Pedestrian Emergency Braking in Ten Weeks

Steven Nguyen
Mechanical Engineer
University of California, Santa Barbara
Santa Barbara, United States
stevennguyen@ucsb.edu

Zillur Rahman
Electrical and Computer Engineering
University of Nevada, Las Vegas
Las Vegas, United States
zillur.rahman@unlv.edu

Brendan Tran Morris
Electrical and Computer Engineering
University of Nevada, Las Vegas
Las Vegas, United States
brendan.morris@unlv.edu

Abstract—In the last decade, research in the field of autonomous vehicles has grown immensely, and there is a wealth of information available for researchers to rapidly establish an autonomous vehicle platform for basic maneuvers. In this paper, we design, implement, and test, in ten weeks, a PD approach to longitudinal control for pedestrian emergency braking. We also propose a lateral controller with a similar design for future testing in lane following. Using widely available tools, we demonstrate the safety of the vehicle in pedestrian emergency braking scenarios.

Index Terms—Control theory, autonomous vehicles, pedestrian safety, implementation

I. INTRODUCTION

While autonomous vehicles (AVs) were once a far-fetched dream only found in science fiction, it has become increasingly clear that they will be integral to the future of our transportation systems. In recent decades, many commercial cars have featured tools to assist drivers, like adaptive cruise control and lane following technology. With all the research in the field, there is a wealth of information available for researchers interested in establishing an AV platform.

The goal of this project was to develop a control algorithm for longitudinal motion of an AV and demonstrate, on a real-world vehicle, the ability to safely stop when approaching a pedestrian. A proportional and derivative gain (PD) controller was designed for control of throttle and braking and was interfaced with the vehicle controller area network (CAN) via the Robot Operating System (ROS). Pedestrians were detected using the YOLOv5 model [1]. The safety of the proposed longitudinal PD controller was demonstrated by testing the vehicle's ability to stop at a safe distance from the pedestrian. In addition, it is shown that the proposed controller can stop in a manner that is comfortable for the passengers, by gradually slowing down rather than slamming on the brakes. A similar PD controller was designed for lateral control of the vehicle in lane following, but has not yet been tested in-vehicle.

II. RELATED WORKS

Challenges in AV control have been tackled using a variety of techniques in the past. One of the most popular techniques

for vehicular control is model predictive control (MPC), which excels in nonlinear optimization problems. In [2], researchers demonstrated that MPC can be used to consolidate the lateral (steering) and longitudinal (braking) controllers in a manner that allows a vehicle to accurately follow a given path despite tortuous conditions of snowy, slippery roads even at high speeds. In addition, MPC was shown in [3] to be suitable for improving fuel efficiency of AVs. By defining the controller to minimize fuel consumption of the vehicle, researchers increased fuel efficiency by up to 20%.

Alternatively, controllers based on fuzzy logic can offer an intuitive approach to controller design that closely mimics human driver behavior [4]. Through the use of fuzzy inference systems, the controller can make inferences on the state and the appropriate control action without necessitating rigorous dynamical models.

Another approach that has gained popularity is the use of artificial intelligence, through deep neural networks and reinforcement learning [5], [6]. Deep learning approaches for AV control often rely on training models to extract features from input images and mapping them to the desired output, which are the desired control actions. Sharma et al. [5] demonstrated that a deep learning approach, even with limited training data, was capable of navigating tracks in the TORCS simulation environment with minimal lane departure. Alternatively, reinforcement learning is valuable because it converts the challenge of designing the controller into a challenge of training an agent that learns a desirable control law. Kendall et al. [6] presented a reinforcement learning model that can achieve lane-following with just 30 minutes of training.

Lastly, another common approach to controller design is to use proportional, integral, and derivative (PID) controllers. As one of the simplest to implement control schemes, PID controllers are computationally light and offer a conceptually straightforward design approach. Though PID controllers are not typically as robust as nonlinear approaches, some researchers were able to adopt an adaptive PID controller, where the control gain is tuned in real-time depending on the driving conditions [7], to great success. Their approach successfully demonstrated the ability for path tracking and was accurate to within 0.5 meters. Alternatively, PID controllers were designed in a nested loop in [8] for lateral control. In this design, the outer feedback loop is a PID controller to determine the desired yaw rate of the vehicle, and the nested feedback loop

is a separate PID controller to stabilize the yaw rate of the vehicle to the reference yaw rate. A similar approach to this is proposed in this study for PD controllers for both longitudinal and lateral control.

