections are among the most dangerous for pedestrians, accounting for up to 60% of injuries caused by vehicles in cities such as Montreal [1]. This vulnerability demands a safer approach to managing traffic; one method of achieving safer crosswalks is to deploy V2X-enabled roadside units for adaptive intersection control.

Vehicle-to-vehicle (V2V) [2], [3], vehicle-to-cyclist [4], and vehicle-to-infrastructure (V2I) communications [5] are incorporated into modern vehicles to optimize for functions such as avoiding delays or minimizing stop counts [6], [7]. V2I optimization methods are categorized as NP-hard problems [8], limiting the scope of usability to centralized data due to the algorithmic complexity of the computations [6]. Similar V2X approaches to scheduling optimization [9] use real time traffic information instead of centralized data and can enable drivers to take early action [10]. Dedicated Short-Range Communication (DSRC) radios [11] have long been used for V2V communication [12], and are utilized

This project is supported by the National Science Foundation REU Site Awards #2150096 and #2150292. The electric vehicles used for this research are sponsored by MOBIS, US Army GVSC, NDIA, DENSO, SoarTech, Realtime Technologies, Veoneer, Dataspeed, GLS&T, and LTU.

*Corresponding author: Michael Evans, mevan028@odu.edu

¹Old Dominion University; ²The Ohio State University; ³North Carolina A&T State University; ⁴Butler University; ⁵New Mexico State University; ⁶University of Minnesota; ⁷The University of Arizona; ⁸Lawrence Technological University; ⁹DeepTech Lab, Michigan State University.

in research implementations of safety features such as crash avoidance. By aggregating existing technologies from DSRC and cellular networks, V2X offers a software framework for exchanging information between vehicles and components of the intelligent transportation system (ITS) [13].

We propose a cost-efficient V2X wireless communication architecture with a roadside unit to augment the metrics of fuel efficiency, safety, comfort, and driving behaviors in cities. This research emphasizes low costs, as widespread deployments of RSUs by the U.S. Department of Transportation (DoT) have been canceled due to the large funding requirements [14]. Our connected intersection and crosswalk scenario model is implemented using two Polaris Gem e2 electric vehicles (EVs) known at LTU as Autonomous Campus Transport (ACTor) vehicles. Each ACTor is equipped with the hardware necessary for self-driving, including a Dataspeed Drive-by-Wire kit [15], HDR Camera, 2D and 3D LiDAR sensors, two Swift Piksi GPS modules, and computer for programming the Drive-by-Wire system with ROS [16]. The dimensions of ACTor 1 are shown in Figure 1 below.

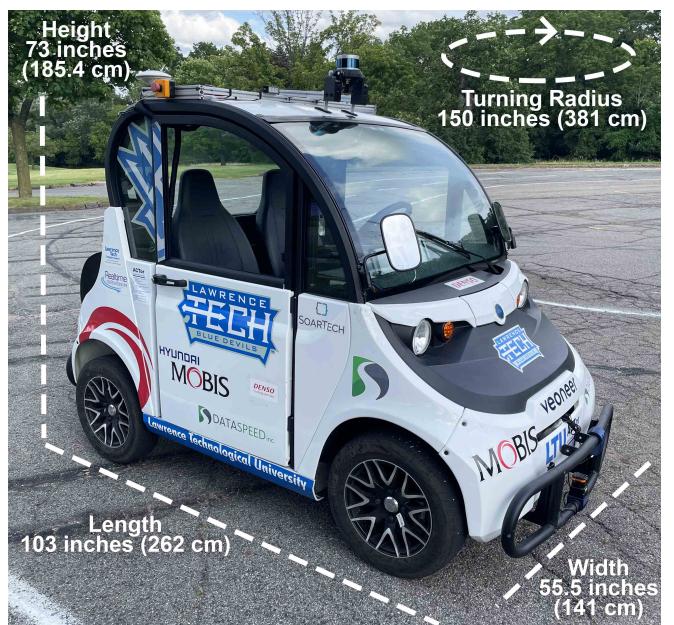


Fig. 1: Image of the Polaris GEM e2 “ACTor 1” vehicle.

Our work formalizes an architecture that supplements the data-sensing capabilities of connected vehicles by introduc-

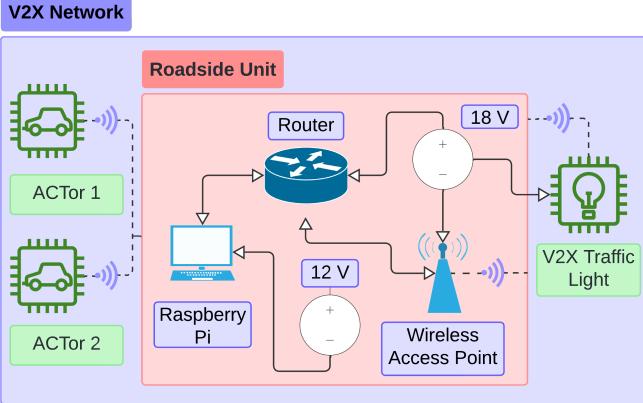


Fig. 2: Simplified architecture of the RSU and V2X network. The connected vehicles receive traffic light information routinely published by the RSU via a 2.5 GHz Wi-Fi connection. Vehicles connected to the wireless access point react to the state of the light without computer vision or human control.

ing a roadside unit (RSU), the centralized server facilitating intra-agent communication. As shown in Figure 2, both vehicles establish a wireless connection by joining a private network hosted on the RSU Linux server, continuously distributing information about the current and future state of the intersection. To generate a V2I connection between the vehicles and RSU, we use a 12V battery, 12V-5V converter, a Raspberry Pi 4 B single-board computer running the Ubuntu Server 20.04 operating system, and an external Netgear router with range extenders and a wireless access point. A minimal hardware configuration enables cost-effective deployment in a largely unrestricted range of environments due to the unit's power source and physical dimensions.

