Remote driving testbed with force feedback based on slip angle estimation

Quang Son Le, Shai Arogeti, *Member, IEEE*, and Avinoam Borowsky

***Abstract*—While automotive research plays a significant role nowadays, most of the experimental activity demands costly platforms and involves safety issues. Plenty of driving simulators were proposed to reduce the costs and guarantee safety. However, they still cannot reflect the physical world, resulting in subjective assessments in any aspect of the study. This paper introduces an affordable remote driving testbed based on small-scale car-like mobile platforms and a physical road. The driver in the remote driving station observes a real-time video taken from a front-facing camera installed in the car. For a realistic driving experience, we have developed a torque feedback mechanism based on the small-scale car motion to mimic the influence of the physical linkage between the front wheels and the steering wheel of a standard car. This mechanism demands knowledge of the car's side-slip angle that is not directly measured. Here, we introduce a supervised learning-based combined regression model (RidgeCV and Bootstrap aggregating decision tree) that estimates the side-slip angle for highly non-linear behavior.**

***Index Terms*—Remote driving testbed, Torque feedback, Side-slip angle, regression, Ridge cross validation, Bagging decision tree (Bootstrap aggregating)**

I. Introduction

Automotive industry witnesses a huge impact from the digital era along with supported interdisciplinary research. For any type of driving experiment, a physical vehicle and a driver are generally needed. Besides the cost of using an actual vehicle for experiments, the driver's safety (or other participants) is a subject of concern. Driving simulators based on vehicle models and graphical animation are the common solution. They provide a safe environment to study traffic dynamics that involve human behavior [3]. Nevertheless, its ability to provide drivers with the feeling of real-world driving is still limited [4]. Although many of the simulation aspects were improved to make the environment more realistic (such as integrating Virtual Reality technology), the virtual world is still "simpler" than the real world. Additionally, high-resolution graphics increase the processing load drastically [28]. Another important factor is that the interaction with autonomous cars is only simulated based on computational models which may not be realistic, especially due to the rapid progress in this field [30]. In order to resemble realistic vehicle dynamics, efforts were made to develop vehicle models that integrate into a virtual reality simulation [3], [22]. However, these models still need to be validated in real driving [22] conditions. This is reasonable, because the model, regardless of its complexity cannot represent the real physical world exactly with its nonlinear and uncertain nature.

This paper introduces a new approach to experiments involving human driving that is affordable and safe but still reflects the physical effects of actual driving. Our testbed includes small-scale car-like mobile platforms, a small road and remote driving stations. The car is a 1/10th model of a real car, and the remote driver sees a video taken by a camera installed in the moving car. The goal is to provide the drivers with a driving environment that resembles the driving experience of a real car.

Due to the mechanical decoupling of the remote steering wheel and the car's front steering mechanical system, the remote driver does not experience the forces acting on the steering wheel due to the car's motion. The same problem characterizes other steer-by-wire systems [17], and it is generally solved by adding a torque feedback actuator at the steering wheel side. The actuator applies forces that mimic the forces the driver experiences when steering the front wheels mechanically (as is the case of a standard car). Torque feedback plays a significant role in actual driving [1]; it allows better curve negotiation [5], provides the driver with an improved orientation in tight turns [7] and adds stability to lane-keeping controllers [19]. For these reasons, a torque feedback mechanism at the steering station is needed. However, calculating the required torque from a conventional car model is based on the car's side slip angle that is not measured directly [1]. The vehicle side-slip angle describes the direction of the car's velocity at its center of gravity, measured with respect to the car frame. It is defined as follows,

$$\beta = -\arctan(\frac{v_y}{v_x}). \tag{1}$$

where v_x and v_y are the longitudinal and lateral velocity elements.

The side-slip angle is a key state variable that is utilized for vehicle stability control [6] and driver assistance systems design [11]. Also for autonomous cars, slip angle is a significant variable needed for steering control [20], Robust anti-sliding control [15], and path following [16], [26]. It is prerequisite information used in the calculation of the tire forces, which eventually determines the motion of the car [32]. Nevertheless, professional devices that can provide side-slip

Quang Son Le is with the department of Industrial and Management Engineerin, Ben-Gurion University of the Negev, Israel, (e-mail: officialquangsonle@gmail.com).

