

A Unified Architecture for Scalable and Replicable Autonomous Shuttles in a Smart City

Sukru Yaren Gelbal

The Ohio State University
Center for Automotive Research
Department of Electrical and Computer Engineering
gelbal.1@osu.edu

Haoan Wang

The Ohio State University
Center for Automotive Research
Department of Electrical and Computer Engineering
wang.6184@osu.edu

Nitish Chandramouli

The Ohio State University
Center for Automotive Research
Department of Mechanical and Aerospace Engineering
chandramouli.11@osu.edu

Bilin Aksun-Guvenc

The Ohio State University
Center for Automotive Research
Department of Mechanical and Aerospace Engineering
aksunguvenc.1@osu.edu

Levent Guvenc

The Ohio State University
Center for Automotive Research
Department of Mechanical and Aerospace Engineering
Department of Electrical and Computer Engineering
guvenc.1@osu.edu

Abstract— Connected and automated driving vehicles have become a present and near future reality. There are already a slowly increasing number of trucks that do autonomous test drives on highways. Level 3 Highway Chauffeur applications are expected to be on the market as early as 2020 and Level 4 Highway Pilot applications are expected to be available around 2025. There is also a drive towards smarter cities with smarter mobility choices that include the use of smaller, lower speed automated driving vehicles. This paper proposes a unified basic computing, sensing, communication and actuation architecture and scalable and replicable automated driving control systems built upon that architecture. A passenger sedan and a small electric vehicle are used to illustrate the application of the proposed unified architecture to these two different sized vehicles. Their automation along with a scalable control algorithm for path following are used for presenting the scalability and replicability of the unified architecture proposed in this paper.

Keywords—*automated driving, scalable and replicable control, unified architecture, path following, parameter space control*

I. INTRODUCTION

Along with the traditional Original Equipment Manufacturers (OEM), software based technology companies and start-up companies have also entered the market of autonomous driving. OEMs and software based technology companies develop proprietary solution architectures and algorithms through big investments in research in this area. On

the other hand, start-up companies work mostly on niche markets like low speed autonomous shuttles operating on a fixed route and also spend most of their relatively limited resources on research and product development. These fixed route autonomous shuttles are currently available and are used in several cities around the world. Regardless of the type of company, most of the research efforts involve parallel and redundant that will benefit from a common, unified architecture and a scalable and replicable solution approach. As more companies are driving towards Level 4 automated driving and as series production is expected in the near future, testing of autonomous vehicles for certification is also going to be a significant near future necessity. The required testing and certification phase will benefit from the use of a standardized architecture with scalable and replicable building blocks. This paper reports our initial work mainly aimed at low speed autonomous shuttles using a simple, unified main architecture with emphasis on interoperability, scalability and replicability. As such, the automation and automated path tracking application of our automated driving sedan is carried over to our low speed autonomous shuttle to demonstrate the effectiveness and usefulness of a common unified architecture that allows the use of replicable and scalable control systems.

Development of a unified architecture has already been recognized as a potential area of progress towards making large scale deployment of autonomous vehicle deployment a near reality. Behre and Törngren [1] have worked on developing a

functional reference architecture where a solution has been proposed which will work without dependence on specific technologies. They have divided the architecture into three categories as: Perception, Decision Control and Vehicle Platform Manipulation. They were also able to apply their results in making autonomous vehicles in various applications with continuous refinement. Ines et al [2] have worked on low speed control of an autonomous vehicle that was restricted to only longitudinal control. They used an analog signal for automatic throttle with a switch to change to the original throttle signal which is very similar to the implementation in this paper. Their brake actuation was realized by control valves going to the Anti-lock Braking System (ABS) via Controller Area Network (CAN). This paper uses a more conservative approach due to the small electric vehicle lacking an ABS system. They used a fractional PI controller for following a specific drive cycle. Yoshizawa et al [3] have worked on a path following control of low speed autonomous vehicle using only a Light Detection and Ranging (LIDAR) sensor. They have used LIDAR data to detect the curbs of the road and generate a digital map on which a preview lateral control was implemented.