In terms of implementation, one consideration that must be made when implementing any control scheme in an AV is the communication between the various sensors and computational units in the vehicle. ROS is a commonly used, open-source software that simplifies this communication by allowing the user to create nodes that publish and read information that can be read and processed by the main CPU. For example, ROS was used in [9] for implementation in an electric golf cart and in [10] for an autonomous mobile robot. Alternatively, Guo et. al [11] used User Datagram Protocol (UDP) to handle the communication between processing units when implementing MPC in an AV. UDP is used in lieu of ROS by Jones et. al [12] for its favorable speed in an implementation that was similar in scope to this work. However, their work is more focused on simulations and neural network training whereas this work will focus on the implementation of a closed loop control scheme. In this work, ROS is chosen due to its widespread use and support for AVs for rapid development.

III. CONTROL SCHEMES

The control basics consists of a longitudinal controller to regulate acceleration and braking and a lateral controller to handle steering of the vehicle.

A. Longitudinal Control

For the challenge of braking in the presence of pedestrians, it is of utmost importance that the longitudinal controller ensures the vehicle will stop at a safe distance. To assist in this, the proposed control algorithm subtracts a predetermined distance from the pedestrian's position to use as the stopping position. This distance can be changed depending on the driver's comfort. Position and velocity feedback are both used to prevent overshoot on the desired stopping position and to ensure the vehicle gradually stops in a manner comfortable for the passengers.

The PD controller design is shown in the block diagram in Fig. 1. The outer feedback loop uses position feedback and proportional and derivative control to generate a reference velocity, $r_v(t)$,

$$r_v(t) = K_p e_1(t) + K_d \frac{d}{dt} e_1(t). \tag{1}$$

This reference velocity is then used in the inner feedback loop, which is designed to stabilize the velocity of the vehicle and gradually bring it to a stop using a proportional gain, K. The inner loop generates the braking force, $u(t)$,

$$u(t) = K e_2(t), \tag{2}$$

which is converted into a braking command for the vehicle.

The combination of these feedback loops ensures that, as the vehicle approaches the referenced stopping position, $r(t)$, the reference velocity similarly approaches 0, and the vehicle
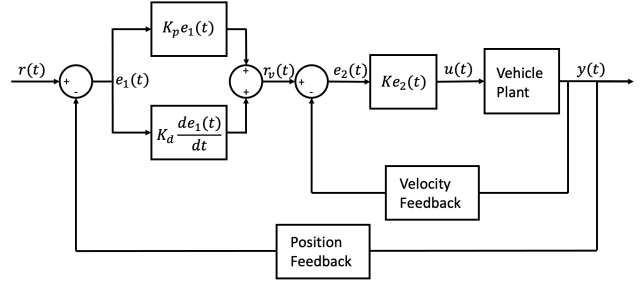

Fig. 1: Block diagram for longitudinal control algorithm with both position and velocity feedback.

comes to a stop. This approach is similar to that used by Marino et al. in [8].

For the stability analysis, a simple model of the vehicle longitudinal dynamics was used for the plant as shown below:

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u. \tag{3}$$

It accounts for the inertia from the vehicle mass, m, and treats the braking force as the input. The car's position is the variable x. Drag was excluded in stability analysis because of its nonlinear dependence on the square of velocity and its minimal impact at the velocities during testing (under 25 miles per hour). In practice, drag serves to increase the braking force, and increases the safety margin. Drag effects are included in the simulations in Section V.

The complementary sensitivity function from the reference position to the vehicle position is given as:

$$G_{\text{Position},R}(s) = \frac{K(K_p + K_d s)}{ms^2 + KK_d s + K + KK_p}, \tag{4}$$

where K, K_p, and K_d represent the controller gains as depicted in the block diagram, and m is the mass of the vehicle. By definition, all the coefficients of the denominator polynomial are positive, so the system is stable by the Routh-Hurwitz stability criterion.

