Parameter Space and Model Regulation based Robust, Scalable and Replicable Lateral Control Design for Autonomous Vehicles*

Sheng Zhu[a], Sukru Yaren Gelbal[b], Xinchen Li[b], Mustafa Ridvan Cantas[b], Bilin Aksun-Guvenc[a], *Member, IEEE* and Levent Guvenc[a,b], *Member, IEEE*

***Abstract*—This paper introduces a unified, scalable and replicable approach to make implementation of the autonomous system on a new vehicle faster while preserving its autonomous performance. The main idea of this approach is to create a standard hardware architecture, along with a Simulink or similar library and templates for autonomous driving for a unified approach to vehicle autonomy, making it easier to scale the solution and replicate it on other vehicle platforms. However, this scaling and replicating of the autonomous driving system between vehicles remains difficult especially for low-level controller design due to parametric difference between vehicles. This paper, hence, demonstrates a sequential controller design procedure with specific example of lateral control for a chosen vehicle. The same design process can be replicated to adapt controller parameters for other vehicles. The parameter space approach is applied here to ensure robust path following performance of a proportional-derivative (PD) steering controller, considering uncertainties of vehicle load, speed and tire cornering stiffness. To further reduce the tracking error and handle unmodeled dynamics and reject disturbances, a model regulator was added based on overall system analysis. To evaluate the control strategy, a validated high-fidelity model of an autonomous research vehicle is used within a hardware-in-the-loop (HIL) simulation environment. Soft sensors were also connected to the soft automated vehicle in the HIL environment to test high-level control and decision making mechanisms. The road used for the simulations is a replica of a designated real world short AV pilot route in the Ohio State University West campus. Traffic is generated with Simulation of Urban MObility (SUMO) software in order to analyze the problems due to the presence of other vehicles and evaluate performance more realistically in the HIL simulator.**

I. Introduction

Solutions to autonomous driving or advanced driver assistance systems (ADAS) continues to grow interest from research and industry. Traditional Original Equipment Manufacturers (OEMs) and software-based technology companies have mainly focused on passenger vehicle solutions due to its largest market share, while start-up companies have been working mostly on niche markets like low-speed autonomous shuttles operating on a fixed route. Regardless of different applications, their research and product development involve similarity and redundancy in some aspects. Meanwhile, as autonomous driving technology is advancing and series production would be expected in the near future, extensive testing processes for autonomous vehicles will be necessary. A standardized in-the-lab testing process is crucial to ensure that all autonomous driving functions operate as planned before proceeding with public road testing and deployment.

Therefore, a common and unified architecture of road vehicle autonomy that is easily scalable and replicable is beneficial for fast product development, saving development resources, and facilitating the validation process. There are examples of unified architectures that have already been investigated by several researchers [1-3]. In our work, these goals are achieved by defining a standard hardware architecture, by forming a software library and developing generic autonomous driving functions at different levels of vehicle autonomy [4].

Replicability of this unified architecture and scalability of the high-level decision making and low-level controllers are crucial to ensure easier implementation on various vehicle platforms [5]. Along with the unified structure, control algorithms also need to be replicable and scalable. This is challenged by significant parametric differences between vehicles. A standardized design procedure for vehicle control is, therefore, needed, to realize replicability and scalability.

In this paper, the design procedure for robust vehicle lateral control to follow the planned path is focused upon and the efficacy of our proposed approach is demonstrated by using our neighborhood electric vehicle (Dash), an experimental autonomous shuttle, as an example. The same control architecture and design procedures can be replicated for other vehicle platforms to adapt controller parameters. Indeed, the hardware and software libraries and basic autonomous driving functions were borrowed, scaled down and replicated from our 2017 Ford Fusion Hybrid research autonomous vehicle. In the steering control application considered here, path following performance should be ensured despite parameter variations like vehicle load, speed and tire cornering stiffness. This demands a robust design for the lateral control by taking these uncertainties into consideration. This paper proposes a robust proportional-derivative (PD) controller design using the parameter-space approach. D-stability and mixed-sensitivity requirements were imposed to ensure its robust performance. To further reduce tracking error, a model regulator was also designed in combination with the robust PD controller.

The organization of the paper is as follows. Section II describes the unified architecture and automation library.

*This work was supported in part by the National Science Foundation under Grant 1640308, and by the U.S. Department of Transportation Mobility 21: National University Transportation Center for Improving Mobility (CMU) sub-project titled: SmartShuttle: Model Based Design and Evaluation of Automated On-Demand Shuttles for Solving the First-Mile and Last-Mile Problem in a Smart City.

Authors are with Center for Automotive Research and with [a] Department of Mechanical and Aerospace Engineering or [b] Department of Electrical and Computer Engineering, Ohio State University, OH 43212 USA (e-mail: zhu.1473@osu.edu; gelbal.1@osu.edu; li.6312@osu.edu; cantas.1@osu.edu; aksunguvenc.1@osu.edu; guvenc.1@osu.edu).

