2021-01-0163

Assessing the Access to Jobs by Shared Autonomous Vehicles in Marysville, Ohio: Modeling, Simulating and Validating

Karina Meneses Cime, Mustafa Ridvan Cantas, Pedro Fernandez, Bilin Aksun-Guvenc, and Levent Guvenc Ohio State University
Adit Joshi, James Fishelson, and Archak Mittal Ford Motor Company

Abstract

Autonomous vehicles are expected to change our lives with significant applications like on-demand, shared autonomous taxi operations. Considering that most vehicles in a fleet are parked and hence idle resources when they are not used, shared on-demand services can utilize them much more efficiently. While ride hailing of autonomous vehicles is still very costly due to the initial investment, a shared autonomous vehicle fleet can lower its long-term cost such that it becomes economically feasible. This requires the Shared Autonomous Vehicles (SAV) in the fleet to be in operation as much as possible. Motivated by these applications, this paper presents a simulation environment to model and simulate shared autonomous vehicles in a geo-fenced urban setting. To simulate the aforementioned applications, a simulation environment that has a realistic rendering of the chosen real-world environment with realistic traffic generated around the SAVs is developed first using a geo-fenced area centered at the city of Marysville in Ohio as an example. This paper, then, presents an algorithm to optimally utilize multiple autonomous vehicles for shared rides based on modeling of pickup locations corresponding to affordable housing at the periphery of the geo-fenced area connected to destination locations corresponding to jobs and other locations of opportunity. The presented work showcases SAV operation as a solution to the spatial mismatch between affordable housing and job locations in a realistic simulation environment in an urban setting.

Introduction

Automated driving algorithms and their robust controls [1] and the associated possibility of improving fuel/energy economy [2] are the natural results of decades of research and development starting with active safety systems and Advanced Driving Assistance Systems (ADAS) [3], [4], [5], [6], [7]. These developments have made autonomous taxis and shuttles a feasible possibility. Current taxi systems have adapted to models that allow for optimization of passenger cost and comfort. With the introduction of autonomous taxis and/or shuttles, a new opportunity to further optimize this structure presents itself. Particularly, vehicles will no longer be tied to human driver restrictions. Vehicles will be able to run for longer periods of time. In fact, projection studies indicate that with the introduction of Autonomous Vehicles, there will be a reduction on the number of privately owned vehicles along with an increase in the number of vehicles on the road at any moment [8]. With these new opportunities, we can plan to improve road safety, optimize the land needed for transport, mitigate manpower constraint for bus services and enhance mobility for underserved communities including the elderly and disabled [9]. In this paper, our focus is to optimize the necessary resources for transport by providing a simulation environment that allows us to understand the different ways that can be used to maximize the number of shared miles and provide lower-resource communities an affordable mobility choice for access to locations of opportunity.

The current literature has established that utilizing Autonomous Vehicles (AV) as the main transport service cannot be supported in general by balancing costs for passengers and AV owners alone. These studies utilize known routing algorithms such as Dijkstra's shortest path algorithm [10] to complete their tasks or impose restrictions on the paths of the AVs [11]. Further, the current consensus among experts is that with the current approach to AVs, these systems do not make economic sense and even in comparison to systems with higher economic operation costs, current AV based mobility solutions only slightly improve travel time savings [11]. Thus, we are in need of a flexible simulation system that will allow us to change the back-end of the simulation process such that we can try different dispatching and routing algorithms to improve the efficiency of these systems to make them feasible to operate.

This paper is divided into two main sections. The first introduces our simulation environment and the second discusses a case scenario. The discussion on our simulation environment begins with the software chosen to create such simulation environment as well as the workflow between each of the necessary components. Then, we discuss the mechanics behind simulating SAVs and the parameters chosen for such task. We, then, present a simple model for SAV management and deployment as well as discuss some improvements over the current SAV deployment approaches. Finally, we conclude with the overall process to generate road networks for SAV deployment.

The second part of the paper presents a closer look into incorporating the simulation environment into a real-life example. For this, the city of Marysville, Ohio is chosen. We first discuss the general trends in the city and the modeling of such. Then, we provide the empirical model of traffic creation and trip distribution. We then conclude with some remarks.