This paper proposes a cost-effective implementation model for a V2X-connected dynamic intersection control system. With each roadside unit costing \$17,680 in 2014 [17], we aim to lower this barrier to fuel-efficient intersections by offering a minimally expensive software solution. Our architecture emphasizes ease of deployment and implementation, using generalized communication protocols via ROS and Wi-Fi. Deploying RSUs in place of new 4-way traffic signals costing \$400,000 [18] in the future may help supplement the cost of transitioning to more connected and intelligent cities.

To make the progression towards adaptive intersections for autonomous vehicles, we introduce a V2X proof-of-concept system consisting of an RSU, 2 self-driving electric vehicles, and an Arduino-powered traffic light for state visualization. The significant contributions of our research, outlined in the remainder of the paper, are as follows:

- 1) Cultivate a range of machine learning algorithms for lane following to supplement the vehicles in our system.
- 2) Simulate our proposed solution first in a virtual environment of our Lot H test course using GazelleSim.
- 3) Develop and evaluate a real-world representation of our V2X model operating in both an emulated 4-way intersection and cross-walk scenario.

II. REVIEW OF LITERATURE

Most existing research efforts on traffic management systems with V2X are only tested in simulation [19], [20], [21], [22] due to the costs and logistics of real-world models for such implementations. State-of-the-art projects combine network simulators such as ns-3 [23] with traffic simulators like SUMO [24] for applications on vehicle platooning, sensor sharing, and communication protocol conformance testing [25]. While these projects aim to provide realistic V2X scenarios by running robust simulation, their pursuits are still not perfect representations of the physical environments they aim to model. Communication simulation delay is introduced in virtual environments, which, if high, can not reflect reality [25]. Obstacles such as adverse weather conditions, variances in the kinematics between vehicle types, uneven pavement conditions, driver ability [26], and networking limitations can not be understood by simulation alone. The speeds of these simulations is also a concern, where performance degrades exponentially when modeling high traffic densities [27].

Gaps in the research also exist for work that does include field testing. Lu, Jung, and Kim [28] propose a solution called *Vehicle-to-Intersection*, which assumes all vehicles are capable of autonomous collision avoidance and are controlled by an Intersection Control Agent, however, they do not consider deployment costs. An approach for displaying occupancy grids of vehicles in close proximity with an RSU has been covered in [29], but does not provide the autonomous intersection control that our research emphasizes. Specific use cases for intersection control are proposed in [30] for crossing-path collision avoidance by broadcasting sensor information about other road users at set frequencies. Efforts to lower the cost of deployment exist by reducing the number of units required through optimizing coverage ratios [31], [32], or supplementing parked cars as RSUs [14], [33], but do not address maximizing the cost efficacy of a single unit.

Other works considered in this section examine specific and important components of connected traffic management with V2X. The existing gaps in research are due to using fully simulated environments, not considering deployment costs, or requiring human input in response to an in-car display. Our research contributes to this area by incorporating an adaptive-speed algorithm into a connected intersection enabled by communication with a cost-efficient roadside unit.

III. SYSTEM DESIGN AND METHODS

To evaluate the performance of the V2X system with our proposed intersection control, we first develop a virtual simulation using GazelleSim [34], and an aerial-view mapping shown in Figure 3 of the Lawrence Technological University (LTU) Parking Lot H located in Southfield, Michigan. This simulator is used for its ability to simulate multiple agents simultaneously, and lightweight power consumption, enabling simulation testing on field laptops. Next, we define the implementations of the real-time system including the RSU, ACTor vehicles, and traffic light for human visualization. An overview of the ROS software architecture components and detailed summary of each respective development follows.

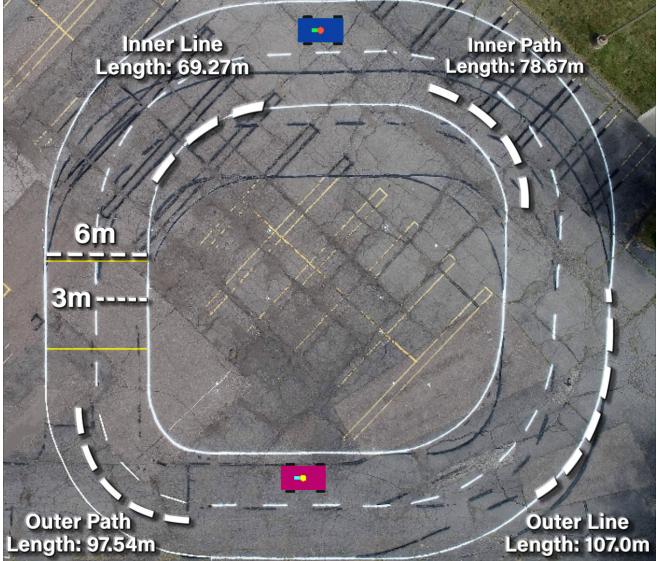


Fig. 3: Map of the Lot H test course in GazelleSim.