Section III shows the design procedures of the robust PD controller and add-on model regulator by taking the Dash vehicle as an example. Section IV depicts the high-level decision making strategies. Section V explains the validation and evaluation of the control strategy along with sensor placement and autonomous decision making within the hardware-in-the-loop environment. This is followed by the simulation validation of the overall unified architecture and our conclusions.

II. UNIFIED ARCHITECTURE AND AUTOMATION LIBRARY

Our previous work [4,6] on unifying the structure with scalability and replicability is to create a standard base for hardware structure along with a library to be used by developers for faster and easier automation of vehicles. Hardware structure includes different types of sensors to achieve enough coverage, resolution and also robustness to external disturbances. Data from these sensors is being processed by a high processing power computer to create meaningful information, which is used by a low-level controller, i.e. a dSpace MicroAutobox in our vehicles, to drive the vehicle autonomously by interfacing with actuators and sending necessary commands. The unified architecture is shown in Fig. 1.

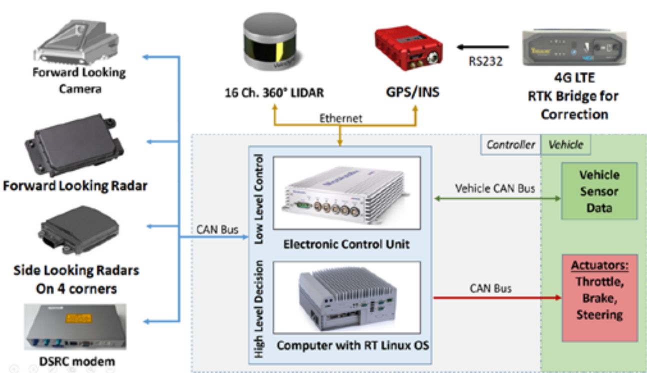

Figure 1. Unified architecture.

Using this unified architecture, two different sized vehicles were automated. Perception sensors such as Lidar, Camera, Radars are implemented as well as GPS Sensor for localization. The dSpace MicroAutobox unit is used for low level controls and an in-vehicle Linux PC with a GPU is used for sensor data computation. Moreover, DSRC (dedicated short-range communications) radios are added to have the capability of communicating with other vehicles, pedestrians, bicyclists and infrastructure. Pictures of the vehicles and the implemented hardware were shown in our previous papers as well as the evaluation of the scalability approach.

Along with the unified architecture, a unified Simulink library was created. This library shown in Fig. 2 consists of different types of blocks, including low level control blocks for steering, throttle, brake, shift; sensor blocks for receiving data from the sensors in order to have environment perception and localization; and finally control and decision-making blocks for low and high level control of the autonomous vehicle. We are currently extending this library for use with NVIDIA Drive PX 2 GPUs. It is slightly modified for CarSim soft sensors and then used in the HIL simulations reported in this paper.

Figure 2. Simulink vehicle automation library. (Not legible, to be replaced)

III. VEHICLE LATERAL CONTROL

Vehicle longitudinal control is realized by tuning a PID controller to follow the speed profile and so is not covered here for the sake of brevity. This paper describes the design of lateral control in detail as it is crucial for path following performance. The vehicle lateral control is comprised of a robust proportional-derivative (PD) controller and an add-on model regulator. While this section took the Dash vehicle as an example, the same lateral control structure and design procedures could easily be applied to other vehicles.

A. Vehicle Model

The bicycle model was used to design the lateral control, as depicted in Fig. 3.

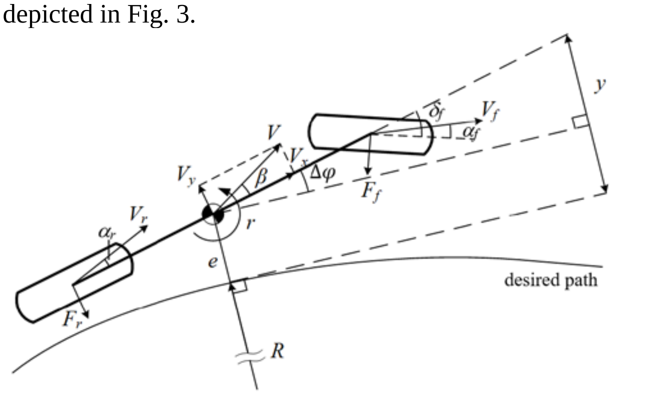

Figure 3. Bicycle model and its deviation from the path

The state-space equation to describe the vehicle states and its deviation from the planned path is [7]:

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

where the coefficients used are

$$a_{11} = -\frac{C_{\alpha f} + C_{\alpha r}}{\tilde{m} V_x}, \qquad a_{12} = -1 - \frac{C_{\alpha f} l_f - C_{\alpha r} l_r}{\tilde{m} V_x^2},$$

$$a_{21} = -\frac{C_{\alpha f} l_f - C_{\alpha r} l_r}{\tilde{I}_z}, \qquad a_{22} = -\frac{C_{\alpha f} l_f^2 + C_{\alpha r} l_r^2}{\tilde{I}_z V_x},$$

$$b_{11} = \frac{C_{\alpha f}}{\tilde{m} V_x} , \qquad b_{21} = \frac{C_{\alpha f} l_f}{\tilde{I}_z} .$$