Modeling Shared Autonomous Vehicles

Software

Simulation of ride hailed Shared Autonomous Vehicles is not a standard feature of common software. Thus, the following study requires software development outside the scope of regular simulation software. We develop a framework in which we define our own. The microscopic traffic simulation software PTV Vissim is used to conduct the main network simulation along with developed add-on modules to create the necessary behavior simulated. Vissim provides options to control the behavior of the simulation by using the Python COM interface.

Workflow

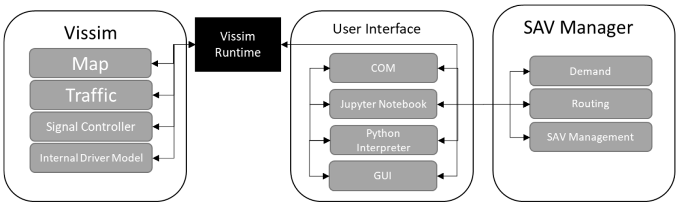


Figure 1. Data Flow between different simulation components.

The workflow of the created simulation environment shown in Figure 1 takes advantage of the COM interface that PTV Vissim provides. There are two kinds of extensions on the Vissim Runtime, one which manages the SAVs and one which manages the traffic flow. There are two ways in which to implement these extensions. One is implemented as an event-based script and the other is implemented as a complete external script. The event-based scripts are faster but provide less flexibility while the external scripts provide great simulation flexibility at the cost of reduced simulation speed.

The overall SAV Manager is an external library extension written in Python. The extension is written using Object Oriented Programming which compounds with the external scripting to reduce the cost of simulation speed. The details of the overseeing algorithm will be discussed in the next section. On the other hand, the traffic flow is controlled by an external script which dictates the behavior of traffic depending on the time of the day.

Simulating SAVs

SAVs are modeled as an extension of Autonomous Vehicles. We base the parametrization of the SAVs on the CoEXist project [13]. The CoEXist project aimed to simulate different levels of autonomous vehicles in a microscopic traffic simulation environment. The parametrized levels of autonomy differ from the standard SAE guidelines and are broken down into the categories of Basic, Intermediate and Advanced. Each of these categories is parametrized by driving logics. The developed driving logics for this purpose were named rail safe, cautious, normal, and all-knowing. While these logics are modeled by altering default Wiedemann 99 parameters, authors also model how different levels of AVs handle lane changing, car following, and gap acceptance at intersections.

In this paper, we model SAVs as overseen by an agent which assigns and controls destinations while the driving behavior is modeled with a Wiedemann 99 driver model fine-tuned to replicate the "intermediate" AV model. The parameters for the AV driver are listed in Tables 1 and 2 and obtained from the CoEXist project [13]. The standard simulation workflow of this process is outlined in Figure 2.

Road Type	Intermediate
Motorway	Normal
Arterial	Cautious
Urban Street	Cautious
Shared Space	Manual

Table 1. Driving behavior assignment for intermediate level AVs.

Wiedemann 99 Parameter	Cautious	Normal
CC0 – Standstill distance (m)	1.5	1.5
CC1 – Spacing time (s)	1.5	0.9
CC2 – Following variation (m)	0	0
CC3 – Threshold for entering "following" (s)	-10	-8
CC4 – Negative "following" threshold (m/s)	-0.1	-0.1
CC5 – Positive "following" threshold (m/s)	0.1	0.1
CC6 – Speed dependency of oscillation ($10^{-4}\ rad/s$)	0	0
CC7 – Oscillation acceleration(m/s^2)	0.1	0.1
CC8 – Standstill acceleration (m/s^2)	3	3.5
CC9 – Acceleration at 80 km/h (m/s^2)	1.2	1.2

Table 2. Wiedemann '99 parameters necessary for intermediate level AVs.

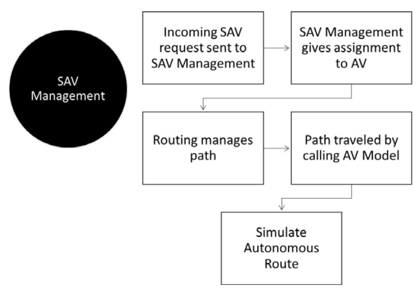


Figure 2. SAV simulation logic: An extension from AV parameters.