The references [1-3] have each developed their own different architecture and have not treated the issue of replicating or scaling their architecture for different vehicles and different levels of autonomy. This paper proposes a unified architecture with scalable and replicable control algorithms for a more standardized approach to developing autonomous vehicles. This unified approach is a natural extension of the earlier work of some of the authors presented in [4-5].

The organization of the rest of the paper is as follows. The unified architecture and replicability is presented in section II where the two automated driving vehicles, a hybrid electric sedan and a fully electric two seater shuttle, in this paper are explained with respect to how they were converted to autonomous driving vehicles and how their autonomous functions were built. Section III is on the lateral model of the vehicles used and also treats the automated path determination and path following application. Section IV is on the design of the steering controller in parameter space. Using parametric models for both vehicles, the automated path following steering controller for the passenger sedan is scaled down and replicated for the smaller electric shuttle. Section V presents model-in-the-loop simulation and road test results that are used to demonstrate the effectiveness of this proposed scaling and replication approach in the context of the automated path following application. A simple evaluation and rating system is also proposed and used in evaluating and comparing the simulation and experimental automated path following results. The paper ends with conclusion in the last section.

II. UNIFIED ARCHITECTURE AND REPLICABILITY

Two different sized vehicles are automated using the same unified architecture shown in Fig. 1 in this paper. One of these vehicles is a Ford Fusion Hybrid SE sedan which was automated first. The other vehicle which was automated later using the architecture in the sedan is a small, low speed full electric two seater shuttle used for ride sharing applications (Dash EV). The architecture and replication/scaling method presented in this

paper is general in nature and will be applied to other vehicles also in our future work.

The throttle, steering and brake functions of the vehicles are first converted into drive-by-wire actuators that receive CAN bus commands from an electronic control unit that handles all low level command, control and basic decision making functions. Localization and perception sensors and a separate computer are added to achieve environment perception and autonomous driving capability as shown in Fig. 1.

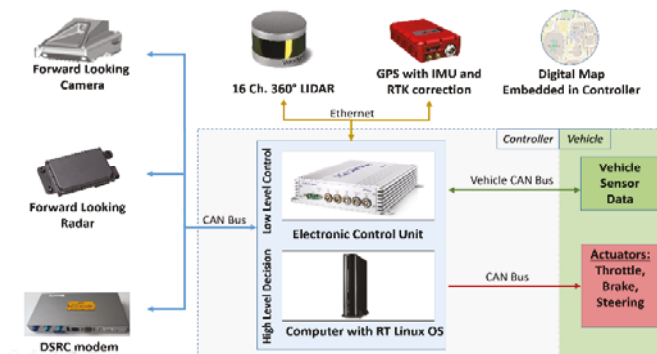


Fig. 1. General unified architecture used for automation.

The sedan shown in the top part of Fig. 2 is converted into drive-by-wire by using a commercially available CAN bus solution that allows us to use the existing steering, brake and throttle actuators in the vehicle. This is the preferred approach in automating a vehicle for drive-by-wire functionality as the existing actuators are already built for automotive standards on durability, maintenance, fault diagnostics and fault tolerance. The low speed autonomous shuttle is shown in the bottom parts of Fig. 2. Like most low speed shuttles, it was not built with throttle, steering and brake actuators with a CAN interface. Drive-by-wire functionality was, therefore, added by the authors through a smart electric motor connected to the steering rack, a linear electric motor that pulls the brake pedal and an electronic by-pass circuit that reads the accelerator pedal potentiometer and changes it with the automated driving system supplied value. These three actuators all have a CAN bus interface just like the sedan (see Fig. 1).