$C_{\alpha f}$, $C_{\alpha r}$ are lumped tire cornering stiffness for front and rear sets of tires respectively; V_x is the vehicle longitudinal speed; l_f, l_r are the lengths from the vehicle center of gravity to its front and rear axle respectively; β is the side slip angle; r is the yaw rate of the vehicle; $\Delta\psi$ is the vehicle heading angle error from the path; δ_f is the steering angle of the front wheel; ρ_{ref} is the planned path curvature; y is the look-ahead error from the path at forward distance l_s and can be calculated from lateral error e and heading error $\Delta\psi$:

$$y = e + l_s \Delta\psi , \tag{2}$$

$\tilde{m} = m/\eta$, and $\tilde{I}_z = I_z/\eta$ are defined as virtual mass and virtual yaw moment inertia, η is the tire saturation parameter [8]. In this way, the vehicle mass and the tire coefficient η are lumped together for the convenience of uncertainty analysis.

Since the Dash vehicle is operated at low-speeds without extreme behaviors, the uncertainties in the vehicle lateral dynamics are defined accordingly. The vehicle mass is estimated to be inside the range of [300, 500] kg ranging from its curb mass to full-load with two passengers, the velocity is within [2, 10] m/s, and the tire saturation parameter varies between [0.5, 1]. Its uncertainty region is depicted in Fig. 4, along with the platform 2017 Ford Fusion Hybrid which has higher mass and speed range.

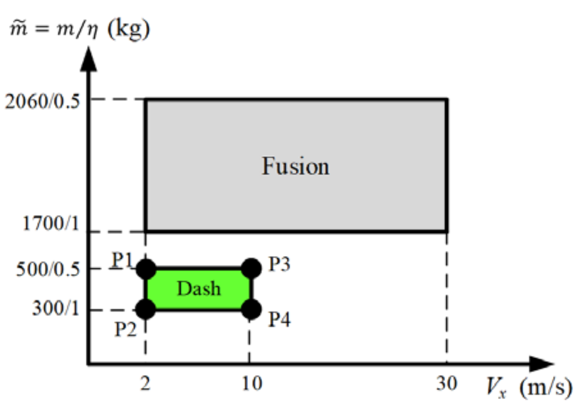

Figure 4. Uncertainty region of vehicle mass m, longitudinal speed V_x and parameter η for Dash and Fusion experiment platforms

B. Robust Proportional-Derivative Steering Control

A robust proportional-derivative (PD) controller was designed with feedback of look-ahead error y to follow the planned path in the presence of model uncertainties:

$$C(s) = k_p + k_d s \tag{3}$$

D-stability requirement was raised to ensure converged system response within setting time of 8 sec and damping ratio larger than 0.4. These requirements were reflected in the D-stability region in s-plane (Fig. 5) with $\sigma = 0.5$, $R = 100$ and $\theta = 66.2°$.

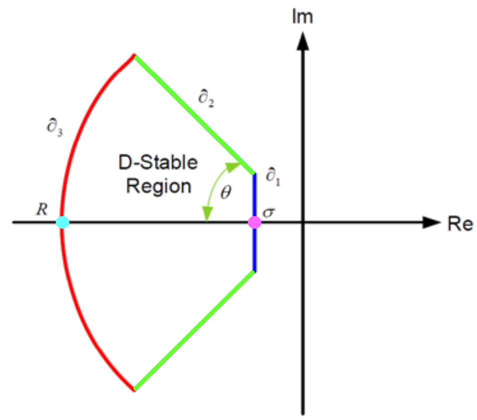

Figure 5. Illustration of D-stability region in the complex plane

To take both tracking performance and robustness to model uncertainties into account, the mixed sensitivity criterion was considered. The robust performance can be ensured by satisfying:

$$\left\| |W_S S| + |W_T T| \right\|_\infty < 1 , \tag{4}$$

or equivalently

$$|W_S S| + |W_T T| < 1 \quad \forall \omega , \tag{5}$$

where S is sensitivity function, T is complementary sensitivity function, W_S and W_T are weights for S and T respectively. The inverse of the sensitivity function weight is chosen as

$$W_S^{-1}(s) = h_s \frac{s + \omega_s l_s}{s + \omega_s h_s} , \tag{6}$$

with $l_s = 0.5$ being the low-frequency sensitivity bound, $h_s = 4$ being the high-frequency sensitivity bound, and $\omega_s = 3$ rad/s. The complementary sensitivity function weight is chosen as

$$W_T(s) = h_T \frac{s + \omega_T l_T}{s + \omega_T h_T} , \tag{7}$$

where the low-frequency gain is $l_T = 0.2$, the high-frequency gain is $h_T = 2$ (corresponds to uncertainty up to 200% at high frequencies), and the frequency of transition to significant model uncertainty is $\omega_T = 20$ rad/s.

