2020 IEEE Intelligent Vehicles Symposium (IV)
October 20-23, 2020. Las Vegas, USA

Systems Integration, Simulation, and Control for Autonomous Trucking

Amir Darwesh[1], Grayson Woods[2], and Srikanth Saripalli[3]

Abstract— **This paper discusses a platform both in simulation and experimentation for testing autonomous heavy trucking. In simulation, we present a novel use of the video game American Truck Simulator (ATS) as the simulation platform, only costing a fraction of commercial simulator software. In experimentation, we present a modified ProStar 122+ using the PACMod system from AutonomouStuff, a popular by-wire kit. Discussion and review of the by-wire kit and sensors is provided. A proof-of-concept of the platform is shown by performing lane keeping at 65 mph using the Stanley lateral controller and MobilEye detection system. Further, we introduce a rapidly developed longitudinal control algorithm using a pedal actuation map, and 3D lookup tables created from braking and acceleration data. Introductory results are presented to aid the research community for single vehicle autonomous trucking.**

I. INTRODUCTION

A literature survey of Automated Trucking over the past 20 years has shown significant academic and or state/federal transportation institution sponsored research into truck platooning. Organizations and projects focused to platooning research include SARTRE, PATH, GCDC, SCANIA, Energy ITS, [1], [2] as well as several others. These projects have tested in simulation and or experimentation platooning of 3 or more heavy-trucks in which at least the longitudinal control is automated for the following vehicles using Vehicle to Vehicle (V2V) communications, while maintaining string stability in the platoon formation. Around 2012, these works had matured enough to gain industry traction in a commercial form of Connected Adaptive Cruise Control (CACC) including investments from Peloton Technology [3], Daimler [4], and Volvo Trucks [5]. Although research on platooning continues in academia and industry, disadvantages of L1-L2 platooning have slowed market adoption.

In early 2019 Daimler trucks dropped future projects involving L1-L2 platooning, citing that the fuel savings were "modest", and the monetary investment would be better suited towards L4 automation systems [6]. Though there is still active commercial development on L1/L2 platooning including ongoing work from Peloton Technology and Volvo Trucks, there is a new interest in mixed L1/L4 platooning where only the lead vehicle may necessitate a driver in certain Operational Design Domains (ODD). Peloton Technology announced in 2019 investment into mixed L1/L4 platooning in which only the leader vehicle demands a driver [7]. Similarly, Locomation.ai, another platooning company, is investing in L1/L4 platooning requiring drivers in all trucks, but with the ability for follower drivers to "clock out and rest" [8]. It is unclear however what the ODD is for L1/L4 platooning, specifically whether they would be able to break and form platoons autonomously.

Outside platooning for trucking automation, there has also been a push from industry focused on single truck L4 automation, where each truck is an individual agent. These companies primarily observe the financial incentive of human labor costs, and hence are focused on removing/reducing drivers via autonomous driving. Some strategies for achieving L4 automation include a decreased ODD, such as Embark's focus of on ramp to off ramp highway automation. Single truck L4 automations has seen early success in initial testing. Embark operated 124,062 miles between 2017-18 and had disengagement rates in Q4 2018 at only once per 1,392 miles [9].

As individual L4 trucking and L1/L4 platooning continue industry development, it follows that an intersection point will likely occur where both technologies can combine to offer human labor and fuel economy savings. By maintaining individual autonomy, trucks need not always operate in the platoon mode, broadening the ODD and market use. As industry continues research, federal and state transportation agencies are tasked with policy decisions and safety regulations. Compared to autonomous passenger vehicles, which has seen a plethora of publications, the authors have not found many similar works for autonomous trucking; however, there may be more research incentive now in L4 trucking, whether it be in platooning or individual trucking.

To aid future researchers and transportation studies interested in developing trucking automation algorithms, we review L4 ready platforms in both simulation and experimentation.

II. SYSTEM REQUIREMENTS IN HIGHWAY DRIVING

In this section, we detail major system requirements for automated highway driving.

1) Lane Detection: In L2 automations or above, lane centering of the truck and the attached trailer is required. The primary sensor for lane detection systems has been monocular cameras and are often placed in the front windshield or on top of the truck.

2) Trailer Pose Estimation: For control over the trailer articulation, relative orientation information is required; a method for a camera-based calculation is discussed in [10].

---

*We would like to acknowledge our CDL driver, Drew Hubbard, for his renewed assistance and patience during experimentation.