After achieving drive-by-wire capability on both vehicles, sensors were added to achieve localization and environmental perception for situational awareness. These sensors include forward looking radar, three dimensional LIDAR, forward looking camera, high accuracy Global Positioning System (GPS) and a Leddar (cheap solid state LIDAR) sensor. The forward looking radar is used for detecting vehicles in front of the ego vehicle while driving autonomously on a highway. It is used for car following and emergency braking applications. The three dimensional LIDAR sensor gives 360° and 16-channel point cloud data which can be used for numerous applications such as Simultaneous Localization and Mapping (SLAM), object classification and obstacle avoidance. The Leddar sensor is a solid-state LIDAR which we use to get information about the obstacles in front of the vehicle. It provides obstacle distance data within a 95° field of view. Our GPS sensors are differential units with Real-Time Kinematic (RTK) capability with about 2 cm accuracy when we use our own base and 5 cm accuracy when

the State of Ohio broadcasted corrections are used. It provides heading even while the vehicle is stationary due to the use of differential antennas. We also use a navigation grade lower accuracy GPS in both vehicles. The sedan uses a Mobileye camera with on-board processing which detects vehicles and lanes while driving autonomously and also provides time-gap information for the target vehicle. Both vehicles use standard cameras with raw data output. These cameras use our own lane detection and vehicle detection algorithms which use several methods ranging from edge detection to deep learning.

In addition to a unified implementation of the sensors and actuators, we use the same main control and decision making architecture for both vehicles. All low level control and decision making algorithms run on a dSpace Microautobox electronic control unit. This unit runs a Simulink model and has numerous Input/Output (I/O) ports and means of data transfer which makes it capable of connecting to and operating with almost all types of sensors and actuators. Automatic code generation is used to seamlessly convert the Simulink automated driving block diagram to code that runs on the Microautobox. This approach improves the time spent on algorithm development and testing considerably. The generated code can later easily be embedded in a series production level electronic control unit at the end of the research and development phase.

Sensors send data to the Microautobox electronic control unit with a means of communication specific to the sensor, like CAN or User Datagram Protocol (UDP) for most of our sensors. This data is fed to controllers running within the device. Controllers are created in the Simulink and outputs of the controllers are connected to output blocks that correspond to I/O ports of the Microautobox. These I/O ports are physically connected to actuators or drivers of actuators to provide reference signal and achieve autonomous driving. This is illustrated in the diagram of Fig. 3 with our low speed electric shuttle components. Our vehicles also have Dedicated Short Range Communication (DSRC) modems to communicate with other vehicles, infrastructure and pedestrians with DSRC enabled smartphones. For Vehicle to X (V2X) communication, all messages are sent using the standard messages of the Society of Automotive Engineers (SAE) J2735 DSRC Message Set and use the standard communication rate of 10Hz.

III. VEHICLE MODEL AND PATH FOLLOWING

A. Single Track Model

A single track model is used for robust path following controller design. The linear path following vehicle model shown in Fig. 4 can be expressed in state space form as

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \\ \Delta\dot{\psi} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ V & l_s & V & 0 \end{bmatrix} \begin{bmatrix} \beta \\ r \\ \Delta\psi \\ y \end{bmatrix} + \begin{bmatrix} b_{11} & 0 \\ b_{21} & 0 \\ 0 & -V \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_f \\ \rho_{ref} \end{bmatrix} \quad (1)$$

where β , r , V , $\Delta\psi$, l_s and y are vehicle side slip angle, vehicle yaw rate, vehicle velocity, yaw angle relative to the desired path's tangent, the preview distance and lateral deviation from



Fig. 2. Implementation of sensors and electronic control unit in both vehicles.

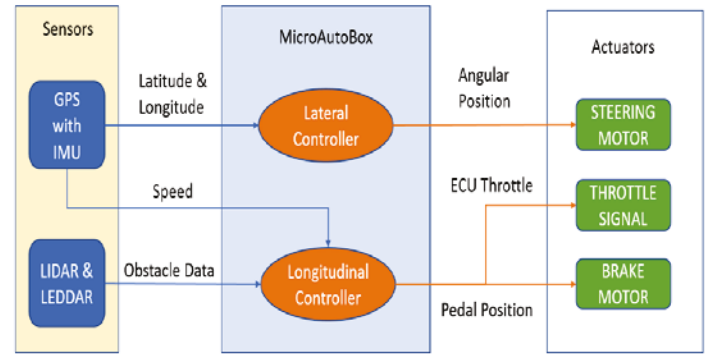


Fig. 3. Dash electric vehicle autonomous system elements and operation.