Shai Arogeti is with the department of mechanical engineering, Ben-Gurion University of the Negev, Israel, (e-mail: arogeti@bgu.ac.il).

Avinoam Borowsky is with the department of Industrial and Management Engineerin, Ben-Gurion University of the Negev, Israel, (e-mail: borowsky@bgu.ac.il).

angle information are expensive or based on GNSS technology that is not available indoors [32]. Therefore a low cost and effective side-slip angle estimator is essential [27].

To support our driving simulator, we have developed a data-driven side-slip angle estimator that is based on onboard simple sensing devices. The estimator is trained by zigzag driving that reflects high slip angles and nonlinear behavior (which makes the estimation more challenging compared to estimation in linear conditions [29]). The obtained side-slip angle is used to calculate the needed torque feedback at the driver steering wheel in real-time conditions. As the testbed is deployed indoors, a simple approach is suggested for side-slip angle measurements in the learning phase, based on tiny onboard Lidars.

There are three contributions to this study.

1) A new approach to driving simulators, based on remote driving and small-scale physical equipment. The testbed was designed for minimal latency in the transmission of control commands and video streaming, to provide a real-time experience.
2) A data-driven side-slip angle estimator for nonlinear driving conditions.
3) An approach to apply torque feedback in remote driving stations, based on steer-by-wire technology.

The structure of the paper is as follows. It starts with the description of related work in section II, divided into three subsections according to the three mentioned contribution items. The proposed methodologies are given in section III, while results and summary discussions are presented in section IV. Then, section V concludes the paper.

II. Relevant studies

A. Driving testbeds

The idea of a laboratory small-scale testbed is already known in the field of autonomous driving [8], [18]. Filho et. al [13] implemented the mixed platoon situation (both manually driven and autonomous cars together), but its manually driven cars (e.g., th leader of the platoon) was in fact driver-less.
Liu et. al [24] conducted a remote manually driving experiments under LTE network to evaluate the influence of latency on driving performance. The authors of [24] showed that the latency, if high enough, significantly affects driving activity; hence it is advisable to minimize latency (e.g., by a 5G network). Nevertheless, Liu et. al [24] does not specify how they implemented the torque feedback, which also plays a significant role in the driving activity.

B. Torque feedback

From the work of Balachandran et. al [1], the steering motor torque can be directly fed back at the driving station. It is indeed a realistic feeling but this torque depends on uncertain hardware properties, if it is not measured directly. Another approach exploits physical model abstraction (of a spring, in this case) to create torque feedback that is speed dependent. However, Balachandran et. al [1] mention these approaches as low-fidelity as they cannot capture all elements of the steering feel, but they agree with their relatively easy implementation. Higher fidelity models on the other hand, as suggested by Balachandran et. al [1], can capture better the elements of the steering feel but hard to tune due to the many model parameters involved.

A recent idea in [10] aims to estimate torque feedback from a tire brush model with the information of the steering rate. Even though its need for low-speed conditions and the vertical load distribution, it has shown a promising direction, indicating that supervised machine learning techniques can be involved in estimating torque feedback of a conventional car for a steer-by-wire system.

In our study, the model of [1] was adapted since it adequately restores the steering feel and the needed information is accessible for real-time implementation. The main challenge is the side-slip angle that [1] obtains from a GPS-based estimator, which is not feasible for indoor conditions. This motivates the following subsection about side-slip angle estimation.

C. Nonlinear slip angle estimation

The classical approach to side-slip angle estimation is based on the system dynamical model and a Kalman filter. The linear observer with a robust gain matrix proposed by Aoki et. al [33] has shown a remarkable result, even for relatively high slip angles of about 20 degrees (which reflects nonlinear behavior). Also Phanomchoeng et. al [14] suggested a nonlinear observer with an impressive performance for high slip angles (up to 20 degrees). Such estimation algorithms are powerful and demonstrate how physics knowledge can be utilized for estimation in nonlinear conditions. But they are challenging for design and need extensive knowledge of system models [29]. Particularly, some of the needed information is not available in our laboratory testbed. Furthermore, our over-steering slip angle can reach 1 radian (about 60 degrees) in the tested conditions, which were not covered by the last-mentioned articles.