By modifying the shown parameters in Table 2, different autonomy levels of AVs are simulated in the presented work. The higher-level sharing task of the selected AVs is managed by the developed SAV Management module. The general logic for SAV behavior is the following: A set number of SAVs are spawned at the beginning of the simulation. The SAVs obtain an incoming request from a selected trip generation model. The SAV Manager application is the overseeing entity on handling these requests. Once an incoming SAV request is communicated, the SAV chooses an appropriate taxi to complete this request. More details on the selection process are highlighted under the next section. The SAV is also given a selected route to follow as assigned by the routing module of the simulation. Then, the SAV follows this path as an AV.

Optimizing multiple AVs

Maximizing the number of miles shared per passenger lies in the overarching SAV manager. We take advantage of the behavior of the SAV as it idles, waiting for a request. We showcase the differences in an SAV which idles per single trip versus an SAV that concentrates on minimizing the number of miles traveled while still completing its necessary tasks.

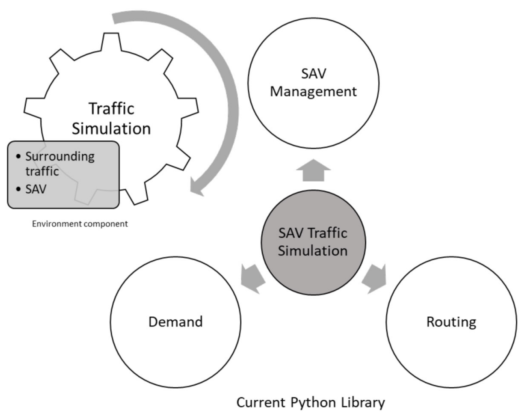


Figure 3. Different module components behind the modeling of SAVs.

First, we explain the design of the add-on Python extension. The Python extension is composed of three classes as shown in Figure 3. As the simulation runs, the shuttles are updated with each time step through the event-based triggered script. The relevant objects are updated and passed to the SAV Manager. The relevant information is delegated through the COM interface which is then delegated to the corresponding classes. The main classes of the simulated SAV Manager consist of the SAV management, the routing manager and the demand simulation manager.

The demand simulation manager handles the creation of trip requests and mitigates this information to the SAV manager. The SAV manager can deploy the necessary SAV according to an optimal choice. The SAV manager distinguishes between two ways of deploying the SAVs, idling vs. sharing behavior. When idling behavior is activated, the taxi prioritizes taking passengers to their destinations and returning to base before taking up a new request. On the other hand, when the sharing behavior is activated the SAV's priority lies on completing trips. That is, as each trip is completed the SAVs are assigned trips using the pseudo distance between the SAV and the incoming trips. If there are no new incoming trip requests, the SAVs are sent to the area with the last most generated trips. This logic is outlined in Figure 4.

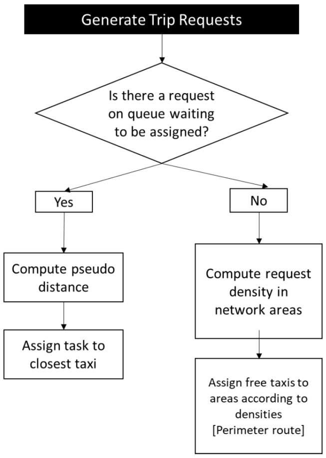


Figure 4. Decision making of the SAV Manager trip request assignment module.

In order to explain our algorithm for deploying SAVs optimally we first establish how to represent the network. For this, we use the standard notation of directed graphs. We use this notion in order to establish the distances between stops and SAVs as number of nodes traversed.

Definition 1. (Directed Graph) A directed graph $\boldsymbol{G}$ is an ordered pair of vertices $\boldsymbol{V} = \{\boldsymbol{v}_i\}$ and edges $\boldsymbol{E} = \{\boldsymbol{e}_j\}$.

The graph is denoted by $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$. Note that each edge is simply a tuple of vertices $\boldsymbol{e}_j = (\boldsymbol{v}_{j_1}, \boldsymbol{v}_{j_2})$ that indicate a path from vertex $\boldsymbol{v}_{j_1}$ to vertex $\boldsymbol{v}_{j_2}$. In this way, we represent streets as edges and intersections as nodes. Further, each vertex is simply a tuple that represents a point in 2D space, i.e. $\boldsymbol{e_j} = (\boldsymbol{x}, \boldsymbol{y})$. The network is thus represented as a directed graph $\boldsymbol{G}$.