the desired path at the preview distance, respectively. The control input is the steering angle δ_f . $\rho_{ref} = 1/R$ is the road curvature where R is the road radius of curvature. The remaining terms are

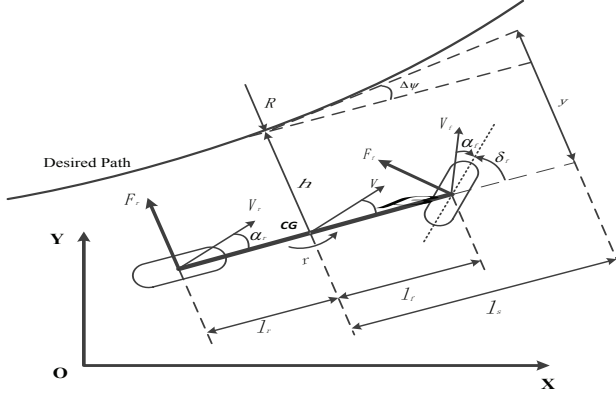


Fig. 4. Diagram of the single track model.

$$a21 = (C_r l_r - c_f l_f) / J, \quad (4)$$

$$a22 = -(c_r l_r^2 + c_f l_f^2) / J V^2, \quad (5)$$

$$b11 = c_f / m V, \quad (6)$$

$$b12 = c_f l_f / J \quad (7)$$

where m is the vehicle mass, J is the yaw moment of inertia, μ is the road friction coefficient, c_f and c_r are the cornering stiffnesses, l_f is the distance from the center of gravity of the vehicle (CG) to the front axle and l_r is the distance from the CG to the rear axle.

B. Offline Path Data Generation

This sub-section and the next present the automated path following algorithm in [6] which has been scaled and applied to the sedan (Fusion) and low speed shuttle (Dash EV) vehicles presented in this paper. The path following algorithm employs a pre-determined path for the vehicle to follow which is derived from the GPS data points collected from either manual driving or are extracted automatically from a map. These data points are then divided into smaller segments with equal number of data points for ease of formulation. These segments of data points formed are used to fit a third order polynomial as

$$X_i(\lambda) = a_{xi} \lambda^3 + b_{xi} \lambda^2 + c_{xi} \lambda + d_{xi} \quad (8)$$

$$Y_i(\lambda) = a_{yi} \lambda^3 + b_{yi} \lambda^2 + c_{yi} \lambda + d_{yi} \quad (9)$$

where i represents the segment number. The reason behind fitting the data points is to be able to replicate the curvatures in the path effectively while discarding the noise in the GPS data points and providing a smooth transition from one segment to another by satisfying the zeroth and first order continuity in X and Y given by

$$X_i(1) = X_{i+1}(0) \quad (10)$$

$$Y_i(1) = Y_{i+1}(0) \quad (11)$$

The X and the Y points derived from the GPS latitude and longitude data are fit using a single parameter λ , resulting in

$$\frac{dX_i(1)}{d\lambda} = \frac{dX_{i+1}(0)}{d\lambda} \quad (12)$$

$$\frac{dY_i(1)}{d\lambda} = \frac{dY_{i+1}(0)}{d\lambda} \quad (13)$$

In Equations (8) and (9), a_{xi} , b_{xi} ... are the polynomial fit parameters and λ is the variable for the fit which varies across each segment between 0 to 1.

C. Path Following Control

After the generation of path data for the lateral controller to be followed, the high accuracy GPS on the vehicle is able to give the position and heading of the vehicle in real time to calculate the path tracking error. Thus, the location of the car with respect to the desired path is evaluated using two measurements – the shortest distance of the CG of the car from the path and the error between current heading of the vehicle and the required heading of the vehicle based on its position. This approach reduces both oscillations and steady state lateral deviation. In order to find an equivalent distance parameter to add to the first component, a preview distance l_s is defined. The lateral deviation used by the controller becomes

$$y = h + l_s \sin(\Delta\psi) \quad (14)$$

where $\Delta\psi$ gives the net angular deviation of heading of the vehicle from the desired heading and y is the total lateral deviation of the vehicle computed at a preview distance of l_s . Fig. 5 illustrates the lateral deviation calculation of (14). The path following steering controller's aim is to reduce y to zero for perfect path following.