[1,2] Amir Darwesh and Grayson Woods are both graduate students in the Department of Mechanical Engineering at Texas A&M University `<adarwesh,graysonwoods>@tamu.edu`

[3] Srikanth Saripalli is an Associate Professor in the Department of Mechanical Engineering at Texas A&M University `ssaripalli@tamu.edu`

978-1-7281-6672-8/20/$31.00 ©2020 IEEE

3) Lateral Controller: With typical lane-widths on highways in the United States at 3.6 meters and truck widths of 2.6 meters, the lateral control algorithm must not exceed lateral errors of 1 meter or more than 2 degrees of heading error over one second at highway speeds.

4) Longitudinal Control: For Adaptive Cruise Control (ACC) applications, reliable control of the vehicle speed and station is required. In heavy vehicles, different loading configurations require adaptable control algorithms and or online model estimation, such as one used in [11].

5) Sensing & Communication Requirements: A single L4 capable heavy truck will require a wide sensing suite. Most experimental vehicles from the single truck automation industry use a wide array of sensors including a mix of monocular and stereo cameras, thermal cameras, LIDARs, automotive RADARs, and GPS+INS; however, sensors are usually confined to the cab to promote the business case. Use of LIDAR in autonomous driving has seen mixed opinions depending on application. In a trucking application, there may be less case for using LIDAR given highway speeds and truck stopping distances, as the detail of even long-range LIDARs is not sufficient [12]. Reliability issues are also a concern; in our personal experience, we observed LIDAR failure in 3-4 months due to excessive road vibrations. Cameras are quite popular in any autonomous driving application, as they are low cost and more reliable; however, they suffer in poor lighting/weather conditions. Thermal cameras compliment some short fallings of cameras due to their sensing ability at night and in fog conditions. Distance perception is a weakness in either monocular or thermal cameras; instead, stereo cameras are often used. Automotive RADARs are well-known sensors and can be used for obstacle detection up to 200 meters. RADAR is not able to classify trackings though, and is limited to 40-45 meters for pedestrian detection [13].

III. PLATFORM & HARDWARE

The hardware platform used in experimentation uses a modified International ProStar truck seen in Fig. 1, a Novatel GPS/INS, and a MobilEye detection system. Other sensors including a Delphi ESR 2.5 Radar, and Velodyne VLP-32c are available, but were not used for any of the experimentation presented here. For computing, an off-the-shelf desktop tower is used, along with several KVaser Leaf CAN-to-USB adapters to support CAN communications. The operating system runs Ubuntu 16.04 and uses the Robotic Operating System (ROS) [14] as the communications framework.

A. International ProStar 122+

1) Powertrain Specifications: For the powertrain, a difference compared to most trucks is the automatic Eaton transmission. Hence, the automation software and automatic gearing are independent systems, which has certain advantages and disadvantages. Benefits include system robustness, and a manufacturer tuned engine-gear map. The drawbacks are increased difficulty in obtaining transmission gear information, and one less control input method for the longitudinal dynamics.

Fig. 1: International ProStar 122+ with Sleeper Cabin

2) By-wire Kit: The modified 2013 ProStar 122+ includes by-wire functionality over the turn signals, throttle, braking, and steering. The truck was retrofitted by AutonomouStuff, and utilizes an EPAS Actuator by Allied Motion for steering and braking control. The braking system is actuated via a pulley cable system attached to the EPAS motor, as illustrated by Fig. 2. For steering, the EPAS motor is directly connected to the steering column. The throttle and turn signals are digitally controlled from PACMod by utilizing the ProStar's J1939 CANbus.

Though the input design for the steering and throttle are sufficient, it should be mentioned that the braking input design is not desirable. First, the brake pulley system actuates the brake pedal directly, introducing an additional mechanical failure mode. Second, the position of the brake pedal is not a preferred control input, as the same actuation of the pedal may not always yield the same braking torque due to temperature, air supply, and slack developed in the pulley cable. It is suggested instead that the actuation should be designed to control brake pressure, which can achieve a braking torque or deceleration with higher repeatability.