Next, we compute the location of a particular SAV on the network indexed by the SAV manager as SAV $\boldsymbol{i}$. Denote the $\boldsymbol{i}$th SAV and its location on the 2D plane by $\boldsymbol{s_i} = (\boldsymbol{x}, \boldsymbol{y})$.

Assume that $\boldsymbol{s_i}$ is placed along edge $\boldsymbol{e_j}$ then the distance between $\boldsymbol{s_i}$ and $\boldsymbol{e_j}$ is defined as $\left\|\boldsymbol{s_i} - \boldsymbol{v_{k_1}}\right\|_2$ as shown in figure 5.

Figure 5. Visual representation of the computation of the distance between an edge $\boldsymbol{e_j}$ and an SAV $\boldsymbol{s_i}$.

The location of the SAV $\boldsymbol{i}$ in the network graph representation $\boldsymbol{l(s_i)}$, is defined as

$$\boldsymbol{l(s_i)} = \mathbf{argmin}_{e_j}\, \boldsymbol{d(s_i, e_j)} \quad (1)$$

In short, the location is denoted by the closest edge to the shuttle. Similarly enumerate the SAV stops and define the location of an SAV stop $\boldsymbol{s_l}$ as the closest edge to the stop as above.

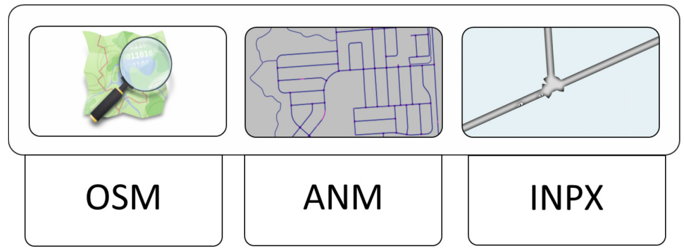
The pseudo distance between SAV $\boldsymbol{i}$ and SAV stop $\boldsymbol{j}$ is defined as the minimum number of nodes through which a vehicle must pass to get from $\boldsymbol{l(s_i)}$ to $\boldsymbol{l(s_j)}$. To obtain an optimal choice of assignment, the SAV Manager assigns trips to the closest (in terms of pseudo distance) SAVs.

Finally, the routing module provides information on how the SAVs should proceed to their next destination.

Further, we are also concerned with maximizing the total number of shared miles. This can be done by making a distinction between shared and single-purpose rides and prioritizing shared rides. In single-purpose rides, we allow passengers to request rides from a location to the next without expecting to stop along the way. This is typical taxi behavior and is non-optimal. The sharing behavior allows for SAVs to be assigned requests at the time of trip of a passenger, thus maximizing the number of miles shared.

Building the Network

The network is built with the help of OpenStreetMap (OSM). First, the network is exported as an osm file and is then converted into an anm file. Finally, this is transformed into a Vissim .inptx file. This process is illustrated in Figure 6.



The inpx file serves as a blueprint for the relevant streets and intersections. In this case-study, we do not concern ourselves with traffic lights, leaving that to future work. Thus, we only further fine-tune the network by repairing intersections and routing decisions. These are all done by hand and we concentrate on the major freeways and highways. These are determined by the traffic generation numbers as discussed later.

Figure 7. Simulated Marysville Network.

Sample Network

Marysville

The City of Marysville, Ohio whose road network is shown in Figure 7 is chosen as a network model. Based on data from the US Census [14], the flow of traffic in the city is characterized by the movement of people shifting from the outskirts of the city towards the center for access to work or resources. To facilitate the simulation of the Marysville network, the city was first divided into its 13 different census block sections to characterize different features of the city as shown in Figure 8.