While a PD controller with position feedback would have been sufficient to ensure stability of the closed-loop system, velocity feedback was incorporated in the proposed manner to address the requirement of comfort for passengers. The sensitivity function for the velocity feedback loop is given as:

$$G_{E_2(t),R_v}(s) = \frac{ms}{ms + K}. \tag{5}$$

By the final value theorem, the error of the vehicle velocity with respect to the reference velocity, $e_2(t)$, is bounded in ramp response. This is significant because it means that for any given pedestrian distance and initial velocity, the acceleration of the car can be tuned depending on the reference velocity. With the proposed controller, this means that the gains K_p and K_d can be tuned to increase or decrease the maximum acceleration of the vehicle.
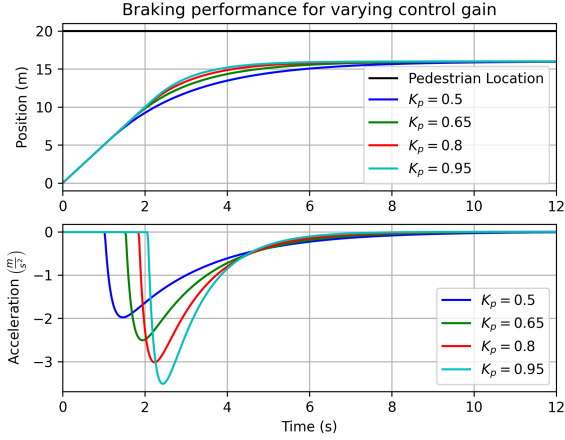
Fig. 2: This design offers control over the acceleration that the passenger experiences by tuning K_p.

Fig. 3: Proposed control algorithm uses both position and velocity feedback.

In Fig. 2, the effect of varying K_p is shown. Lowering K_p results in a lower acceleration because the vehicle will begin braking earlier and will more gradually approach the stopping point. This extra control over the acceleration of the vehicle does not sacrifice any safety, as all of the trajectories converge to the same final position. The experimental behavior of this control system and the simulated responses are shown in Section V.

For the experiments conducted in Section V, gain values of $K_p = 0.8$, $K_d = 0.1$, and $K = 10,000$ were used. The large value for K is because the force $u(t)$ accounts for the vehicle mass of approximately 1,725 kilograms.

B. Lateral Control

A lateral control algorithm is also proposed with a similar design to the longitudinal controller. Whereas longitudinal control governs how a vehicle will brake or accelerate, lateral control governs the steering and yaw rate of the vehicle.

The block diagram for the lateral controller is shown in Fig. 3. It is identical to the longitudinal controller in most respects, but different because the inner loop is used for feedback control of the yaw rate of the vehicle rather than the velocity. Instead of a reference velocity, $r_v(t)$, there is a reference yaw rate, $r_{\dot{\psi}}$.

To model the vehicle yaw rate dynamics, the linearized model presented by Antonov et al. [13] was used. The state space model is

$$\frac{d}{dt}\begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -\frac{c_r+c_f}{mv_x} & \frac{c_rl_r-c_fl_f}{mv_x}-v_x \\ \frac{c_rl_r-c_fl_f}{I_zv_x} & -\frac{c_rl_r^2+c_fl_f^2}{I_zv_x} \end{bmatrix}\begin{bmatrix} v_y \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \frac{c_f}{m} \\ \frac{c_fl_f}{I_z} \end{bmatrix}\delta_f + \begin{bmatrix} 0 \\ \frac{1}{I_z} \end{bmatrix}M_z. \quad (6)$$