The parameter space approach [9] is able to reflect the D-stability boundaries and mixed sensitivity point conditions to the design space of controller parameters k_p and k_d. The parameter space for the four vertices of the uncertainty region is shown in Fig. 6 (a-d). The colored lines are reflections of D-stability boundaries of same colors as in Fig. 5. The points of blue envelop curves were obtained by substituting different frequency values at critical condition of (5). The PD controller parameter was hence selected to be $k_p = 0.5$, $k_d = 0.035$ as indicated with the red cross in the overlapped selectable region (Fig. 6 (e)), satisfying the design requirements at all four uncertainty vertices. The correspond-

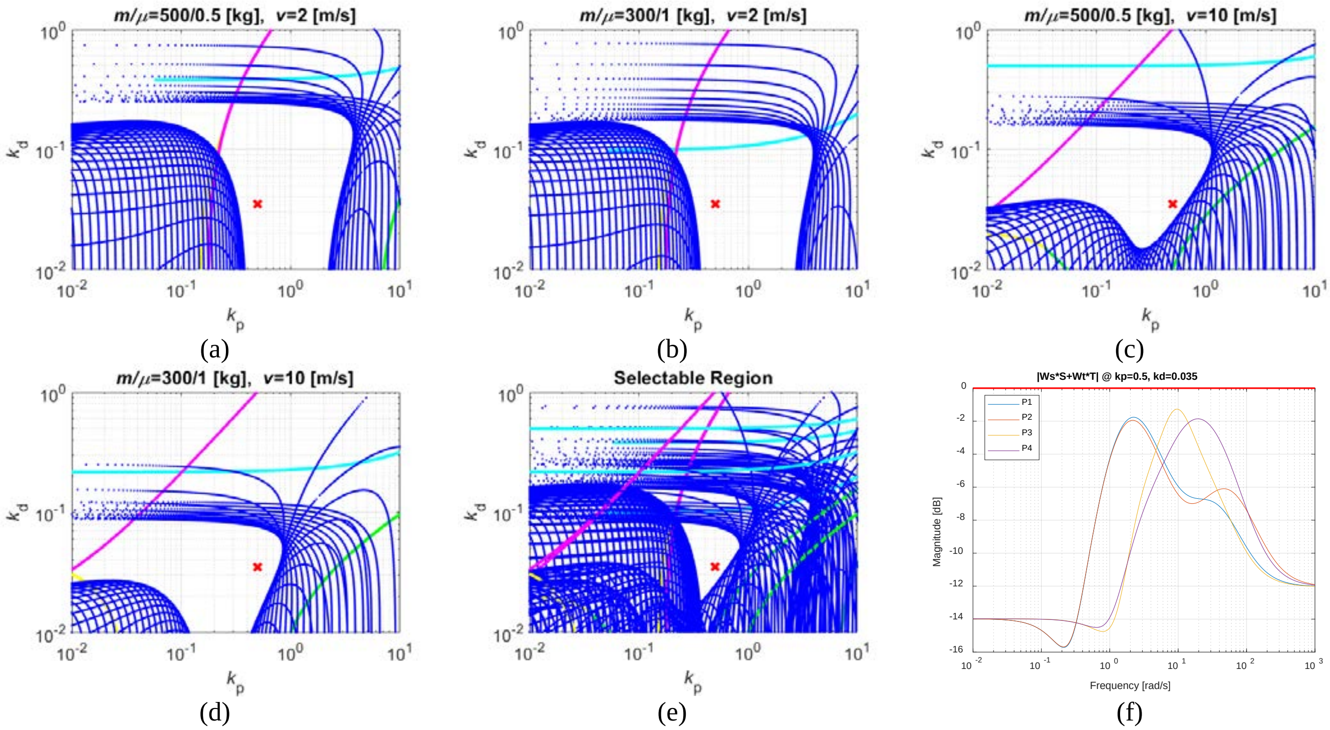

Figure 6. Parameter Space region at apex (a) P1 (b) P2 (c) P3 (d) P4 and (e) overlapped selectable region; (f) Robust performance with selected PD parameters (red cross) at each apex

ing $|W_S S| + |W_T T|$ magnitude plot at each vertex of the uncertainty region is shown in Fig. 6(f) and validates that the mixed sensitivity constraint is met.

C. Add-on Model Regulator

To further reject the look-ahead error, a model regulator was added together with the previously designed robust PD controller (Fig. 7). The model regulator, also referred to as disturbance observer, is proven effective in disturbance rejection and in achieving insensitivity to modelling errors. Its applications in for example in direct drive positioning [10] and friction compensation [11] are successful.