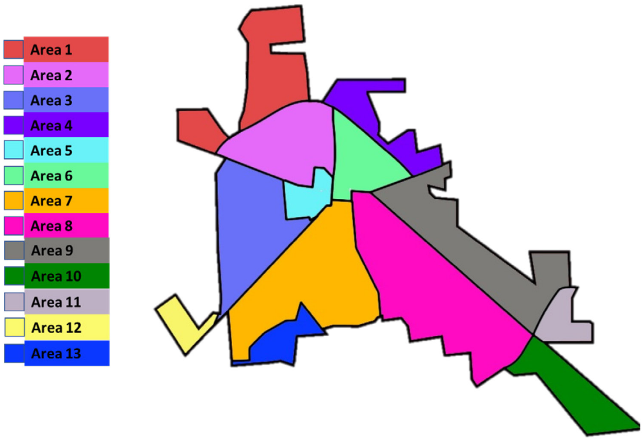


Figure 8. Marysville divided into 13 different areas as indicated in [14].

Traffic Generation

The traffic generation of this network is acquired from empirical samples of data and extrapolated inferences from national averages. From the Ohio Department of Transportation (ODOT) website [16], we can find information regarding general average traffic counts per day as shown in Figure 9.

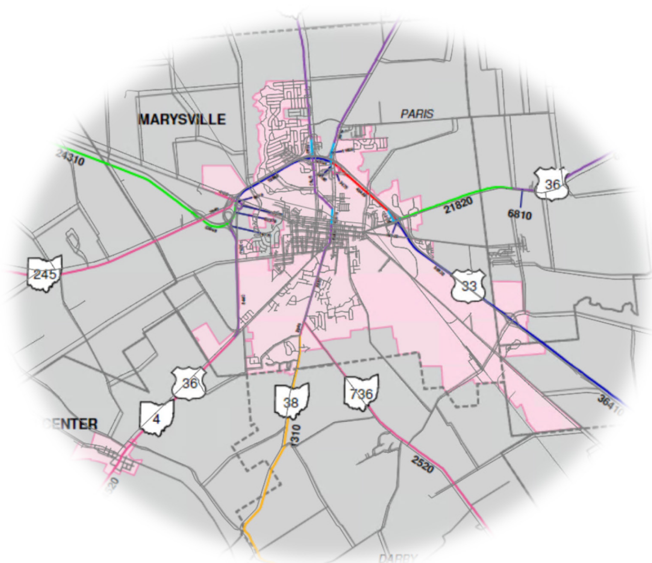


Figure 9. Average influx of vehicles on major highways, freeways and roads [15].

The average daily traffic counts are extrapolated to simulate hourly traffic counts by following Ohio Urban trends given by the ODOT [16]. We take the main highways as main sources of traffic influx. The traffic volume per hour V_h at time h becomes

$$V_h = T_v \cdot P_h \qquad (2)$$

where T_v is the total average volume per day and P_h is the percentage traffic volume that takes place on average at time h. We are, thus, able to generate hourly traffic for the network as shown in Figure 10.

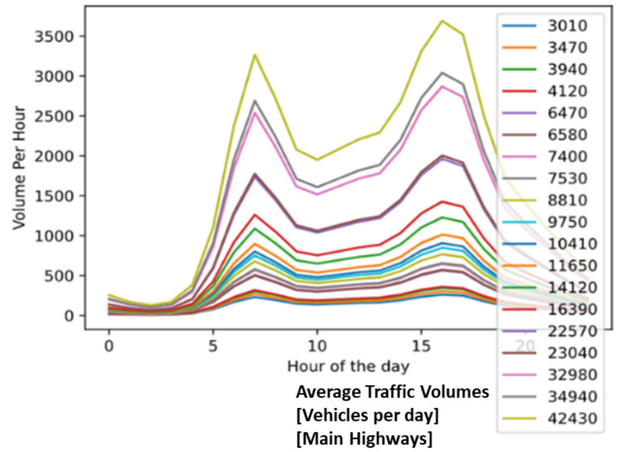


Figure 10. The legend shows the traffic volume in major roads and highways in Marysville. Meanwhile the graph shows the traffic volume per hour of the day

Trip Requests

The next step in our simulation set-up is to model trip requests which will be a part of the backend process of the demand module in our SAV simulation. We do this empirically. In a 2012 study [17], a correlation between household family income and average number of daily trips was established. The general trend showed a positive correlation between the number of average trips and household income. The Marysville network areas were sampled for average household income using US Census Data [14]. Based on these assumptions we obtain a number T_I for the number of trips per selected zone. Further, we must approximate the percentage of trips out of the total number of trips that would be done using an SAV. For this, in [8] we find an approximated distribution from a telesurvey which allows us to approximate the compositions of future trips per hour of the day P_h. The number of trip requests $T_{z,h}$ is then dependent on the Zone z and hour of the day h selected and given by

$$T_{z,h} = T_I \cdot P_h \qquad (3)$$

The general trend given by this model is showcased in Figure 11.