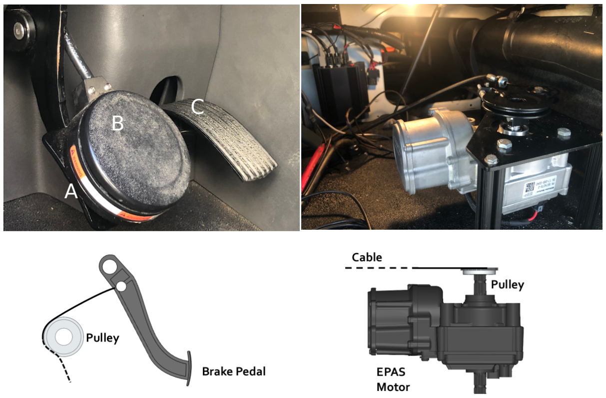


Fig. 2: Brake Pedal Pulley Diagram with (A) Brake Pedal, (B) Disengagement Safety Switch, and (C) Throttle

B. Novatel SPAN System

For odometry, the Novatel SPAN system is utilized, which includes a ProPak 6 GNSS receiver, two VEXXIS GNSS-500 antennas, and an IMU-IGM-S1 module. GPS/INS information is provided over Ethernet to the computing platform and is logged at 50 Hz, while IMU data is published at 125 Hz.

1) Antenna Placement & Configuration: The two antennas are mounted on either side of the truck, located on top of the side view mirrors. The IMU module is mounted centrally inside the cab. To setup the Propak 6, offset measurements from the antennas and IMU are needed. Due to the large size of the ProStar, it is difficult to obtain high measurement accuracy. Offsets were taken using a laser distance tool to the best of our abilities, but uncertainties were around 10 cm, which is then propagated into the reported uncertainty from the Novatel driver.

2) Accuracy: To increase the accuracy from the GPS+INS solution, wheel speed is provided to the Novatel ROS driver from PACMod. Repeated tests were performed with and without wheel speed information supplied. The most notable decrease was in the yaw uncertainty, with a reduction of standard deviation by over 50%. Other methods to increase GPS/INS solution accuracy include a GPS correction subscription service such as TerraStar; however, with consistent uncertainties of around 24cm in open sky conditions, the corrections subscription did not justify the high costs.

C. MobilEye 630

The MobilEye 630 with extended logging features, provided by AutonomouStuff, is utilized as the main lane detection method during experimentation. The MobilEye updates at 10 Hz, and provides information of the lane, lead vehicle, speed limit signage, and any pedestrians present.

1) Configuration and Placement: The MobilEye detection system requires that the vehicle speed must be provided, usually via CAN from the vehicle's ECU. However, speed information was provided over CAN from PACMod instead for wiring simplicity. The MobilEye was centered inside near the of the top windshield, with the camera aimed such that just the tip of the hood was visible in the setup software.

2) Lane Information: Lane information is presented as coefficients c_n for cubic polynomials in the form of Eq. (1):

$$y(x) = c_3x^3 + c_2x^2 + c_1x + c_0 \tag{1}$$

where x and y are the longitudinal and lateral directions of the vehicle, respectively. Hence, the lateral offset between the left and right lanes can be obtained by summation of the c_0 coefficient. Additionally, the MobilEye reports confidence on the lane information represented in three levels. Figs. 3c and 3d show excellent (green), fair (yellow), and poor (red) lane detections. Corresponding Figs. 3a and 3b show images of the lane in successful and partial failure detections. To combat partial failure lane detection, a bias algorithm was implemented to reject a lane coefficient if the confidence was less than 2, and use instead a previous estimated width of the lane, as shown in Algorithm 1.

3) Calibration: Since the MobilEye is difficult to perfectly center, two different methods to calibrate an offset between the left and right sides were used. First, the distance from the center of the camera to the outer wheels was measured on either side. Second, the left and right coefficients c_0 were measured while the truck was aligned to touch either the left or right lanes. These calibration values are then also included in the summation for the mid-lane offset, as shown in Algorithm 1.

(a) Separate camera image from successful detection

(b) Separate camera image from partial failure lane detection

(c) Successful Detection (d) Partial Failed Detection

Fig. 3: Lane Detection Success and Failures

Algorithm 1: Mid-lane offset algorithm

```
input  : Lane coefficients $c_{0,l}$ $c_{0,r}$
         Lane Confidence $\mu_l$ $\mu_r$
         Offset Calibration $\delta_l$, $\delta_r$
         Lane Width LaneWidth
output: Mid-lane offset $e_y$
if $\mu_r >= 2$ and $\mu_l >= 2$ then
    $e_y = c_{0,l} + c_{0,r} + \delta_l + \delta_r$;
else if $\mu_r >= 2$ then
    $e_y = c_{0,r} - \mathsf{LaneWidth}/2 + \delta_r$;
else if $\mu_l >= 2$ then
    $e_y = c_{0,l} + \mathsf{LaneWidth}/2 + \delta_l$;
else
    return error
end
return $e_y$
```