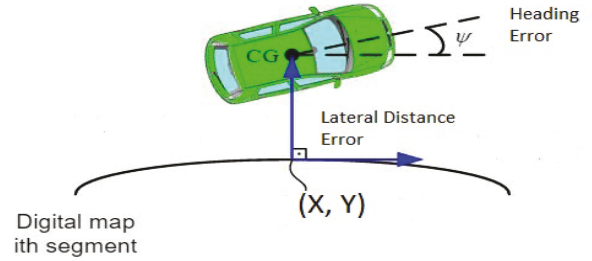


Fig. 5. Error calculation.

To demonstrate the capability of the path data generation algorithm and control, a path following experiment carried out with the sedan vehicle is used. An oval path is generated for the vehicle to follow. Position data and lateral error are recorded

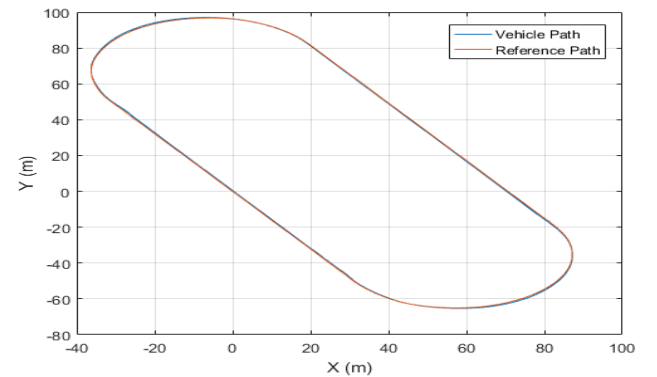


Fig. 6. Path following with Ford Fusion.

while the vehicle is doing autonomous path following of the path shown in Fig. 6.

The sedan (Fusion) is able to follow the oval path accurately using the algorithm as seen in Fig. 6. We can see the reference path and vehicle path are overlapping for most of the path. There is a small error on the curved parts of the path as seen in Fig. 7. The lateral error has an Root Mean Square (RMS) value of 0.121 m.

Fig. 7. Lateral error while autonomous path following.

IV. CONTROLLER DESIGN AND SCALABILITY

A PD steering controller is used for path following. The PD controller gains K_d and K_p are used as the two design parameters in the parameter space approach used here [7]. The D-stability requirements shown in Fig. 8 are used here for guaranteed settling time, damping ratio and bandwidth upper limit. The closed loop system is D-stable when roots of the closed loop characteristic equation lie in the D-stable region in Fig. 8. The controllers used will be improved with additional phase margin and mixed sensitivity bounds in our future work.

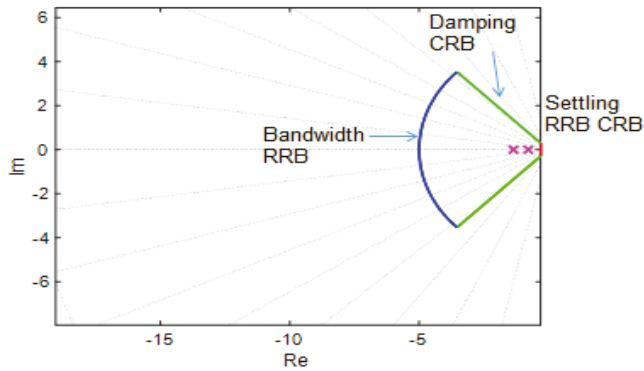


Fig. 8. D-stability region in complex plane.