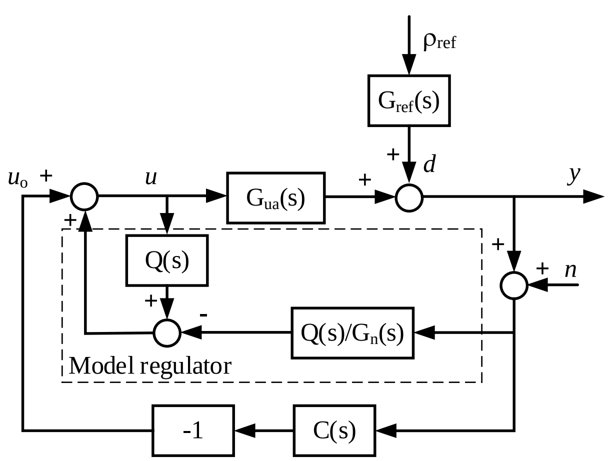

Figure 7. System diagram with the PD controller and model regulator

The filter Q(s) is chosen to make inverse of nominal model $Q/G_n(s)$ causal with a cutoff frequency at 10 rad/s:

$$Q(s) = \frac{1}{(0.1s+1)^2}. \tag{8}$$

Since the steering is also affected by the look-ahead error through the PD control C(s), the design of model regulator should consider the overall system. The loop gain is:

$$L(s) = \frac{G_{ua}(C + Q/G_n)}{1-Q}. \tag{9}$$

Transfer function of look-ahead error over path curvature and noise can be expressed as:

$$\frac{y}{\rho_{ref}} = \frac{G_{ref}}{1+L} = \frac{G_{ref}(1-Q)}{1-Q+CG_{ua}+G_{ua}Q/G_n}, \tag{10}$$

$$\frac{y}{n} = \frac{-(C+Q/G_n)G_{ua}}{(1-Q)(1+L)} = \frac{-G_{ua}(C+Q/G_n)}{1-Q+CG_{ua}+G_{ua}Q/G_n}. \tag{11}$$

Curvature of the path usually only presents low-frequency characteristic. Therefore to further reject the response of look-ahead error due to curvature, $|y/\rho_{ref}|$ should approach zero at low frequency. To reject noise influence, $|y/n|$ should have small amplitude at high frequency. Considering the characteristic of the filter $Q(s)$, if the nominal model G_n is chosen to approximate G_{ua} closely at high frequency, the stated requirements can be satisfied.

The transfer function of $G_{ua}(s)$ and $G_{ref}(s)$ have the form:

$$G_{ua} = \frac{n_2 s^2 + n_1 s + n_0}{(d_2 s^2 + d_1 s + d_0)s^2}, \tag{12}$$

$$G_{ref}(s) = -\frac{l_s V_x \left(s + \frac{V_x}{l_s}\right)}{s^2}, \tag{13}$$

where the coefficients n_i, d_i (i =1,2,3) are related to vehicle parameters. Frequency responses of $G_{ua}(s)$ at the uncertainty vertices of Fig. 4 are shown in Fig. 8 and exhibit large differences at low frequencies.

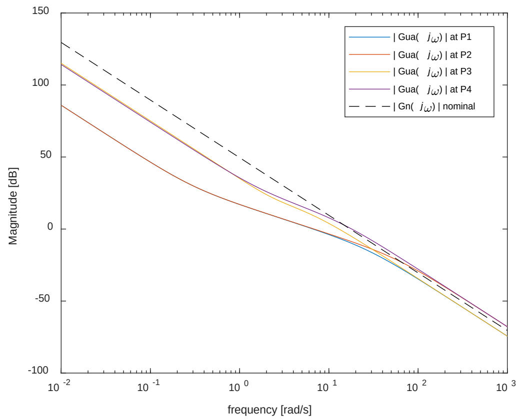

Figure 8. Magnitude $G_{ua}(j\omega)$ for the four vertices

It could be observed that the magnitude of $G_{ua}(j\omega)$ are similar at high frequency for the vertices of the uncertainty region. Therefore, we could use a single nominal model $G_n(s)$ to approach $G_{ua}(s)$ at high frequency for all the conditions inside the uncertainty region:

$$G_n(s) = \frac{k_n}{s^2}, \tag{14}$$

where magnitude of $G_n(s)$ at k_n =300 was also shown as the dashed line in Fig. 8.

The magnitude responses $|y/u|$, $|y/\rho_{\text{ref}}|$ and $|y/n|$ before and after adding the designed model regulator are shown in Fig. 9. The magnitude $|y/u|$ converges especially below cut-off frequency of filter Q(s), suggesting good model regulation effect. $|y/\rho_{ref}|$ showed effective rejection of path curvature at low frequency, meaning steady-state tracking error will be greatly reduced. Meanwhile, the noise rejection still remains satisfactory at high frequency as seen from $|y/n|$.

IV. Rule Based Decision Making

To accomplish the function of auto-driving, considering road traffic and infrastructure, we design a control logic based on rule-based decision-making method. Information about desired path, ego-motion, traffic sign and traffic light is provided by the in-vehicle Linux PC or sensors directly. The decision making used in the autonomous drive in our AV test pilot route (from Car to Car West) is represented as a FSM (Finite Sate Machine) in Fig. 10.