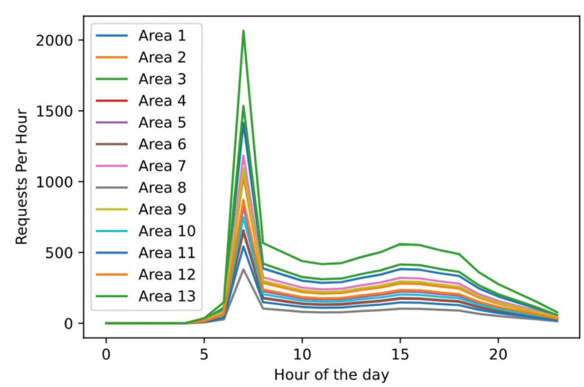


Figure 11. Number of trip requests per hour of the day.

Thus, we have the average number of trips per household per family income and the composition of trips. From this information, we may infer total trip requests per hour per zone emanating from our previously chosen zones. The next step in our model is to assign trip request locations and bases. Parking spaces are placed in each zone and assigned a probability from a Uniform distribution in the [0,1] interval. This probability signifies the randomness within each of these zones. The parking lot area and SAV stops are designed to allow the flow of people from the permiter of Marysville to the center where the concentration of schools, shops and jobs lie as shown in Figure 12.

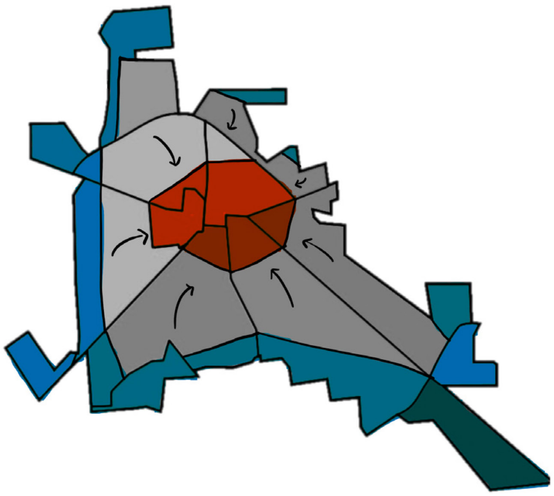


Figure 12. SAV stop targets.

We now have a full model of the Marysville network with traffic generation and SAV integration that is dynamic with the time of day.

Simulation Results

The SAVs were simulated inside the Marysville network and compared against a baseline Marysville network (without the addition of SAVs). The traffic distribution from afternoon peak hour was chosen as a comparison point and randomly initated traffic was set for verifying the simulation results by repeatedly running the simulation with different random seeds. Figure 13 shows the traffic volume distribution after the network populates and is run for an hour of simulated time. In this network, we see little movement from the outskirts of the network and most movement throughout the main highways and the interior of the network. On the other hand, the introduction of 10 SAVs shows more frequent movement from the perimeter of the Marysville area into the interior of Marysville. Yet, this movement also shows an increase in traffic jams throughout the network. This is due to placement of pick-up/drop off locations, indicating that such placement must be optimally chosen in order to avoid disturbances in traffic. This is the next chosen direction of this research; the developed tool introduced in this paper will be used to further optimize the placement of SAVs.