IV. SIMULATION

A. American Truck Simulator

Unlike car simulators, there are fewer options available for simulating a heavy-truck vehicle. Typical software used in research and academia include TruckSim [15], ASM Truck/Trailer by dSpace [16], and Truckmaker [17]. Though these simulators offer advanced and configurable dynamic simulations, they are less accessible due to large investment costs. In the Autonomous Driving community, Grand Theft Auto V (GTA 5) has seen popularity as a low-cost car simulator [18], as well as CARLA [19]. Similarly, we introduce another video game, American Truck Simulator (ATS) [20], as a low cost simulation platform. ATS simulates an 18-wheeler truck, where players can emulate a truck-driver. ATS simulates the truck engine, transmission, brakes, suspension, and even road traction. Although these simulations are mostly not configurable, and the dynamic models simulated are not publicly available, ATS is advantageous in its low cost and hardware requirements.

B. *Simulation Interface*

Interface to the simulation is made possible through a custom Telemetry SDK plugin installed in the game. The plugin has been configured to publish TCP packets of the vehicle and truck's stateful information, which includes, Position & Orientation, Linear & Angular Velocities, Linear & Angular Accelerations, Engine Gear & RPM, Effective Braking, and Throttle & Steering values.

The packets are parsed by a custom made ROS wrapper, and are converted into standardized ROS messages. The simulator does not provide any world information - lanes positions, construction zones - or information of other vehicles on the road.

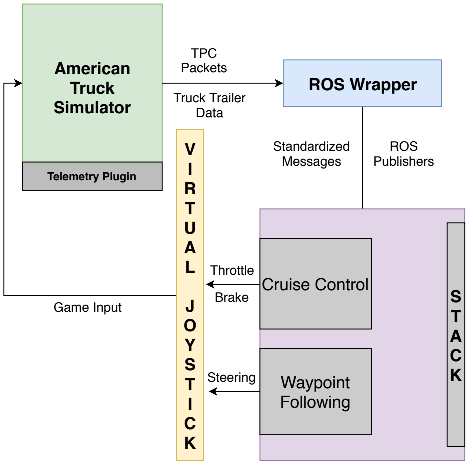


Fig. 4: American Truck Simulator Communication Flow

C. *System Response Comparison*

With the same input sequence, both the ProStar and ATS truck have similar first order lag responses to the pedal command, shown in Fig. 5. Similarly to the Prostar, throttle and brake pedal inputs are given over the range of 0-1, which corresponds to the percent of pedal deflection. Differences between the two responses can be attributed to the simplified dynamic model, and a different engine/transmission used in game.

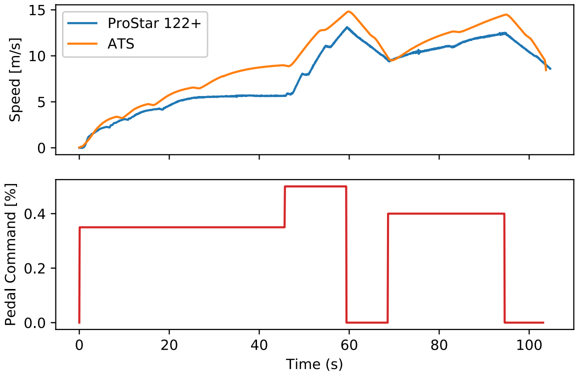


Fig. 5: Longitudinal Response of ProStar 122+ and ATS

Another difference in simulation is the actuation delay between the pedal and subsequent acceleration response. These actuation delays affect the controllability of the system, and can be seen in the longitudinal control performance in Section V.

Though there are quite a few more differences in simulation, ATS still offers a valuable platform to develop autonomous driving functions due to ease of testing. Further, the ROS wrapper developed for ATS utilizes the same input/output topics as the PACMod module on the Prostar, which provides easy transition from simulation to experimentation.