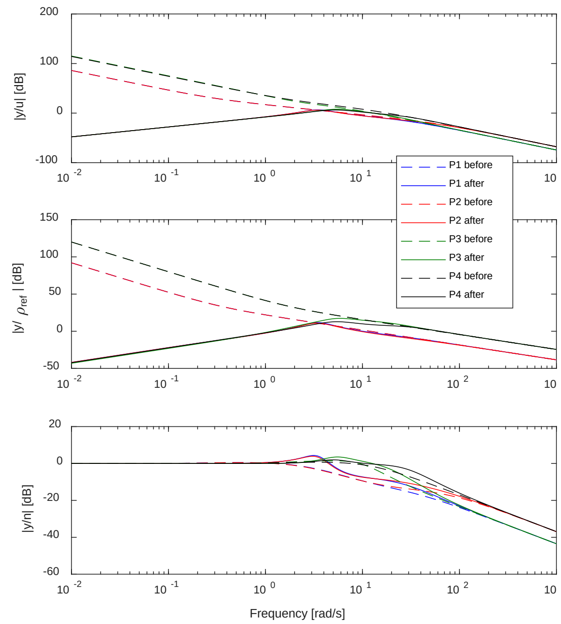

Figure 9. Magnitude $|y/u|$, $|y/\rho_{\text{ref}}|$ and $|y/n|$ before and after applying model regulator

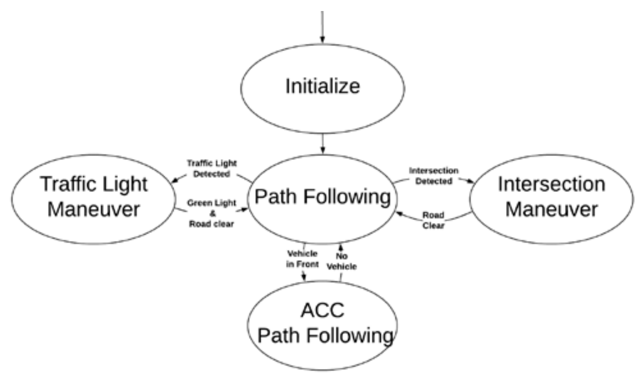

Figure 10. Decision making chart for autonomous drive

At the Initialize state, the vehicle checks whether all sensors are working properly. Other states are explained below:

- Path Following: the vehicle goes along the planned path with the aforementioned lateral control algorithm.
- ACC (Adaptive Cruise Control) Path Following: When a vehicle is detected in front, the ego-vehicle adapts its peed to the target while following the planned path.
- Traffic Light Maneuver: Trigger when receiving traffic signal phase and timing (SPaT) information: If the light is Green, our vehicle will check crossing traffic until the road is clear and switch back to path following; if the light is yellow or red, it will be triggered to stop at the traffic light to wait for the red light turning into green.
- Intersection Maneuver: Triggered when the vehicle comes to an intersection or a stop sign, ego-vehicle will wait at the intersection, detecting crossing traffic until the road is clear.

Based on the control logic, the soft version of our Dash autonomous vehicle was able to run the shuttle task with the designed low level robust steering and speed controllers, without running into problems in repeated simulations with random traffic in our HiL simulator.

V. HIL Environment and Simulations

To extensively evaluate the performance of the developed control strategy, along with high level control, decision making and sensor placement, a hardware-in-the-loop (HIL) simulator is employed (Fig. 11). A high-fidelity CarSim vehicle model runs in dSpace Scalexio HIL platform to simulate vehicle lateral and longitudinal motions alongside the perception sensors in real time, while the MicroAutobox controller implements the actual control scheme at 100Hz sampling frequency.

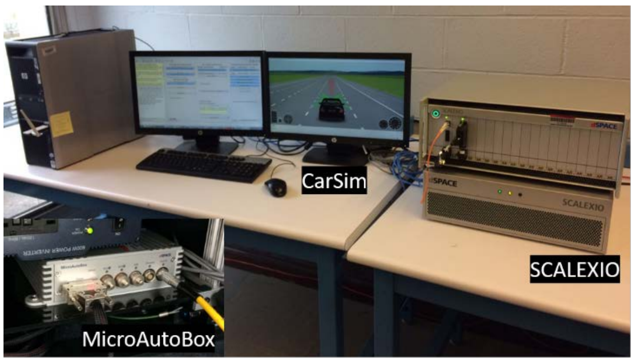

Figure 11. Hardware-in-the-Loop setup

A. Slalom Path

Fig. 12 shows the reference slalom path and the trajectory of Dash at 10m/s with the designed lateral control, along with another of Ford at 30m/s following the same lateral control design procedure. Both vehicles were able to follow the slalom path of tracking error within 0.15 m, suggesting the replicability of the lateral control design procedure.
The effect of the added model regulator was also proved to be effective in reducing look-ahead error as compared to the use of the PD steering controller alone (Fig. 13). The steering angle and yaw motion during the process were without apparent oscillation (Fig. 14). Contributions from PD controller and model regulator respectively are also shown in Fig. 14.

B. CARWest-to-CAR Route

The OSU AV pilot test route from CAR West (our lab location) to CAR (Center for Automotive Research – our main research center) shown in Fig. 16 was chosen and constructed in CarSim to evaluate the vehicle's decision making and lateral control performance. To incorporate the real traffic into simulation, information about other vehicles on the road were imported from SUMO software (Fig. 15).