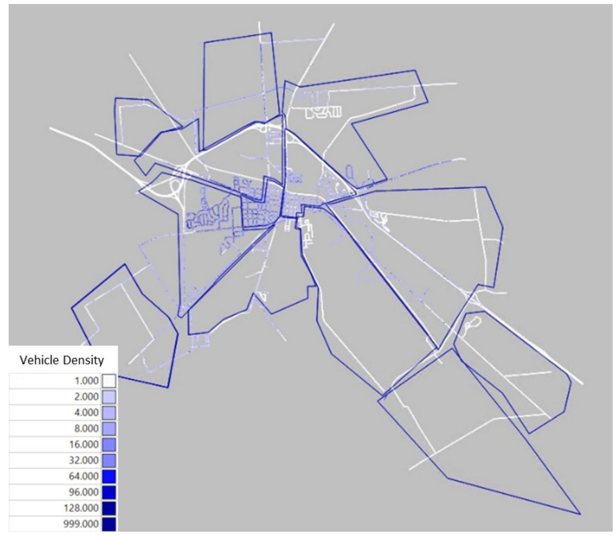


Figure 13. Marysville Network traffic density without the introduction of SAVs.

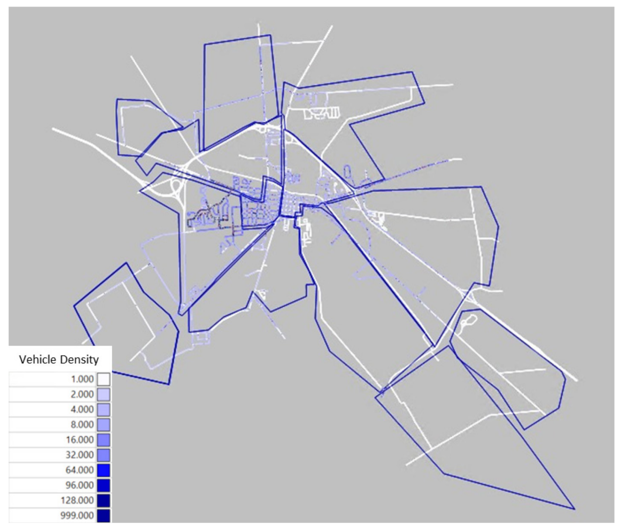


Figure 24. Marysville Network traffic density after the introduction of SAVs.

Summary/Conclusions

Marysville, Ohio is undergoing a transformation to provide individuals with access to more resources. Part of this transformation lies in enabling its residents to reach destinations to attend school, shop and obtain better jobs. Since there is no public transport in Marysville and since a large portion of the population fo not own or operate private vehicles, the logical solution is to use an SAV service. This makes Marysville a perfect candidate to showcase the effects of the introduction of SAVs in a traffic network with respect to network flow and movement of people. Thus, we have presented a way to simulate the introduction of said SAVs. Our approach starts with the theorical part of the simulation environment; coding in the necessary software to model behavior. Then, we use this in addition to different driving parametrizations to realize a realistic SAV behavior. The engine behind trip requests and traffic modeling is empirically based and allows us to be flexible with the specifications behind these. Using data, we are able to model an SAV manager and optimize the dispatching of trips. Thus, we have created a realistic testbed for SAVs. In our future work, we seek to implement more features such as more sophisticated trip management and dynamic routing as well as to create a co-simulation environment with autonomous vehicle dynamics [18], [19], [20], [21]. We seek to use this simulation environment to aid the engineering of SAVs.

Contact Information

Karina Meneses-Cime
menesescime.1@osu.edu
Automated Driving Lab, Ohio State University
1320 Kinnear Rd., Columbus, OH, 43212

Acknowledgments

The Ohio State University authors would like to thank Ford Motor Company for partial support of this work. The Ohio State University authors acknowledge the support of this work in part by the National Science Foundation under Grant 2042715.