V. CONTROL

A. *Lateral Control*

For lateral control, the Stanley Controller [21] was implemented, shown in Eq. (2). The steering control law is readily programmable, as the MobilEye both provides the heading error $\theta_e(t)$, and lateral error from the mid-line of the lane $e_y(t)$, as calculated in Algorithm 1. A k_{soft} gain is implemented to avoid over steering at velocities less than 1 m/s.

$$\delta(t) = \theta_e(t) + \arctan\left(\frac{k_p e_y(t)}{v_x(t) + k_{\text{soft}}}\right) \tag{2}$$

In experimentation, the straight route (highlighted in yellow in Fig. 6) yielded less than 30 cm lateral error testing at speeds of 65 mph. While testing in turns, circled red in Fig. 6, severe instabilities from Stanley controller were observed due

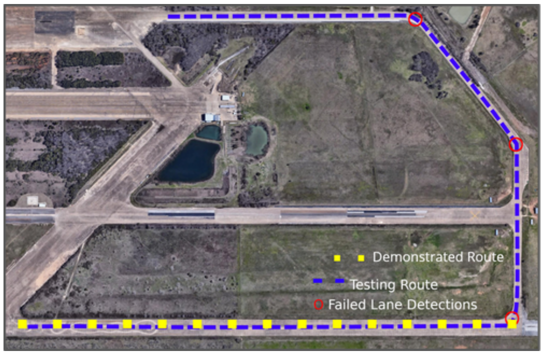


Fig. 6: RELLIS Testing Facility & Lane Keeping Route

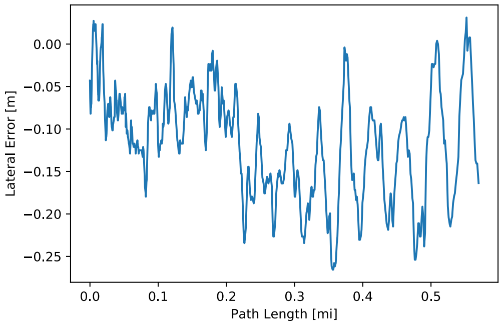


Fig. 7: Lateral Error reported by MobilEye with Stanley Controller during Demonstration Route

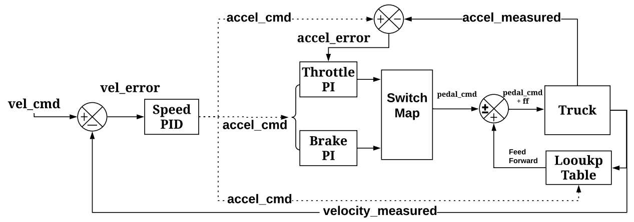


Fig. 8: Longitudinal Control Flow Chart

to failed detections (Figs. 3c and 3d) from the MobilEye. For simulation, experiments over a straight lane scenario yielded errors close to near zero. Without disturbances, such as wind and uneven road surfaces, the Stanley control was more successful.

B. Longitudinal Control

A PID controller is applied to the error from the desired and current velocity, generating a desired acceleration. Next, two separate PI controllers with feed-forward terms are used for the throttle and braking with acceleration error as the input. The feed-forward terms are generated by a mapping between the pedal actuation, measured velocity, and acceleration responses in the form of a lookup-table. The longitudinal control algorithm, shown in Algorithm 2, outputs the pedal commands that are sent to PACMod for actuation. A control flow diagram is illustrated in Fig. 8.

Algorithm 2: Longitudinal Control Algorithm

```
input : Desired Speed $v_{cmd}$
        Current Speed $v_{act}$
        Current Acceleration $a_{act}$
output: Pedal Commands $u_{thr}$, $u_{br}$
$a_{cmd}$ = speedPID($v_{cmd}$, $v_{act}$);
$case$ = switchMap($a_{cmd}$, $v_{act}$);
if $case$ == $throttle$ then
    $u_{thr}$ = thrPI($a_{cmd}$, $a_{act}$) + thrMap($a_{cmd}$, $v_{act}$);
    $u_{br}$ = 0;
else if $case$ == $coast$ then
    $u_{br}$ = 0;
    $u_{thr}$ = 0;
else
    $u_{thr}$ = 0;
    $u_{br}$ = brakePI($a_{cmd}$, $a_{act}$) + brakeMap($a_{cmd}$,
      $v_{act}$);
end
return $u_{thr}$, $u_{br}$
```

1) Pedal Switching: To reduce chatter between the brake and throttle actuation, a throttle-brake-coast switching map was generated as shown in Fig. 9. The actuation map is based on deceleration measurements taken during coasting at various speeds, and is fitted into a parabolic function. Next, an upper and lower bound were added to the parabolic function to generate an actuation map of applying either (or neither during coasting) the throttle or brake. If the PI controller requires a deceleration, the switching map is used to determine the appropriate pedal actuation type. The switching map also enforces no simultaneous pedal action.