An alternative to physical models is based on data to create data-driven models. Although Graeber et. al and Abdulrahim [23], [29] with the traditional and recurrent network achieve well-fitted curves of slip angle estimation, we assume that complex models based on neural networks could be computationally expensive for real-time performance. If the torque feedback is generated with phase lag, it has a negative effect on steering stability and consequently the driving feel is not realistic. Hence, we aim at other simpler supervised learning-based regression. In [12], a simple linear regression reaches an acceptable error of slip angle estimation. On the other hand, [12] worked with a relatively small slip angle and close to linear behavior, which is not suitable for our needs. While the simple model of [12] has worked satisfactorily in simulations, its realistic performance in the real world was not tested.

In the method suggested in [29], slip angles of about 40 degrees were estimated in snow conditions. Automotive drifting experiments with slip angles that reached 60 degrees are reported in [23]. The slip angle was estimated by a neural network with a maximum error of about 20 degree (about 0.4

radian). More important, the estimator of [23], [29] can capture the behavior of the slip angle variable with a limited error.

Inspired by [23], [29], we are suggesting a regression model that can estimate large side-slip angles that can reach 60 degrees (about 1 radian). By that, we capture slip angle behavior without the use of a neural network that could pose a challenge for real-time applications. Our goal is to achieve a maximum error similar to [23] while allowing real-time processing. The driving maneuver we perform for learning resembles a zig-zag motion and with the slow longitudinal velocity, generates high slip angles and over-steering conditions. Such a maneuver makes the side-slip angle behavior highly nonlinear [29], and the estimation more difficult.

III. Methodology and Design

A. Driving testbed

Our design is depicted in Fig.1. The car is a 1/10 RC car model with two control boards. An Arduino Mega is directly connected to peripheral devices (such as the driving motor, steering servo and all sensors), and through a bilateral (UART) communication link exchanges information with a Raspberry 4 (Pi4) single board computer. The Pi4 collects streaming video from a front-facing Pi V2.0 camera and exchanges data with the remote driving station via a 5G WiFi network. Streaming is processed on open Source Gstreamer API to ensure lowest latency. At the driving station, a Logitech Gaming G29 steering wheel, pedals, and dedicated driving seat are installed to make it a car-like driving station.

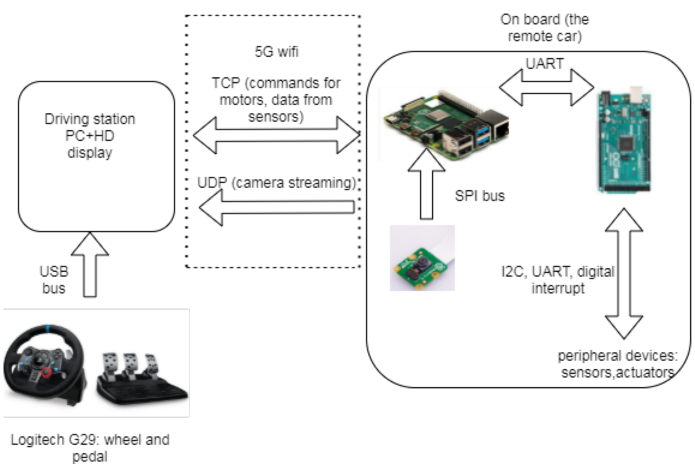

Fig. 1. Testbed design layout

B. Torque feedback calculation

The calculation of the applied torque feedback (notated by τ_{motor}) is based on the following model from [1],

$$\tau_{motor} = \tau_{damp} + \tau_{inertia} + K\tau_{assisted\ tire\ moment}$$

where,

$$\tau_{inertia} = -\Delta J\ddot{\delta}_{rw},$$

$$\tau_{damp} = -\Delta b\dot{\delta}_{rw},$$

$$\tau_{assisted\ tire\ moment} = (\tau_{jack} + \tau_{align})W_f,$$

and,

$$\tau_{jack} = \begin{cases} -k_{db}\delta_{rw} \text{ if } |\delta_{rw}| \leq \delta_{db} \\ -k_{jack}\delta_{rw} - k_{db}(sgn(\delta_{rw})\delta_{db}) \text{ if } |\delta_{rw}| > \delta_{db}. \end{cases}$$