The variables c_f and c_r represent the cornering stiffnesses of the front and rear axles, m is the mass, v_x is the longitudinal velocity of the vehicle, v_y is the lateral velocity of the vehicle, and l_r and l_f are the distances of the rear and front axles
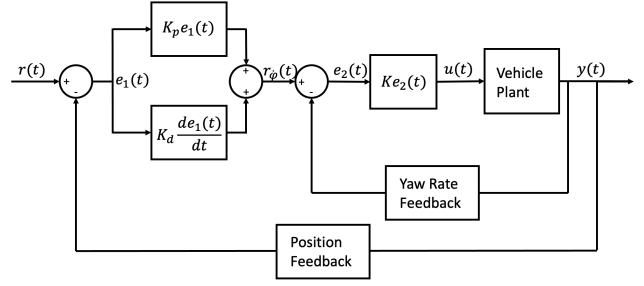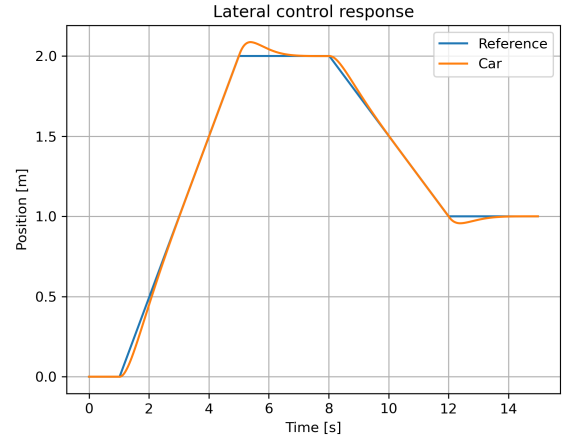

Fig. 4: Response of lateral control algorithm shows that it is stable in step response.

from the vehicle center of mass [13]. M_z is the yaw torque, but is taken to be 0 in our model. Stability analysis was not performed on the lateral control algorithm because the dynamical model of the vehicle's position as a function of the yaw rate was a nonlinear equation.

While the lateral control algorithm was ultimately not tested on the vehicle, the simulated response of the system to a varying reference position is shown in Fig. 4. The response of the closed-loop system is stable in step response. While there is some overshoot, it is negligible compared to the width of an average road lane.

IV. CAR SETUP AND IMPLEMENTATION

The pedestrian brake controller was implemented on a vehicle for real-world experimentation. Table I consolidates information on the vehicle setup.

A. Drive-by-Wire System

The vehicle platform is a 2017 Lincoln MKZ Hybrid, fitted with a drive-by-wire system provided by Dataspeed [14]. ROS is used to communicate with the vehicle controller area network (CAN). The CAN system signals are decoded

using Dataspeed's algorithms, so information from the vehicle sensors can be read and commands can be sent to the vehicle from a computer. It is through the CAN system that we are capable of applying feedback control to drive the vehicle. The drive-by-wire system included a Logitech F310 joystick controller that can be used to electronically control the vehicle.

For safety purposes, one researcher was always present in the driver's seat to take control of the car if necessary. Manual inputs to the steering wheel or brakes were hard-wired to disable the drive-by-wire system immediately.

B. Vehicle camera setup

To mount the camera on the vehicle, a rack of extruded aluminum bars was attached to the roof of the vehicle. The bar that the camera was attached to was mounted such that it could be slid to the side of the vehicle in order to hold the camera at an offset distance from the side of the vehicle. This setup and the positioning of the pedestrian relative to the vehicle can be seen in Fig. 5. The purpose of this setup was to allow for safer testing with pedestrians. With the camera protruding from the side of the vehicle, the pedestrian could stand clear of the vehicle's trajectory while they were detected as if they were in front of the vehicle.

C. Pedestrian Detection Model

For the pedestrian detection and distance estimation from a single RGB camera feed, a monocular 3D detection model [15] was used. In this model, a pre-trained VGG network [16] without dense layers was used to regress 3D object properties. Then, these estimates were combined with the geometric constraints provided by a YOLOv5 [1] 2D object detector. This model provides the distance of the detected object which was then sent to the ROS longitudinal control node along with current car velocity to generate the appropriate brake commands. Fig. 6 shows a sample detection where the person is approximately 20 meters away from the camera.

D. Lane Detection Model

The YOLOP model [17] was investigated for its lane detection capabilities. YOLOP is a driving perception model that is capable of traffic object detection, drivable space segmentation, and lane detection. By default, the YOLOP demo program will output the drivable space and lane markings. In order to track the center of the lane, a Kalman filter was developed, and its output shown in Fig. 7 as the horizontal red line. The green color and red lane lines in the image were outputs of the default demo program. The center-of-lane detection works simply by finding the left and right bounds of the road and taking averages to find the center of the road and the center of the lane. While this was not used for a test, it may be useful for future work with the lateral control algorithm, where the vehicle is tasked with lane following.