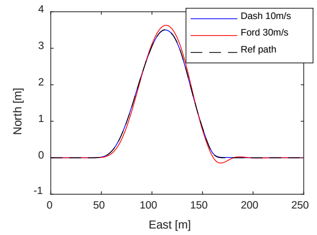

Figure 12. Slalom trajectory of Dash at 10m/s and Ford at 30m/s

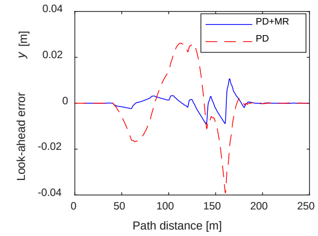

Figure 13. Look-ahead error with and without model regulator (Dash 10m/s)

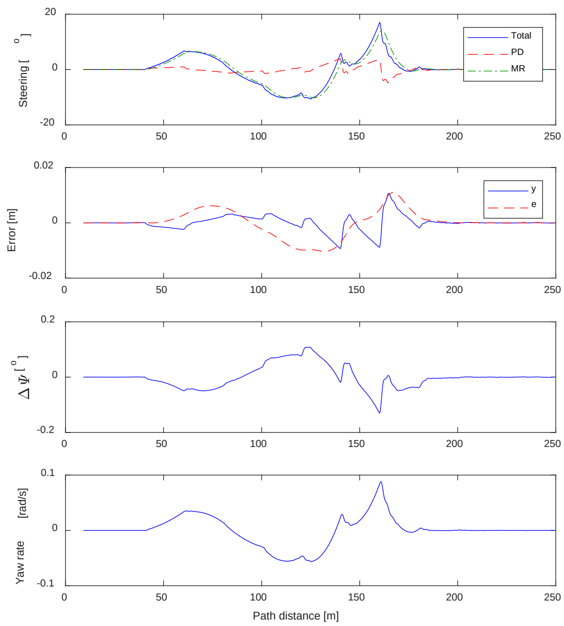

Figure 14. Steering, tracking error and yaw rate of slalom test (Dash 10m/s)

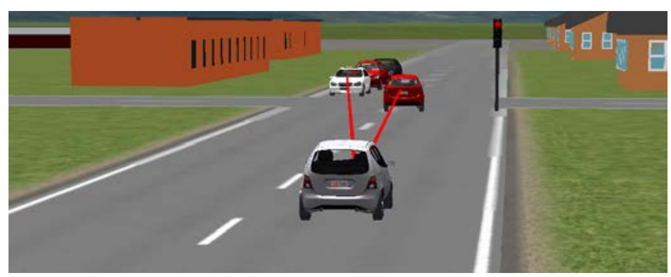

Figure 15. Constructed road with traffic in Carsim

Placement and field of view (FOV) of the sensors described in section II can be seen in Fig. 17. These were implemented as soft sensors in the HIL simulator.