Here, k_{jack}, k_{db} and δ_{db} are the jacking torque spring constant, the dead band spring constant and the steering dead band.

The following relations are involved in the torque feedback calculation,

$$\tau_{align} = -F_{y_f}(t_m + t_p),$$

where,

$$F_{y_f} = -C_{\alpha_f}\tan\alpha_f(1+\frac{C_{\alpha_f}^2}{3\mu F_{z_f}}|\tan\alpha_f| - \frac{C_{\alpha_f}^3}{27(\mu F_{z_f})^2}\tan^3\alpha_f)$$

is the front wheels lateral force,

$$t_p = t_{p0} - sgn(\alpha_f)\frac{t_{p0}C_{\alpha_f}}{3\mu F_{z_f}}\tan\alpha_f$$

is the pneumatic trail,

$$W_f = e^{\frac{-\alpha_f^2}{2\sigma_{ps}^2}}(1-\gamma) + \gamma$$

is the weighting function, and α_f is the front tires slip angle.

The front tire slip angle α_f and the vehicle side-slip angle β are related in the following way

$$\alpha_f = \tan^{-1}(\beta + \frac{ar}{v_x}) - \delta_{rw}, \tag{2}$$

where δ_{rw} is the front wheel equivalent steering angle, v_x is longitudinal velocity, r is yaw rate, and a is the distance from the center of gravity to the front axle.

In the following, we list down the design parameters that need to be tuned to achieve the desired torque feedback.

- ΔJ is the inertia change.
- Δb is the damping change.
- t_m is mechanical trail and t_{p0} is pneumatic trail at zero front slip angle.
- C_{α_f} is the combined front tires cornering stiffness, μ is the surface coefficient of friction, and F_{z_f} is the front tires vertical load.
- σ_{ps} and γ are the standard deviation and the lower limit of the weighting function.

For our design case:

- $\Delta b = 1Nm/rads^{-1}, \gamma = 0.5, K = 0.7$
- $k_{jack} = 3Nm/rad, C_{\alpha f} = 1300N.rad^{-1}$, reflecting the fact that the mass of the small-scale car is much smaller than the mass of the actual car in [1].
- ΔJ, if increases, will reduce the on center capability and effective stiffness of the wheel (see [1]). The second order derivative of δ_{db} was set to zero in [1] (due to

numerical difficulties). Here, after turning, we set it to $0.002 Nm/rads^{-2}$.

- t_m is 10mm
- μ is set as 0.85 (similar to [1] and extracted from [2])
- t_{p0} is difficult to be measured and no information is available regarding the remote car. Ref. [21] shows results of curve fitting of side slip tests with a "Jilin" tire that if the load is $4500N$, t_{p0} is about 0.025, therefore we set t_{p0} to 0.025mm while F_z is $150N$ (to fit the small scale remote car).
- δ_{db} is suggested to be 50% of the entire steering range [2], so it was set to $\frac{\pi}{12}$ rad (with the range from $\frac{\pi}{6}$ to $\frac{\pi}{6}$).
- k_{db} is suggested to be close to k_{jack} but with $k_{db} < k_{jack}$, so k_{db} was set to 2.8Nm/rad.
- σ_{ps} is selected as 0.01 to fit a typical weighting function (see [1]).
- a is taken as half of the vehicle's wheelbase, that is $0.11[m]$ for the case of the small-scale car.