The sedan path following steering controller designed and presented in our earlier work was used here in obtaining the experimental result of Figs. 6 and 7. This controller had PD gains of $K_p=0.15$ and $K_d=0.35$. For the electric shuttle, the values of the parameters used are $J=350 \text{ kgm}^2$, $l_f=1.09 \text{ m}$, $l_r=0.96 \text{ m}$, $l_s=2 \text{ m}$, $C_f=C_r=18,917 \text{ N/rad}$. D-stability boundaries are formed by choosing settling time constraint σ to be 0.3 sec^{-1} and bandwidth constraint R to be 5 rad/sec . A minimum damping

ratio corresponding to $\theta=45^\circ$ is determined as 0.707 as the other design requirement. From Fig. 8, which shows the location of characteristic equation roots achieved, we can see that two dominant poles of the system are within the D-stable region. The resulting parameter space which satisfies the stability requirements is shown in Fig. 9 where the design point for K_d and K_p is selected as (0.0801, 0.9272) and is marked with a red dot. The model parameters used and the resulting controllers are shown in Table I. The controller design procedure is parameterized with respect to the model parameters such that the Dash (electric shuttle) controller was obtained interactively and very fast using the existing design procedure for the Fusion (sedan)

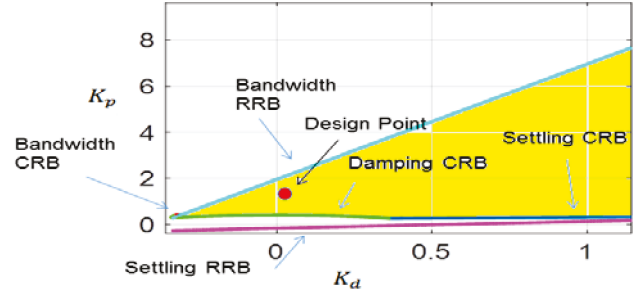


Fig. 9. D-stability solution region.

TABLE I. PARAMETERS AND CONTROLLER COEFFICIENTS

	Ford Fusion	Dash
m	1977.6 kg	350 kg
J	3728 kgm^2	350 kgm^2
l_f	1.3008 m	1.06 m
l_r	1.54527 m	0.96 m
R	0.3225 m	0.24 m
C_f	1.9e5 N/rad	1.8917e4 N/rad
C_r	5e5 N/rad	1.8917e4 N/rad
K_p	0.15	0.9272
K_d	0.35	0.0801

V. EVALUATION

After the controller design, real world experiments were carried out for the purpose of analyzing the effectiveness of both controller parameters and the scaled and replicated design. The experimental results for the Fusion (sedan) were already presented in Figs. 6 and 7. The scaled and replicated path following control system is applied to the Dash (electric shuttle) in another closed path as shown in Fig. 10. An oval path was repeated twice in order to test both performance and repeatability. The first path following experiment is carried out by using the steering controller coefficients designed for the sedan vehicle on the shuttle. The second experiment is carried out with the controller coefficients designed for the shuttle.

In Fig. 10, we can see the reference path with comparison to the vehicle path while it is doing autonomous path following. We can also see two different controllers and how the performance is affected by changing coefficients according to the system parameters scaling down from one

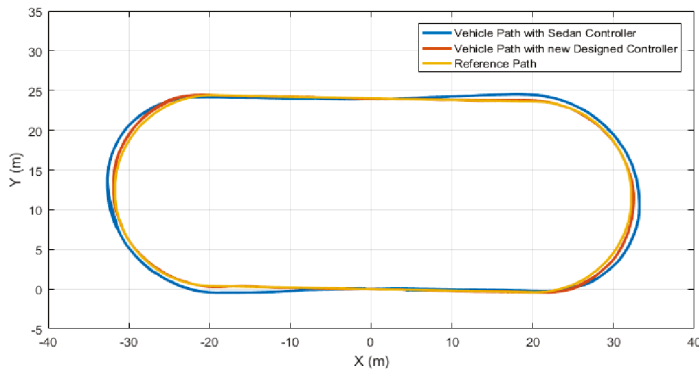


Fig. 10. Path following comparison for two different controllers.

vehicle to the other one. As expected, the sedan controller is not able to keep up with the path when there is a curve and creates a significant error as shown in Fig. 11. There is also a very small error while it is following a straight road. The shuttle controller, on the other hand, is better in both following a straight line or following a curved road as seen in both of Figs. 10 and 11. The shuttle follows the same path for both of its first and second laps in the experiments with both of the controllers without an appreciable difference, meaning repeatability is good for both of the controllers.