A. Gazelle Simulation

As suggested in Section II, it is not always practical or possible to acquire the hardware needed for real-world testing of a V2X system. In such situations, or for initial testing of new software features, algorithms are first implemented in simulation. This method of software development allows for the debugging of new features in a controlled, safe, and weather-independent environment. We use a lightweight simulator developed at LTU: GazelleSim, two Ackermann steering robots [35] with turning capabilities similar to the ACTor vehicles, and an aerial photo of the Lot H course as our simulation environment. The simulator uses a *meters per pixel* parameter to accurately display the position of both vehicles on the map as they would appear in a real-world test at that GPS location. By combining the meters per pixel parameter with a starting point GPS coordinate plotted on the map, our simulated environment accurately represents the real-world location both visually and geographically.

Each vehicle follows a software architecture similar to Figure 4, using software-in-the-loop (SIL) testing [36] for the Drive-by-Wire system and GPS which are simulated and handled by GazelleSim. In this virtual environment, the vehicle's controller publishes twist messages in the form of linear and angular x, y, and z values instead of native Drive-by-Wire commands. All simulated agents have access to their x and y coordinates for continuous tracking position without requiring latitude and longitude from a GPS node. Another deviation from real-world testing is that the RSU is implemented as a ROS node: an efficient virtualization of the physical Ubuntu server and network structure detailed in Figure 2. RSUs can be exorbitant to implement [37], often necessitating the use of virtual testing before implementing the hardware and networking required of a physical system. The remainder of the simulation software aligns with the real-world implementation as discussed in Section III-B.

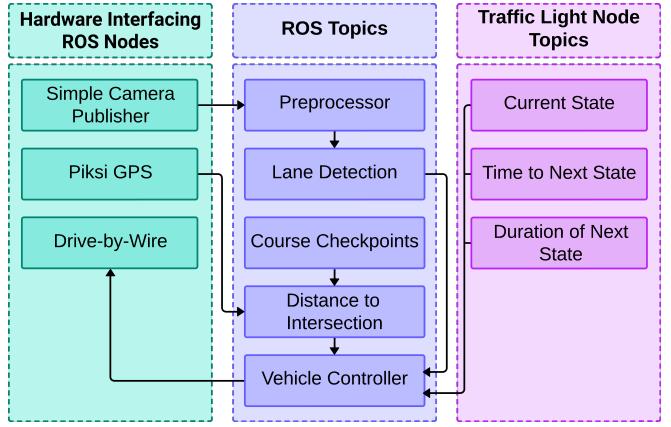


Fig. 4: Diagram of the V2X software architecture.

B. Software Architecture

We develop a high-level software architecture and a corresponding ROS workspace directory structure to increase modularity and ease of iteration. Our codebase is organized into three primary categories: the hardware interface, drive logic, and traffic light model. This modular structure allows for the efficient substitution of discrete code sectors, such as the lane detection node, for making exploratory system performance comparisons with minimal overhead.

Our hardware-interfacing logic handles three primary functions: a package for reading image information from the ACTor's HDR camera, one that receives GPS information from the Piksi GPS nodes, and an API for communicating movement directives to the ACTor vehicles. These packages function as a mediator between physical objects and the data they capture. The next section will outline the responsibility of each software node in the V2X architecture.

We use an image preprocessor node to receive raw images from the camera-compatible package and prepare them for lane detection through processing algorithms such as median blur, canny edge detection [38], and white filtering. This node provides the functionality of altering the raw image separately from the lane following logic, as some algorithms perform better with differing parameters. Without this, each lane-following algorithm will need to process the image individually, causing large repetitions in the code. Next, the processed image is input into the lane detection node which calculates the locations of lane borders to derive a desired turn angle. This node benefits from the previous step, as our K-means [39] algorithm prefers canny edge detection, while DBSCAN [40] performs better without it.

With lane following handled by the previous steps, our adaptive speed algorithm logic will now be explained. The vehicle's current location and internally saved course waypoints are used to calculate the distance to the intersection from either lane. Further information regarding this calculation are explained in Section III-D. The code implemented on the RSU, which is activated via an SSH connection into the Raspberry Pi at runtime, works alongside the distance-to-intersection node to manage the intersection. On the RSU is

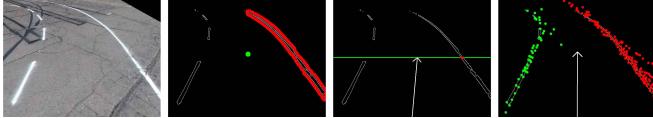


Fig. 5: Largest-contour, K-means, and DBSCAN algorithms applied to the raw camera image output from the vehicles.

the traffic light algorithm, which keeps track of an abstraction of its current light color, and the duration of all states. Each traffic light state consists of a configurable length of time that it will remain in this configuration for and a binary representation of either a red or green light. This node publishes information on state changes of the traffic light, the time left in the current state, and the duration of the next state in three unique ROS topics. Each topic is subscribed to by the vehicle controller node, which, together with distance calculation, allows for our adaptive speed algorithm to function. The RSU's primary function is to publish information about the traffic light to provide intersection awareness to the vehicles, which individually alter their speeds in response if required.

Finally, the vehicle controller node takes information from the distance-to-intersection node and the lane detection node to make a concluding decision on vehicle movement. All other nodes serve in some capacity as input parameters to the final output command from the vehicle controller.