E. ROS Setup

For this experiment, we used ROS1 in an Ubuntu 20.04 computer equipped with an Nvidia RTX 2070 GPU. For the

TABLE I: AV Platform Resources

Resource	Description
Vehicle	2017 Lincoln MKZ Hybrid
Drive-by-Wire	Dataspeed
Camera	Logitech C920 Webcam
Joystick Controller	Logitech F310
Computer Specs	Intel Xeon E5-2603 and Nvidia RTX 2070 GPU
ROS	Open source robotics library
Dataspeed ROS Driver	dbw_mkz_ros
Pedestrian Detection	YOLOv5 and 3D Object Detection

programming languages, both C++ and Python were used. One ROS node received the raw RGB images and published them as ROS "Image" messages which were then received by the pedestrian detection node. The object distance published by that node was sent to the longitudinal control node. The longitudinal controller as described in section III generates the brake level based on current velocity and distance. Generally, the greater the velocity, the earlier it generates brake command to avoid possible collision. The brake message was then converted to ROS "CAN" messages and sent to the Dataspeed CAN system.

V. RESULTS AND DISCUSSION

Experiments were designed to verify the operation of sub-components before testing with a pedestrian on the road. First, the ability of the controller to stop the car was verified by generating a synthetic pedestrian signal at a specified distance and verifying that braking resulted in stopping at the safety distance. After confirming the controller's ability to reliably stop, the pedestrian detection program was characterized, and finally a full system in-car pedestrian emergency brake test was performed.

A. Pedestrian Detection

The quality of distance measurement was evaluated with the vehicle stationary and having a moving pedestrian to test the accuracy of the detection at various ranges. The detection results are shown in Fig. 8. The pedestrian stood at 5, 10, 15, 20, and 25 meters from the vehicle camera for approximately 5 seconds each during this test.

As shown in the plot, the precision of the raw detection decreased as the distance from the camera increased, though noise was clearly evident throughout the entire test range. The noise is unsurprising since the pedestrian distance is estimated using a monocular 3D algorithm. However, this noise was problematic for the braking tests and is discussed below.

B. Braking Performance with Pedestrian

The control algorithm was tested in the vehicle setup described in Section IV. For this test, the vehicle was driven via the Logitech F310 controller to accelerate up to a speed of approximately 8.13 meters per second, or approximately 18 miles per hour, and the brake commands were generated independently by the longitudinal control algorithm. The results of this test were compared with simulated results shown in Fig.
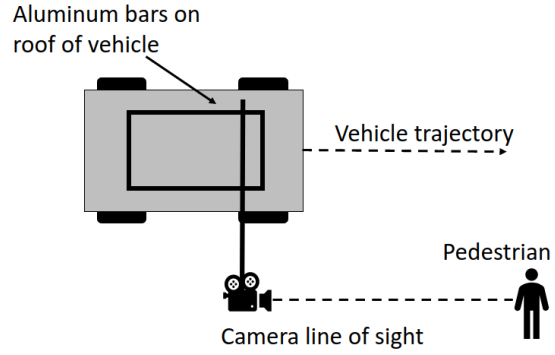
(a)   (b)

Fig. 5: (a) Picture of vehicle setup with camera mounted. (b) Diagram of test with pedestrian offset from vehicle trajectory.

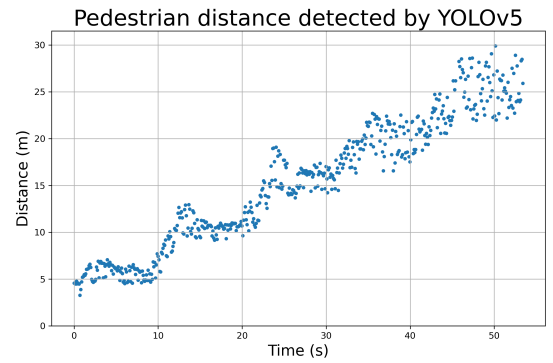Fig. 6: Sample pedestrian detection at ~20m.