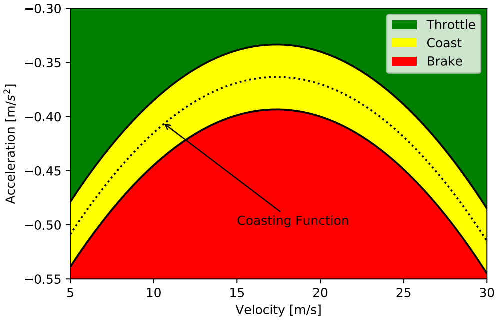
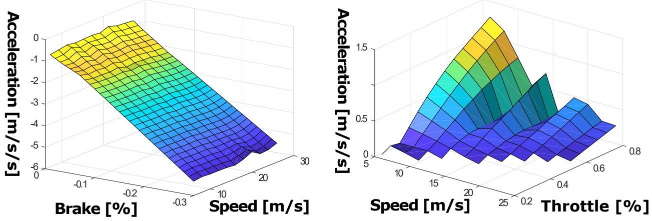


Fig. 9: Deceleration Actuation Map



(a) Brake Pedal Mapping

(b) Throttle Pedal Mapping

Fig. 10: Simulation Brake/Throttle Mapping

2) Throttle & Brake Mapping: Creating a dynamic model of the powertrain is often a difficult task, requiring either manufacture information on the ECU, or collection of large datasets of the transmission gear, RPM, and wheel velocity to estimate drive-train parameters [22]. Because the information of transmission gear and RPM is not known in experimentation, an approach similar to [23] is taken where throttle and braking pedal deflection percentages are mapped to both the current vehicle velocity and measured acceleration. Throttle and braking data are then binned and a 3D look-up table is created with inputs as desired acceleration and current velocity, and the output as a predicted pedal command. Two tables are created for both throttle and braking and are visualized in Figs. 10a and 10b.

3) Speed Tracking: The longitudinal controller was successful at tracking the desired speed profile in Fig. 11, which shows both the ProStar 122+ and ATS tracking over a series of speed commands ranging from 0-10 m/s. The throttle and braking look-up tables were advantageous in supplementing the PI pedal controllers; however, further tuning and implementation of a gain scheduler is required to remove significant overshoots.

VI. CONCLUSIONS

Given the current trends of industry development in autonomous trucking and mixed L1/L4 platooning, we anticipate that more research works will take place in academia and transportation institutions, as has been for autonomous passenger vehicles. To provide reference to for the autonomous trucking community, we reviewed platforms ready

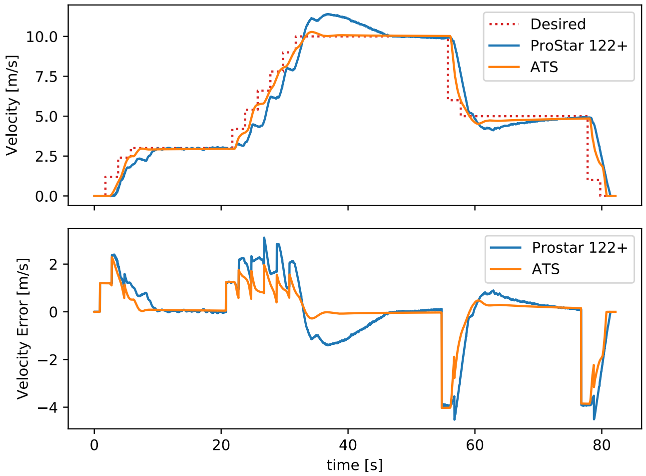


Fig. 11: Longitudinal Tracking of ProStar 122+ and ATS

for L4 in both simulation and experimentation. The simulation platform, American Truck Simulator, is available at a low cost unlike most other truck simulators. A comparison between the longitudinal response in simulation and experimentation is provided, and comparable results are found for the longitudinal dynamics.