C. Lane-following Algorithms

During this project, we develop a DeepLSD [41], DBSCAN, K-means, Least Squares Regression, and largest-contour method of lane following to find the most reliable, see Figure 5. After testing the number of attempts required by each algorithm to complete 5 consecutive laps, we selected K-means as the most successful. This algorithm starts with the preprocessing node and modifies the image to enable more effective data extraction in a later step. First, median blur is applied to remove noise from the image while preserving hard edges. The image is then converted from the RGB color space, which has 3 channels, to a one-dimensional grey scale color space. After this color translation, the new image conserves each white pixel and allows for easier ranges of threshold filtering values to be found. Next, the white regions are masked out and the final image is published to a lane detection node. Finally, dilation is used to increase the white object area and to accentuate them, resolving the issue of pavement cracks and poorly painted lane lines affecting the process of identifying the largest white contour.

The next step focuses exclusively on the central horizontal row to find all locations in which the row has white pixels. In the ideal case, it will detect two groups of white pixels, one for each lane line. Instead of finding the mean of those points, it is a better approximation to apply K-means with K=2 to find the centroid of each group, and then take the average. K-means clustering works by first initializing a set number of cluster centers (centroids). It then iteratively assigns each data point to the nearest centroid and updates the centroids to be the mean of the data points in each cluster. This process

repeats until the centroids stabilize, meaning they no longer change significantly with further iterations.

D. Adaptive Speed Control

To reduce vehicle emissions and noise pollution attributed to idling at red lights [42], [43], the RSU sends data to the vehicle controller which adjusts the vehicle's speeds. In effect, this synchronizes each agent's approach to the intersection. Variations in vehicle speeds at traffic intersections lead to an increase in fuel usage and a decrease in air quality for the immediate and surrounding area [44], [45]. To solve this, we construct an adaptive speed algorithm using GPS waypoints and a kinematics equation to calculate the target average velocity that each vehicle should drive to approach an intersection as the traffic state switches from red to green.

Our process for capturing the waypoints involves pre-recording GPS coordinates of the course and the intersection in lieu of commercially available HD automotive maps data. During this process, one student pilots the ACTor while another indicates where to record each waypoint for a consistent distance of approximately 3 meters between measurements. The latitude and longitude values of each intersection and standard waypoint are saved to 2 yaml files.

At each light state change, the vehicle controller determines if each vehicle can cross the intersection driving at its current velocity before the traffic state changes again. If so, or if the vehicle loses connection to the RSU, the vehicle controller assigns no change in target velocity. If it can not make the intersection in time, the following process is used to calculate an average target velocity to maintain:

Definitions:

Let $v = (\phi_v, \lambda_v)$ be the coordinates of vehicle v .

Let $w_i = (\phi_i, \lambda_i)$ be the coordinates of waypoint w_i .

Let $W = \{w_1, w_2, \dots, w_n\}$ be the set of all waypoints.

Let $I \subset W$ be the set of all intersections.

The distance between 2 points on a sphere can be calculated using the Cosine-Haversine formula [46] $h(\phi_1, \lambda_1, \phi_2, \lambda_2) =$

$$2r \cdot \arcsin \sqrt{\sin^2 \left(\frac{\Delta\phi}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\Delta\lambda}{2} \right)}$$

Find the waypoint w_i with the shortest distance to v :

$$\text{Let } p = \min_{x \in W} h(x, v)$$

Sum $h(w_i, w_{i+1})$ from p to the next intersection:

$$\Delta x_{\text{total}} = \sum_{i=p}^{k-1} h(w_i, w_{i+1}) \quad : w_k \in I \text{ and } k > p$$

Find the target average velocity the vehicle should drive:

$$v_{\text{avg}} = \frac{\Delta x}{\Delta t} \quad : \Delta x = \Delta x_{\text{total}} \text{ and } \Delta t = \text{light duration}$$

E. Traffic Light Visualization

Our traffic light is composed of 4 individual LED lights, relays, a controller for the relays, a power supply, a logic radio module/wifi, and is controlled through an Arduino WeMos D1 board. On state change, the RSU sends a Rosserial message with the state of the traffic lights to the board. This message in turn sets the pins of the traffic light to light up the specific light configuration to show the respective state. As connected vehicles have access to light duration, yellow lights are unnecessary and act as a fail-operational identifier for a loss of connection to the roadside unit. The V2X system is unaffected by connection interruptions to the traffic light.

IV. EXPERIMENT AND RESULTS

In this section, we first introduce the real-world environment setup for testing our V2X adaptive speed algorithm on two electric vehicles. The detailed comparison for each light configuration and how we collected the data for each experiment will be explained. Then, we compare average velocity, acceleration, and data recorded from an inertial measurement unit (IMU) [47] for each task. Finally, we analyze our results to show that our adaptive speed algorithm reduces the total change in vehicle speeds through intersections, and prevents idling due to full stops.

Two scenarios are designed for this project: a crosswalk with synchronized light states allowing pedestrians to pass a critical zone, and a simplified 4-way intersection with no left turns, and independent light states. During our demonstration, both vehicles are tested successfully and simultaneously on both scenarios.¹ However, for the data collection in this section, we test only a single vehicle during runtime and discerning between scenarios becomes ineffectual.

A. Experimental Setup

Data collection for each evaluation begins at the initial green light encountered by the ACTor, which is enabled to drive the first frame that a green light is registered. For each experiment, we also include a human driver to serve as a control. The driver is instructed to maintain a speed of 5 mph (2.24 m/s) and receives a verbal 5-second warning of state changes to emulate real-world yellow lights. Additionally, each experiment is finalized as the ACTor encounters the intersection in the outer lane for a second time, noted by the top yellow line in Figure 3.