Fig. 8: Raw pedestrian detection with stationary vehicle and moving pedestrian.
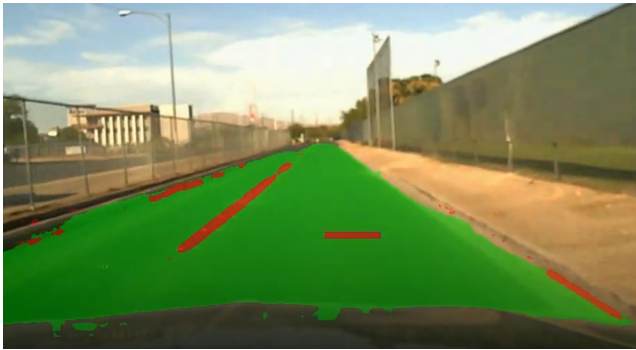

Fig. 7: YOLOP lane detection on UNLV campus road

9. In both the simulated and in-vehicle results, the vehicle was specified to stop 5 meters from the pedestrian.

As evidenced by the figure, the simulated case responds more quickly to the detected pedestrian and stops approximately one meter earlier than the in-vehicle test, at exactly five meters. In addition, the simulated response is smoother than the in-vehicle response. These are likely results of the

noisy pedestrian signal. The noise in the pedestrian distance signal during the in-vehicle test can be seen in Fig. 9a. Without a smooth detection of the pedestrian, the control algorithm will be unable to properly generate the appropriate braking command, which may be why the in-vehicle test stopped closer to the pedestrian than the simulated test.

In addition, while the in-vehicle velocity appears smooth and linear, the braking was not smooth and rapidly engaged and disengaged. The brake commands are compared in Fig. 9c, where the jittery, in-vehicle braking can be observed. This is most likely another problem introduced by the noisy detection. The comparison of the brake commands also shows that when the vehicle first starts braking, both the simulated and actual vehicle use a brake command of 0.4, but their accelerations shown in Fig. 9b do not match. When converting the brake force $u(t)$ into the command signal for the vehicle, the control algorithm uses information from Dataspeed that characterized the braking and acceleration of the vehicle. However, the discrepancy between the simulated and actual accelerations may warrant additional testing to verify the brake torque, or a more nuanced model of the vehicle braking.

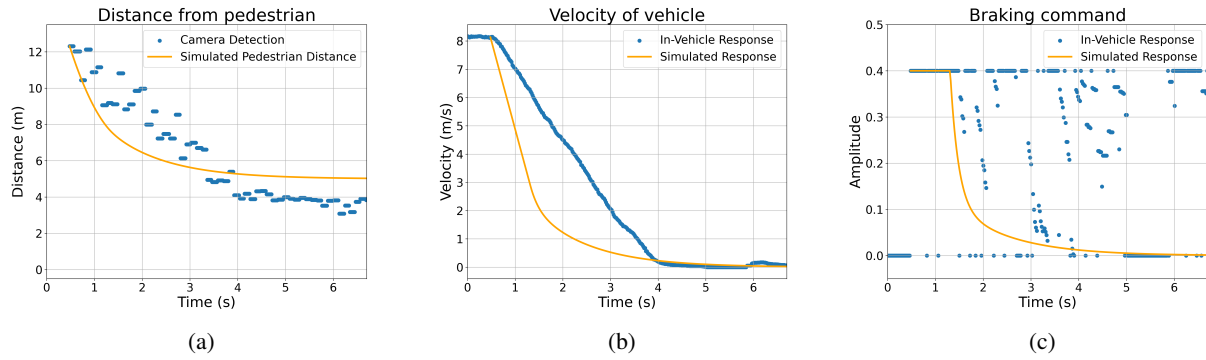Despite the issues introduced by the noisy input, the vehicle

Fig. 9: Comparison between in-vehicle and simulated response for braking test. (a) Distance comparison, (b) velocity comparison, (c) brake command comparison

was able to safely stop with ample distance from the pedestrian. In other tests, it was found that this stopping distance could be tuned to have the vehicle stop farther or closer to the pedestrian as well.