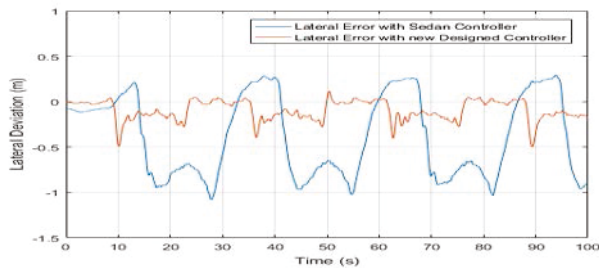


Fig. 11. Lateral error comparison for two different controllers.

Lateral error from the experiment with the sedan controller yields higher peak values and has RMS value of 0.5636 m. The shuttle controller yields much lower peak values and has RMS value of 0.1443 m, which is very close to the value of 0.121 we have from the sedan with sedan controller experiment discussed previously and displayed in Figs. 6 and 7.

VI. CONCLUSION

A unified architecture was presented in this paper for autonomous driving of low speed shuttles. Two different autonomous vehicles, a sedan and a low speed shuttle, were used to illustrate scalability and replicability using the automated path following of GPS waypoints application. Components and operation was also explained. Most of the components and

operation principles are the same for both vehicles. Both vehicles shared the same in-house developed library of Simulink localization and perception sensor blocks. We can also conclude that the same unified architecture and operation principles can be implemented on a third vehicle with minor changes which is a topic for future work.

Real world experiment results were used to demonstrate the performance and effectiveness of the approach. Using the scalability of the control strategy, the same steering controller architecture is applied to the other vehicle and using robust parameter space PD controller design. Other autonomous driving functions will be scaled and replicated in our future work.

ACKNOWLEDGMENT

This paper is based upon work supported by the National Science Foundation under Grant No.:1640308 for the NIST GCTC Smart City EAGER project UNIFY titled: Unified and Scalable Architecture for Low Speed Automated Shuttle Deployment in a Smart City. The authors would like to thank doctoral students Santhosh Tamilarasan and Ridvan Cantas for their help in automating the two vehicles presented in this paper.

REFERENCES

- [1] Behere, S., Törngren, M., "A functional reference architecture for autonomous driving," *Information and Software Technology*, Vol. 73, No. 1, pp. 7-28, 2008.
- [2] Inés, T., Vicente Milanés, J.V., Jorge Godoy, H.H.N., Blas, M.V., "Low speed control of an autonomous vehicle by using a fractional PI controller," *IEEE Conference on Intelligent Transportation Systems*, Oct 2011.
- [3] Yoshizawa, T., Raksincharoensak, P., Nagai, M., "Study of a path tracking control for low speed autonomous vehicle based on LIDAR information," *Trans. Japan Society of Mech. Eng. Series C*, Vol. 77, No. 773, pp. 114-123, 2011.
- [4] Güvenç, L., Uygan, İ.M.C., Kahraman, K., Karaahmetoğlu, R., Altay, İ., Şentürk, M., Emirler, M.T., Hartavi Karcı, A.E., Aksun Güvenç, B., Altuğ, E., Turan, M.C., Taş, Ö.Ş., Bozkurt, E., Özgüner, Ü., Redmill, K., Kurt, A., Efendioğlu, B., "Cooperative adaptive cruise control implementation of Team Mekar at the Grand Cooperative Driving Challenge," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 13, No. 3, pp. 1062-1074, 2011.
- [5] Emirler, M.T., Uygan, İ.M.C., Altay, İ., Acar, O.U., Keleş, T., Aksun Güvenç, B., Güvenç, L., "A Cooperating autonomous road vehicle platform," *IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications (ACATTA 2013)*, September 16-17, 2013.
- [6] Emirler, M.T., Wang, H., Aksun Güvenç, B., Güvenç, L., "Automated robust path following control based on calculation of lateral deviation and yaw angle error," *ASME Dynamic Systems and Control Conference, DSC 2015*, Oct., 2015.
- [7] Güvenç, L., Aksun Güvenç, B., Demirel, B., Emirler, M.T., *Control of Mechatronic Systems*, the IET, London, ISBN: 978-1-78561-144-5, 2017.