B. Performance Analysis

We evaluate this V2X architecture both quantitatively and qualitatively in the form of velocity, IMU data, cost to deploy, and reductions made to acceleration across the intersection. The isolated range of values in the highlighted region of Figure 6 is used to analyze the percent decrease in vehicle speed-ups and braking through an intersection:

$$\left(\frac{\int_{t=35s}^{t=65s} |a(\text{human})| dt - \int_{t=35s}^{t=65s} |a(\text{adaptive})| dt}{\int_{t=35s}^{t=65s} |a(\text{human})| dt} \right) \cdot 100$$

¹<https://www.youtube.com/watch?v=OeRlhWqmIgE>

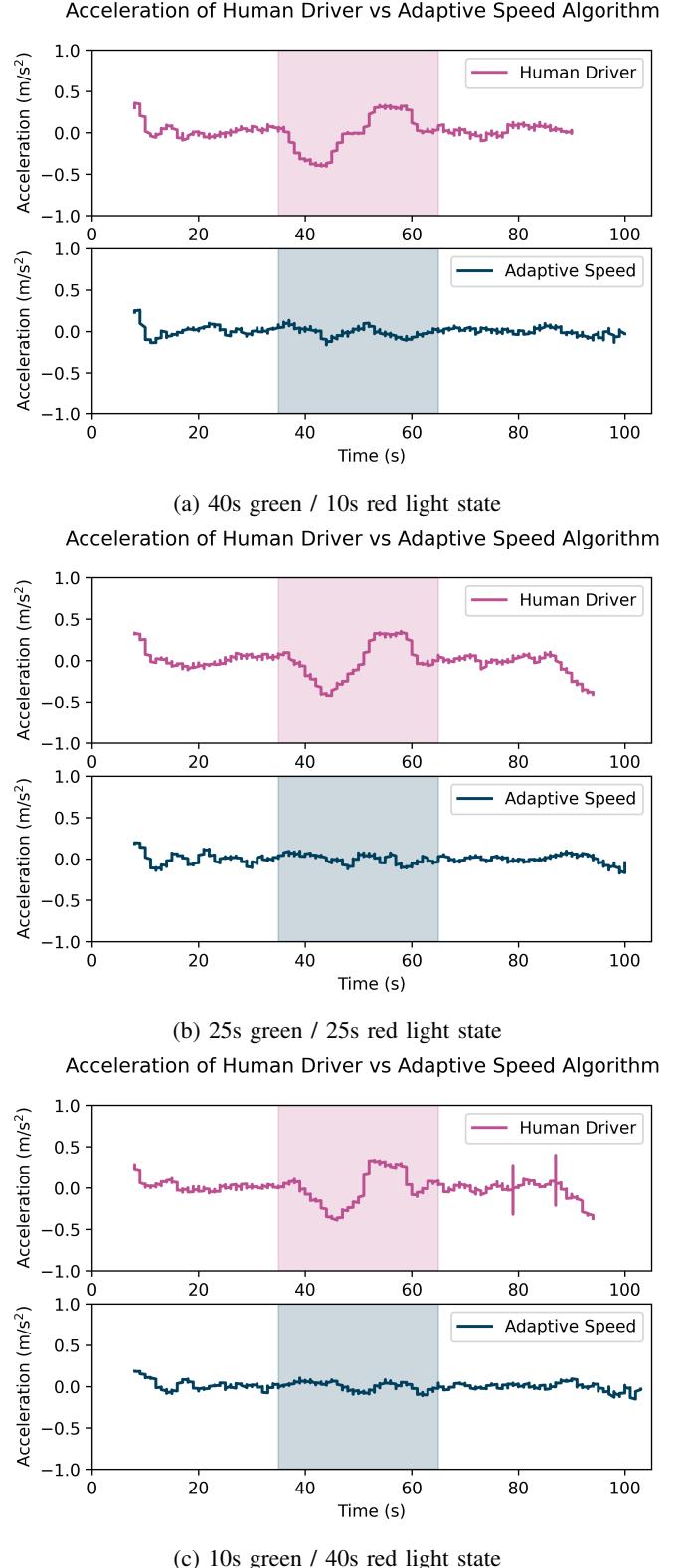


Fig. 6: Acceleration vs time comparison of a human driver and non-adaptive speed algorithms against 3 light state configurations. The highlighted region is from 35s to 65s and captures crossing the intersection.

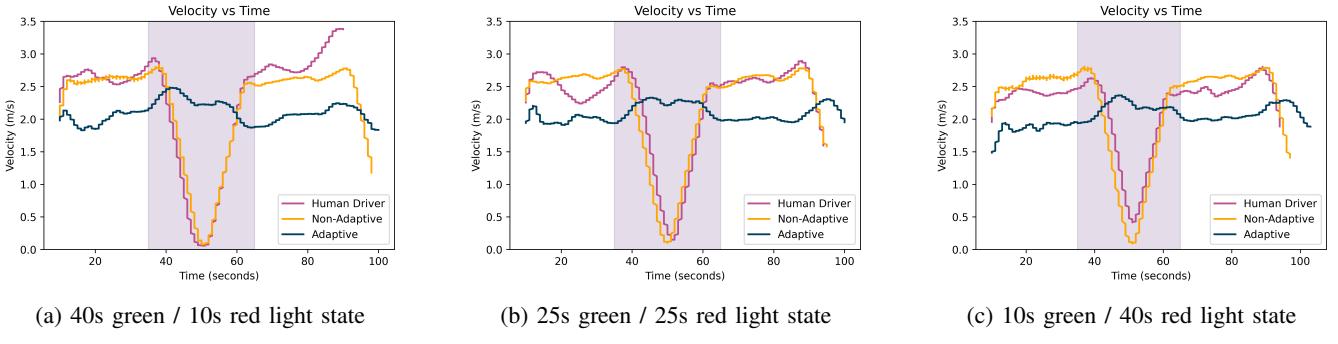


Fig. 7: Velocity vs time comparison of all driving methods against 3 light state configurations.