VI. FUTURE WORKS

For future work, the Lux LiDAR and Delphi ESR RADAR in the car will also be used in tandem with the RGB camera to more accurately estimate the pedestrian position. A tracking method will also be applied to more smoothly track the pedestrian distance. With a smoother and more accurate pedestrian detection, it will be easier to more rigorously test properties of the controller, like its reliability at various speeds and comfort in braking.

Looking beyond the longitudinal controller, future research will test the proposed lateral controller for its ability in lane following and implement it alongside the longitudinal controller for full autonomous control.

VII. CONCLUSION

By building off of the wealth of research in the field of AVs, a controller for pedestrian emergency braking was designed and tested in a matter of weeks. We began designing the controller by reviewing the various approaches to controller design and selected a PID approach for its simplicity and ease of implementation. After simulating the results and testing various components, we were able to implement it in a vehicle with a drive-by-wire system and demonstrated safe braking. In addition, a lateral controller was designed that will be useful for ensuring the vehicle can navigate lanes properly and increase the degree of autonomy.

REFERENCES

[1] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, and M. T. Minh, "ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference," Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6222936

[2] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "A model predictive control approach for combined braking and steering in autonomous vehicles," in 2007 Mediterranean Conference on Control & Automation, 2007, pp. 1–6.

[3] T. Stanger and L. del Re, "A model predictive cooperative adaptive cruise control approach," in 2013 American Control Conference (ACC), Washington, DC, 2013, pp. 1375–1379.

[4] J. Naranjo, J. Reviejo, C. Gonzalez, R. Rosa, and T. Pedro, "A throttle and brake fuzzy controller: Towards the automatic car," vol. 2809, 02 2003, pp. 291–301.

[5] S. Sharma, G. Tewolde, and J. Kwon, "Lateral and longitudinal motion control of autonomous vehicles using deep learning," in 2019 IEEE International Conference on Electro Information Technology (EIT), 2019, pp. 1–5.

[6] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8248–8254.

[7] P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu, and T. Mei, "Design of a control system for an autonomous vehicle based on adaptive-pid," International Journal of Advanced Robotic Systems, vol. 9, no. 2, p. 44, 2012.

[8] R. Marino, S. Scalzi, and M. Netto, "Nested pid steering control for lane keeping in autonomous vehicles," Control Engineering Practice, vol. 19, no. 12, pp. 1459–1467, 2011.

[9] D. Gomez, P. Marin-Plaza, A. Hussein, A. Escalera, and J. Armingol, "ROS-based architecture for autonomous intelligent campus automobile (icab)," UNED Plasencia Revista de Investigacion Universitaria, vol. 12, pp. 257–272, 2016.

[10] S. Gatesichapakorn, J. Takamatsu, and M. Ruchanurucks, "ROS based autonomous mobile robot navigation using 2d lidar and rgb-d camera," in 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), 2019, pp. 151–154.

[11] H. Guo, D. Cao, H. Chen, Z. Sun, and Y. Hu, "Model predictive path following control for autonomous cars considering a measurable disturbance: Implementation, testing, and verification," Mechanical Systems and Signal Processing, vol. 118, pp. 41–60, 2019.

[12] A. Jones, T. Le, V. Vishwanath, S. Singh, C. Uy, K. Murakami, S. Truong, D. Kashikar, and G. Lee, "Eec 193b autonomous car senior design final report."

[13] S. Antonov, A. Fehn, and A. Kugi, "A new flatness-based control of lateral vehicle dynamics," Vehicle System Dynamics, vol. 46, no. 9, pp. 789–801, 2008.

[14] Dataspeed, "Dataspeed adas development vehicle kit," 2022. [Online]. Available: https://bitbucket.org/DataspeedInc/dbw_mkz_ros/src/master/

[15] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," CoRR, vol. abs/1612.00496, 2016. [Online]. Available: http://arxiv.org/abs/1612.00496

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[17] D. Wu, M. Liao, W. Zhang, and X. Wang, "YOLOP: You only look once for panoptic driving perception," 2021.