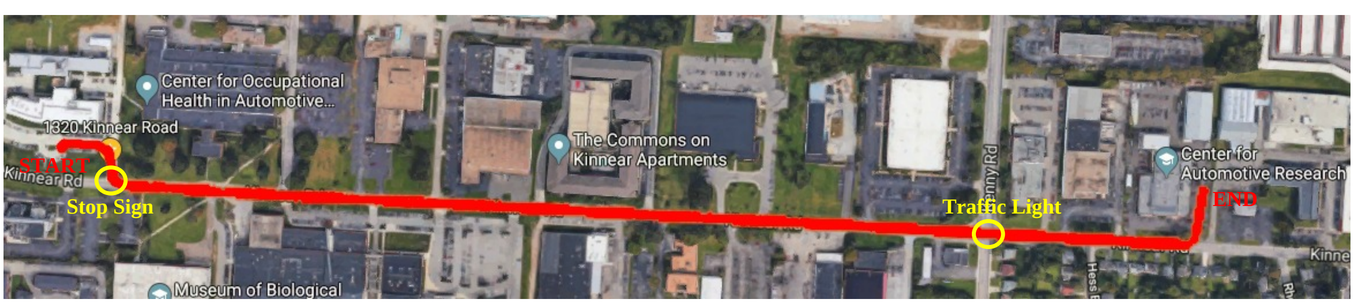

Figure 16. CARWest-to-CAR route

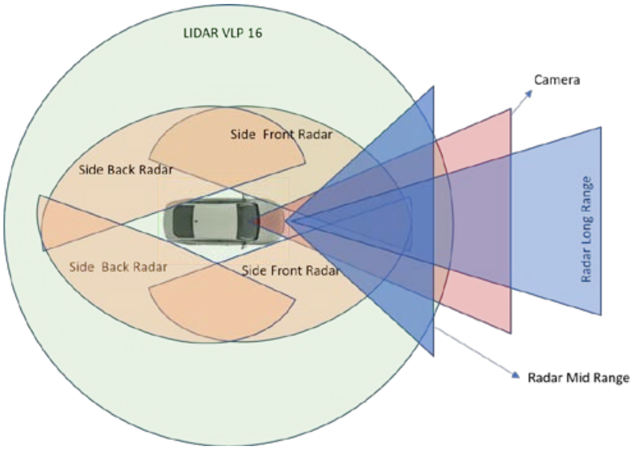

Figure 17. Placement and FOV of the sensors on the car

Scenarios including stop sign, crossing traffic and traffic lights were simulated on the route in Fig. 16 to test the decision-making strategies introduced in Section IV. Fig. 18 shows the vehicle speed, steering and tracking error along the route. Since sharp turns appear when entering and exiting the main straight road, lateral error *e* was expectedly high for these two cases, but the look-ahead error *y* which combines the lateral and heading angle error was still relatively small.

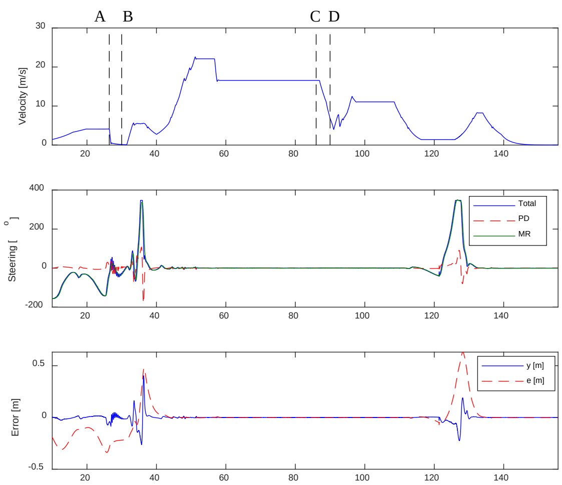

Figure 18. CARWest-to-CAR AV pilot test route simulation results (A-close to stop sign; B-traffic crossing; C-traffic light nearby and turns red; D-traffic light turns green)

CONCLUSION

A unified and replicable approach on lateral control design procedure based on robust PD controller with parameter space design and add-on model regulator is introduced. HIL test on slalom path suggested good path following performance of the lateral control and its replicability to another larger vehicle platform. The lateral control was integrated with our previous developed unified, scalable and replicable autonomous driving solution. Decision-making strategies, sensor perception and lateral control were evaluated in the Ohio State University AV pilot test route with random traffic in a simulation study. Autonomous shuttles are planned to be used on this short route and then extended to the rest of the university campus. The approach used in this paper presented a method of in-lab evaluation in a realistic traffic environment for identifying and fixing possible problems before an actual deployment. A future experiment will be conducted with our experimental vehicles to evaluate our unified architecture.

REFERENCES

[1] S. Behere and M. Törngren, "A functional architecture for autonomous driving," in *Proceedings of the First International Workshop on Automotive Software Architecture*, pp. 3-10, 2015.

[2] K. Jo, J. Kim, D. Kim, C. Jang and M. Sunwoo, "Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Trans.Ind.Electron*., vol. 62, pp. 5119-5132, 2015.

[3] J. Ziegler, P. Bender, M. Schreiber and H. Lategahn, "Making Bertha Drive- An Autonomous Journey on a Historic Route," *IEEE Intell. Transport. Syst. Mag*., vol. 6, no. 2, pp. 8-20, 2014.

[4] S.Y. Gelbal, B.A. Guvenc and L. Guvenc, "SmartShuttle: a unified, scalable and replicable approach to connected and automated driving in a smart city," in *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering*, pp. 57-62, 2017.

[5] F. Bonsignorio, and A. P. del Pobil, "Toward replicable and measurable robotics research," *IEEE Robot. Automat. Mag*., vol. 21, no. 1, pp. 22-25, 2015.

[6] S.Y. Gelbal, N. Chandramouli, H. Wang, B. Aksun-Guvenc and L. Guvenc, "A Unified Architecture for Scalable and Replicable Autonomous Shuttles in a Smart City," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3391-3396, 2017.

[7] L. Guvenc, B. Aksun-Guvenc, B. Demirel and M.T. Emirler, *Control of Mechatronic Systems*, Institution of Engineering and Technology, 2017.

[8] K.L. Talvala, K. Kritayakirana and J.C. Gerdes, "Pushing the limits: From lanekeeping to autonomous racing," *Annual Reviews in Control*, vol. 35, pp. 137-148, 2011.

[9] J. Ackermann, Robust control: the parameter space approach, Springer Science & Business Media, 2012, .

[10] Y.A.I. Mohamed, "Design and implementation of a robust current-control scheme for a PMSM vector drive with a simple adaptive disturbance observer," *IEEE Trans.Ind.Electron*., vol. 54, pp. 1981-1988, 2007.

[11] Z. Jamaludin, H. Van Brussel and J. Swevers, "Friction compensation of an XY feed table using friction-model-based feedforward and an inverse-model-based disturbance observer," *IEEE Trans.Ind.Electron*., vol. 56, pp. 3848-3853, 2009.