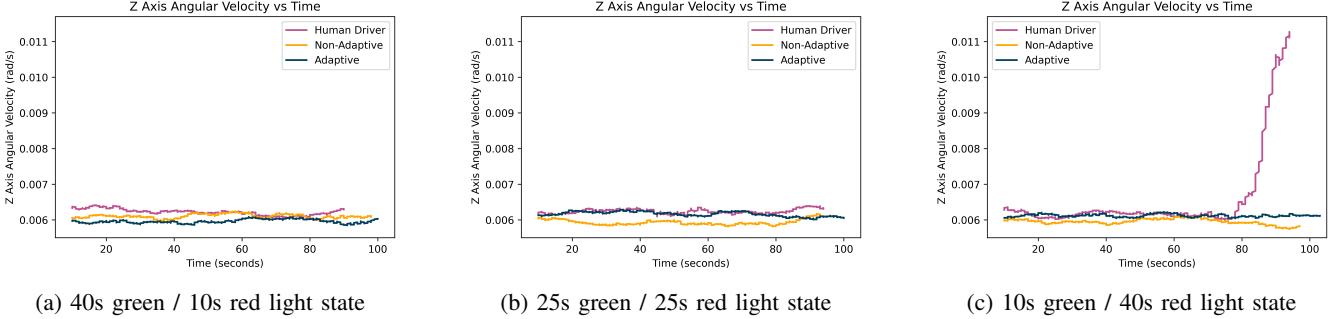


Fig. 8: Z axis angular velocity vs time comparison of all driving methods against 3 light state configurations.

Our proposed algorithm reduces the total change in velocity through the highlighted region of 73.15%, 75.35%, and 73.79% respectively across the three trials when compared to a human driver. By minimizing changes in velocity at intersections, we also reduce fuel consumption [48], [49], [50]. Moreover, this architecture is highly scalable from a funding perspective due to each unit costing less than \$500 to deploy. Minimal overhead may promote the widespread use of pedestrian detection systems that provide a more considerable warning than conventional mechanisms to drivers and smartphone-distracted pedestrians [51], [52], [53].

TABLE I: Comparison of the average velocity of the V2X algorithms against a human driver during 2 laps around our test course.

Light States	Human	Non-Adaptive	Adaptive
40s green / 10s red	2.34 m/s	2.10 m/s	2.07 m/s
25s green / 25s red	2.14 m/s	2.16 m/s	2.06 m/s
10s green / 40s red	2.14 m/s	2.18 m/s	1.95 m/s

Next, we analyze the driving speed of our algorithm in comparison to both a non-adaptive algorithm and a human driver. This non-adaptive algorithm in use is the ACTor vehicle using the same K-means lane following method outlined in Section III-C but with a static stopping behavior in place of an adaptive one. The velocity maintained by the adaptive speed algorithm in Figure 7 is more regular and avoids the visible dip caused by a full stop at the intersection.

Table I also shows a comparison of average velocity for a more quantitative evaluation of our results. One possible

bias that we acknowledge is the duration of the light states. Configurations that do not force a red light encounter on intersection approach could generate less conclusive results. Another potential bias is vehicle speeds and traffic density. Our system is only tested within the capabilities of the lane-following algorithms, unfit for high speeds or dense traffic. These are two known network complications present in cities.

We are also interested in the impact on user experience for passengers. Figure 8 displays the z axis angular velocity captured by the vehicle’s IMU, which has been used previously to evaluate ride comfort on railways [54]. While there is not a significant correlation between the values recorded from the IMU and which algorithm is used, both consistently achieve the same or better performance as the control.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a cost-effective V2X approach for adaptive intersection control that eliminates idling, and reduces the total acceleration and deceleration through intersections by up to 75.35%. Our approach utilizes a cost-efficient roadside unit for wireless communication between vehicles and the traffic light state for a more fuel efficient intersection. As an extension to this research, we plan to shift the adaptive functionality from adjusting the vehicle speed to altering the state of the light. Another limitation to address is implementing additional fail states into our system. Constructing a fail-operational method during a loss of connection not reliant on human intervention or full stops would be ideal. Continued functionality during system failure is a requisite safety feature for deployment in uncontrolled city environments, such as intersections and crosswalks.