References

1. Guvenc, L., Aksun- Guvenc, B., Demirel, B., Emirler, M.T., Control of Mechatronic Systems, the IET, London, ISBN: 978-1-78561-144-5, 2017.
2. Boyali A., Guvenc, L., “Real-Time Controller Design for a Parallel Hybrid Electric Vehicle Using Neuro-Dynamic Programming Method,” IEEE Systems, Man and Cybernetics, Istanbul, October 10-13, pp. 4318-4324, 2010.
3. Aksun-Guvenc, B., Guvenc, L., Ozturk, E.S., Yigit, T., “Model Regulator Based Individual Wheel Braking Control,” IEEE Conference on Control Applications, Istanbul, June 23-25, 2003.
4. Aksun-Guvenc, B., Guvenc, L., “The Limited Integrator Model Regulator and its Use in Vehicle Steering Control,” Turkish Journal of Engineering and Environmental Sciences, pp. 473-482, 2002.
5. Oncu, S., Karaman, S., Guvenc, L., Ersolmaz, S.S., Ozturk, E.S., Cetin, A.E., Sinal, M., “Steer-by-Wire Control of a Light Commercial Vehicle Using a Hardware-in-the-Loop Setup,” IEEE Intelligent Vehicles Symposium, June 13-15, pp. 852-859, 2007.
6. Emirler, M.T., Kahraman, K., Senturk, M., Aksun-Guvenc, B., Guvenc, L., Efendioglu, B., “Two Different Approaches for Lateral Stability of Fully Electric Vehicles,” *International Journal of Automotive Technology*, Vol. 16, Issue 2, pp. 317-328, 2015.
7. Kural, E., Aksun-Guvenc, B., “Model Predictive Adaptive Cruise Control,” IEEE International Conference on Systems, Man, and Cybernetics, Istanbul, Turkey, October 10-13, 2010.
8. Deloitte, “Urban Mobility and Autonomous Driving in 2035.”
9. Cheon Kheong Tan and Kwang Sheun Tham. Autonomous Vehicles, Next Stop: Singapore. Journeys, (November):5–11, 2014.
10. Horl, S. “Implementation of an autonomous taxi services in a multi-model traffic simulation using MATSim,” Master thesis in Complex Adaptive Systems. Chalmers University of Technology.
11. Leich, Gregor, and Joschka Bischoff. "Should Autonomous Shared Taxis Replace Buses? A Simulation Study." *Transportation Research Procedia* 41 (2019): 450-60. doi:10.1016/j.trpro.2019.09.076.
12. Bischoff, Joschka, and Michal Maciejewski. "Simulation of City-wide Replacement of Private Cars with Autonomous Taxis in Berlin." *Procedia Computer Science* 83 (2016): 237-44. doi:10.1016/j.procs.2016.04.121.
13. Peter Sukennik (PTV Group), “Micro-Simulation Guide for Automated Vehicles,” October 31, 2019, [Online], available at: https://www.h2020-coexist.eu/resources/, accessed October 25, 2020.
14. U.S. Census Bureau. Income in the past 12 months (In 2018 inflation-adjusted dollars). Prepared by Social Explorer.
15. “Union County Annual Average Daily Traffic 2012.” Union County Traffic Count Reports, 2012, www.dot.state.oh.us/Divisions/Planning/TechServ/traffic/Pages/Up-to-Date-Counties/Union.htm.
16. “Hourly Percent by Vehicle Type.” Ohio DOT. http://www.dot.state.oh.us/Divisions/Planning/TechServ/traffic/Pages/HrlyPercent.aspx.
17. Olmo Sánchez, M Isabel, Elvira Maeso González, Travel Patterns, Regarding Different Activities: Work, Studies, Household Responsibilities and Leisure, *Transportation Research Procedia*, Volume 3, 2014, Pages 119-128, ISSN 2352-1465, https://doi.org/10.1016/j.trpro.2014.10.097.
18. Emirler, M.T., Uygan, I.M.C., Aksun-Guvenc, B., Guvenc, L., 2014, “Robust PID Steering Control in Parameter Space for Highly Automated Driving,” *International Journal of Vehicular Technology*, Vol. 2014, Article ID 259465.
19. Emirler, M.T., Wang, H., Aksun-Guvenc, B., Guvenc, L., 2015, “Automated Robust Path Following Control based on Calculation of Lateral Deviation and Yaw Angle Error,” ASME Dynamic Systems and Control Conference, DSC 2015, October 28-30, Columbus, Ohio, U.S.
20. Wang, H., Tota, A., Aksun-Guvenc, B., Guvenc, L., 2018, “Real Time Implementation of Socially Acceptable Collision Avoidance of a Low Speed Autonomous Shuttle Using the Elastic Band Method,” *IFAC Mechatronics Journal*, Volume 50, April 2018, pp. 341-355.
21. Zhou, H., Jia, F., Jing, H., Liu, Z., Güvenç, L., 2018, “Coordinated Longitudinal and Lateral Motion Control for Four Wheel Independent Motor-Drive Electric Vehicle,” *IEEE Transactions on Vehicular Technology*, Vol. 67, No 5, pp. 3782-379.