For experimentation, a ProStar 122+ retrofitted with PACMod, a popular by-wire kit popular within the autonomous driving community, is used. Actuation of the brake, throttle, and steering is discussed, and significant shortcomings of the braking actuation method are mentioned. Sensors on the trucking platform include the Novatel GPS SPAN System, and MobilEye 630. In the Novatel SPAN System, uncertainties in open sky conditions for the position and yaw were near 24 cm and 0.11 rad, respectively. We reduced yaw uncertainty by over 50% by simply supplying wheel encoder information to the Novatel driver. The MobilEye lane detection system was largely successful in detecting lanes, however, in the exiting of curves, or when sun glare faced the MobilEye, partial or complete failures were observed.

Lastly, introductory control algorithms were tested to guide the vehicle within a lane while maintaining speed as a demonstration of success for the by-wire ProStar and simulation platforms. With limited information over transmission gear and engine RPM, a calibration lookup table was generated for both the brake and throttle. These lookup tables were used as feed-forward components in PI controllers for the pedals, and were advantageous in that they required significantly less time and programming complexity compared to developing dynamic models of the powertrain. However, the longitudinal controller was not precise, and had overshoots over 2 m/s. For the lateral controller, the Stanley controller introduced by [21] was implemented due to its ease of tie in with the MobilEye output. Lateral errors were observed to be less than 30 cm when testing in straight lanes at 65 mph.

Future work of this project will include development into a better braking system, further longitudinal control tuning, and evaluation of other lane detection systems.

REFERENCES

[1] S. Tsugawa, S. Jeschke, and S. E. Shladover, "A review of truck platooning projects for energy savings," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 68–77, 2016.

[2] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," in *Proceedings of the 19th ITS World Congress, Oct 22-26, Vienna, Austria (2012)*, 2012.

[3] R. Lockwood, "Truck platooning, past, present, and future," Apr 2016. [Online]. Available: https://www.truckinginfo.com/156677/truck-platooning-past-present-and-future

[4] "Daimler trucks tests truck platooning on public highways in the us," Sep 2017.

[5] "Volvo car corporation concludes following the sartre project: Platooned traffic can be integrated with other road users on conventional highways," Sep 2012.

[6] J. Park, "Daimler nixes platooning, focuses on automation," Jan 2019.

[7] "Peloton announces its vision for the trucking industry: Drivers lead, and technology follows," Jul 2019.

[8] "Full autonomy driver augmentation." [Online]. Available: https://locomation.ai/

[9] "Embark disengagement report," *Embark Trucks*, 2018. [Online]. Available: https://embarktrucks.com/files/Embark_Disengagement_Report_2018.pdf

[10] C. Fuchs, F. Neuhaus, and D. Paulus, "Advanced 3-d trailer pose estimation for articulated vehicles," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 211–216.

[11] H. S. Bae and J. C. Gerdes, "Parameter estimation and command modification for longitudinal control of heavy vehicles," 2003.

[12] K. Tiwari, "Why our company's trucks won't carry lidar," Nov 2018.

[13] A. A. Belyaev, T. A. Suanov, I. O. Frolov, and D. O. Trots, "The range of pedestrian detection with automotive radar," in *2019 Radiation and Scattering of Electromagnetic Waves (RSEMW)*, June 2019, pp. 432–435.

[14] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[15] R. Grace, A. Guzman, J. Staszewski, B. Peters, M. Mallis, and D. Dinges, "The carnegie mellon trucksim: A tool to improve driving safety," in *17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No. 98CH36267)*, vol. 2. IEEE, 1998, pp. I35–1.

[16] "Asm truck." [Online]. Available: https://www.dspace.com/en/inc/home/products/sw/automotive_simulation_models/produkte_asm/vehicle_dynamics_models/asm_truck.cfm

[17] I. A. GmbH, "Truckmaker," Jan 2020. [Online]. Available: https://ipg-automotive.com/products-services/simulation-software/truckmaker/

[18] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*. Springer, 2016, pp. 102–118.

[19] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[20] Scs, "American truck simulator." [Online]. Available: https://americantrucksimulator.com/

[21] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing," in *2007 American Control Conference*. IEEE, 2007, pp. 2296–2301.

[22] X.-Y. Lu and J. K. Hedrick, "Longitudinal control design and experiment for heavy-duty trucks," in *Proceedings of the 2003 American Control Conference, 2003.*, vol. 1. IEEE, 2003, pp. 36–41.

[23] F. Zhu, L. Ma, X. Xu, D. Guo, X. Cui, and Q. Kong, "Baidu apollo auto-calibration system-an industry-level data-driven and learning based vehicle longitude dynamic calibrating algorithm," *arXiv preprint arXiv:1808.10134*, 2018.