REFERENCES

- [1] M. Brosseau, S. Zangenehpour, N. Saunier, and L. Miranda-Moreno, "The impact of waiting time and other factors on dangerous pedestrian crossings and violations at signalized intersections: A case study in montreal," *Transportation research part F: traffic psychology and behaviour*, vol. 21, pp. 159–172, 2013.
- [2] A. Demba and D. P. Möller, "Vehicle-to-vehicle communication technology," in *2018 IEEE international conference on electro/information technology (EIT)*. IEEE, 2018, pp. 0459–0464.
- [3] J. E. Siegel, D. Erb, and S. E. Sarma, "A survey of the connected vehicle landscape – architecture, enabling technologies, applications, and development areas," *IEEE Transactions on Intelligent Transportation Systems*, 2017.
- [4] J. Barnett, N. Giziński, E. Mondragón-Parra, J. Siegel, D. Morris, T. Gates, E. Kassens-Noor, and P. Savolainen, "Automated vehicles sharing the road: Surveying detection and localization of pedalcyclists," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 4, pp. 649–664, 2021.
- [5] J. Santa, A. F. Gómez-Skarmeta, and M. Sánchez-Artigas, "Architecture and evaluation of a unified v2v and v2i communication system based on cellular networks," *Computer Communications*, vol. 31, no. 12, pp. 2850–2861, 2008.
- [6] B. Płaczek, "Self-organizing urban traffic control based on fuzzy cellular model," *Archives of Transport System Telematics*, vol. 5, pp. 29–34, 2012.
- [7] J. E. Siegel, D. C. Erb, and S. E. Sarma, "Algorithms and architectures: A case study in when, where and how to connect vehicles," *IEEE Transactions on Intelligent Transportation Systems Magazine*, vol. 10, no. 1, pp. 74–87, 2018.
- [8] A. Giridhar and P. Kumar, "Scheduling automated traffic on a network of roads," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 5, pp. 1467–1474, 2006.
- [9] P. T. Li and X. Zhou, "Recasting and optimizing intersection automation as a connected-and-automated-vehicle (cav) scheduling problem: A sequential branch-and-bound search approach in phase-time-traffic hypernetwork," *Transportation Research Part B: Methodological*, vol. 105, pp. 479–506, 2017.
- [10] J. Huang, D. Fang, Y. Qian, and R. Q. Hu, "Recent advances and challenges in security and privacy for v2x communications," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 244–266, 2020.
- [11] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.
- [12] R. Zhang, F. Schmutz, K. Gerard, A. Pomini, L. Bassetto, S. B. Hassen, A. Ishikawa, I. Ozgunes, and O. Tonguz, "Virtual traffic lights: System design and implementation," in *2018 ieee 88th vehicular technology conference (vtc-fall)*. IEEE, 2018, pp. 1–5.
- [13] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of dsrc and cellular network technologies for v2x communications: A survey," *IEEE transactions on vehicular technology*, vol. 65, no. 12, pp. 9457–9470, 2016.
- [14] A. B. Reis, S. Sargent, and O. K. Tonguz, "Parked cars are excellent roadside units," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2490–2502, 2017.
- [15] A. Goyal and A. Thakur, "An overview of drive by wire technology for automobiles," in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*. IEEE, 2019, pp. 108–110.
- [16] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [17] J. Wright, J. K. Garrett, C. J. Hill, G. D. Krueger, J. H. Evans, S. Andrews, C. K. Wilson, R. Rajbhandari, B. Burkhard *et al.*, "National connected vehicle field infrastructure footprint analysis," United States. Department of Transportation. Intelligent Transportation ..., Tech. Rep., 2014.
- [18] A. D. of Transportation, "Hsip construction pricing examples," Arizona Department of Transportation, Tech. Rep., 2020.
- [19] I. Soto, M. Calderon, O. Amador, and M. Urueña, "A survey on road safety and traffic efficiency vehicular applications based on c-v2x technologies," *Vehicular Communications*, vol. 33, p. 100428, 2022.
- [20] J. Choi and H. Kim, "A qos-aware congestion control scheme for c-v2x safety communications," in *2020 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2020, pp. 1–4.
- [21] S. Hegde, L. Shi, N. J. H. Marcano, R. Shrivastava, O. Blume, and R. H. Jacobsen, "Sidelink group resource scheduling for platoons in cellular vehicle-to-vehicle communications," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, 2021, pp. 1–5.
- [22] S. Hegde, O. Blume, R. Shrivastava, and H. Bakker, "Enhanced resource scheduling for platooning in 5g v2x systems," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 108–113.
- [23] L. Campanile, M. Gribaudo, M. Iacono, F. Marulli, and M. Mastrotianni, "Computer network simulation with ns-3: A systematic literature review," *Electronics*, vol. 9, no. 2, p. 272, 2020.
- [24] A. Kusari, P. Li, H. Yang, N. Punshi, M. Rasulis, S. Bogard, and D. J. LeBlanc, "Enhancing sumo simulator for simulation based testing and validation of autonomous vehicles," in *2022 ieee intelligent vehicles symposium (IV)*. IEEE, 2022, pp. 829–835.
- [25] J. Wang, Y. Shao, Y. Ge, and R. Yu, "A survey of vehicle to everything (v2x) testing," *Sensors*, vol. 19, no. 2, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/2/334>
- [26] M. H. Hosseiniou, H. Ahadi, and V. Hematian, "A study of the minimum safe stopping distance between vehicles in terms of braking systems, weather and pavement conditions," *Indian Journal of Science and Technology*, vol. 5, no. 10, pp. 3421–3427, 2012.
- [27] T. Ormándi and B. Varga, "Mesoscopic v2x simulation framework to enhance simulation performance," *Simulation Modelling Practice and Theory*, vol. 136, p. 103003, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1569190X24001175>
- [28] Q. Lu, H. Jung, and K.-D. Kim, "Optimization-based approach for resilient connected and autonomous intersection crossing traffic control under v2x communication," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 354–367, 2021.
- [29] M. Khalfin, J. Volgren, L. LeGoullon, B. Franz, S. Shah, T. Forgach, and M. Jones, "Vehicle-to-everything communication using a roadside unit for over-the-horizon object awareness," in *Proceedings of the IEOM International Conference on Smart Mobility and Vehicle Electrification, Southfield, MI, USA*, vol. 10, 2023.
- [30] L. Le, A. Festag, R. Baldessari, and W. Zhang, "V2x communication and intersection safety," in *Advanced Microsystems for Automotive Applications 2009: Smart Systems for Safety, Sustainability, and Comfort*. Springer, 2009, pp. 97–107.
- [31] L. Jiang, T. G. Molnár, and G. Orosz, "On the deployment of v2x roadside units for traffic prediction," *Transportation Research Part C: Emerging Technologies*, vol. 129, p. 103238, 2021.
- [32] S. Ben Chaabene, T. Yeferny, and S. Ben Yahia, "A roadside unit deployment framework for enhancing transportation services in maghrebian cities," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 1, p. e5611, 2021.
- [33] A. B. Reis, S. Sargent, and O. K. Tonguz, "Smarter cities with parked cars as roadside units," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2338–2352, 2018.
- [34] G. Pappas, J. E. Siegel, K. Politopoulos, and Y. Sun, "A gamified simulator and physical platform for self-driving algorithm training and validation," *Electronics*, vol. 10, no. 9, p. 1112, 2021.
- [35] J.-S. Zhao, X. Liu, Z.-J. Feng, and J. S. Dai, "Design of an ackermann-type steering mechanism," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 227, no. 11, pp. 2549–2562, 2013.
- [36] M. B. Ayed, L. Zouari, and M. Abid, "Software in the loop simulation for robot manipulators," *Engineering, Technology & Applied Science Research*, vol. 7, no. 5, 2017.
- [37] M. Al Shareeda, A. Khalil, and W. Fahs, "Towards the optimization of road side unit placement using genetic algorithm," in *2018 International Arab Conference on Information Technology (ACIT)*. IEEE, 2018, pp. 1–5.
- [38] P. Bao, L. Zhang, and X. Wu, "Canny edge detection enhancement by scale multiplication," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 9, pp. 1485–1490, 2005.
- [39] K. P. Sinaga and M.-S. Yang, "Unsupervised k-means clustering algorithm," *IEEE access*, vol. 8, pp. 80716–80727, 2020.
- [40] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.
- [41] R. Pautrat, D. Barath, V. Larsson, M. R. Oswald, and M. Pollefeyns, "DeepLSD: Line segment detection and refinement with deep image

- gradients,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17327–17336.
- [42] G. Mahler and A. Vahidi, “Reducing idling at red lights based on probabilistic prediction of traffic signal timings,” in *2012 American Control Conference (ACC)*. IEEE, 2012, pp. 6557–6562.
 - [43] K. Tiwari, R. Singh, and J. Balwanshi, “Fuel wastage and emission due to idling of vehicles at road traffic signals,” *Int. J. Res. Eng. Technol*, vol. 2, no. 10, pp. 43–53, 2013.
 - [44] S. Pandian, S. Gokhale, and A. K. Ghoshal, “Evaluating effects of traffic and vehicle characteristics on vehicular emissions near traffic intersections,” *Transportation Research Part D: Transport and Environment*, vol. 14, no. 3, pp. 180–196, 2009.
 - [45] L.-h. Li, J. Gan, and W.-q. Li, “A separation strategy for connected and automated vehicles: Utilizing traffic light information for reducing idling at red lights and improving fuel economy,” *Journal of Advanced Transportation*, vol. 2018, no. 1, p. 5679064, 2018.
 - [46] C. C. Robusto, “The cosine-haversine formula,” *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.
 - [47] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, “Reviews on various inertial measurement unit (imu) sensor applications,” *International Journal of Signal Processing Systems*, vol. 1, no. 2, pp. 256–262, 2013.
 - [48] M. H. Almannaa, H. Chen, H. A. Rakha, A. Loulizi, and I. El-Shawarby, “Reducing vehicle fuel consumption and delay at signalized intersections: controlled-field evaluation of effectiveness of infrastructure-to-vehicle communication,” *Transportation Research Record*, vol. 2621, no. 1, pp. 10–20, 2017.
 - [49] K. J. Malakorn and B. Park, “Assessment of mobility, energy, and environment impacts of intellidrive-based cooperative adaptive cruise control and intelligent traffic signal control,” in *Proceedings of the 2010 IEEE International Symposium on Sustainable Systems and Technology*. IEEE, 2010, pp. 1–6.
 - [50] A. Stevanovic, S. A. Shayeb, and S. S. Patra, “Fuel consumption intersection control performance index,” *Transportation research record*, vol. 2675, no. 9, pp. 690–702, 2021.
 - [51] F. Arena, G. Pau, and A. Severino, “V2x communications applied to safety of pedestrians and vehicles,” *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, p. 3, 2019.
 - [52] G. S. Larue, C. N. Watling, A. A. Black, J. M. Wood, and M. Khakzar, “Pedestrians distracted by their smartphone: Are in-ground flashing lights catching their attention? a laboratory study,” *Accident Analysis & Prevention*, vol. 134, p. 105346, 2020.
 - [53] S. Xia, D. de Godoy Peixoto, B. Islam, M. T. Islam, S. Nirjon, P. R. Kinget, and X. Jiang, “Improving pedestrian safety in cities using intelligent wearable systems,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7497–7514, 2019.
 - [54] P. Zoccali, G. Loprencipe, and R. C. Lupascu, “Acceleration measurements inside vehicles: Passengers’ comfort mapping on railways,” *Measurement*, vol. 129, pp. 489–498, 2018.