The yaw rate r is measured by an IMU, while the equivalent steering angle δ_{rw} is assumed a linear function of the steering wheel angle that Logitech SDK reports. Lastly, the front tire slip angle α_f is a function of the vehicle side-slip angle β (see (2)) that is estimated by the method suggested in the following sub-section.

C. Real-time nonlinear slip-angle estimation

As stated before, we look for a regression model that is simple, but can still represent the non-linear behavior of high slip angles. To limit the computational load in real-time, the selected approach is not based on a Neural Network.

1) Decision tree regression with CART algorithm: The Classification and regression trees (CART) [9] is a widely used algorithm to solve both classification and regression based on a decision. CART goes through all samples of every feature in the data-set with the option of "splitting" the tree into different nodes for certain indices. It then finds the index that its splitting contributes the best performance metric, which is confirmed as the "splitting" index. That process is repeated at split nodes (and sub-nodes if they exist) separately until the stop criteria are satisfied. Then the tree is formed.

The difference between regression and classification, when implementing the algorithm, is the performance metric. Minimization of a Gini index for classification [31] and minimization of the error between the predicted and the ground truth data for regression; sklearn (also known as Scikit-learn) uses the mean squared error (MSE) as default.

The regression model can deal with non-linearity well and does not require much computational power. Nonetheless, over-fitting (when the training gives the best performance, but the testing gives the worst), which could happen because the tree tries to split the training data into the best groups as possible, should be avoided. With a strict constraint, that if a new testing data is out of the training range, it will be biased.

2) Ridge cross-validation (RidgeCV): The ridge coefficients minimize the cost function,

$$\underset{w}{min} \|Xw - y\|_2^2 + \alpha \|w\|_2^2$$

that is a penalized residual sum of squares. The α is the difference between the cost function of Ridge regression and ordinary Linear Regression. Ridge cross-validation works by tuning the α, which should not be too small (to not over-fit) or too high (to prevent under-fitting). The cross-validation, which randomly separates the training data into sub-training and sub-testing data, keeps the trained model unbiased. The α coefficient is chosen based on the best cross-validation it can contribute. While RidgeCV can deal with over-fitting well if α is properly chosen, it is a linear regression algorithm in general. At the same time, our slip-angle estimator needs to handle non-linear behavior.

3) Bagging: Bagging (or Bootstrap Aggregation) is one of the Ensemble methods that use more than one decision tree from several subsets of the original data-set by randomly choosing and replacing logic. The final decision tree is the averaging of several found trees. This approach reduces the variance in a single tree to reduce over-fitting.

4) The combined regression: Our learning model combines the decision tree-based CART, Ridge CV and bagging, which can exploit the simplicity of all, the non-linearity of a decision tree, anti-over-fitting of RidgeCV and variance reduction of bagging. Bagging decision tree-based CART and RidgeCV are trained separately and the final output is their average (voting regression with weights that are [0.5,0.5] (see Fig.2).

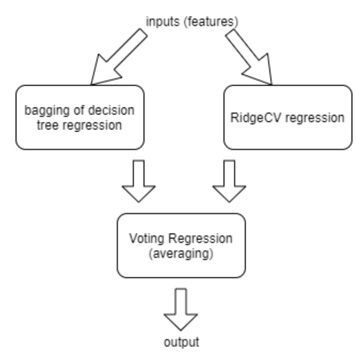

Fig. 2. The data flow of the combined regression model. The data is trained in 2 separate regression models (RigdeCV and bagging decision) then the voting regression averages the individual predictions to form a final prediction.

5) The referent slip angle: As mentioned earlier, our goal is not to be based on expensive optical sensors or a GPS to measure the slip angle as the ground truth for the learning and evaluation phases. We propose a simple approach to obtain accurate slip angle measurements based on dedicated (yet simple) infrastructure. The same method can also be applied to full-size cars. From [25] chapter 2 page 34, the lateral velocity $\dot{y}$ can be obtained by,

$$\dot{e}_1 = v_y + \int_0^t v_x \dot{e}_2 dt \tag{3}$$

or

$$v_y = \dot{e}_1 - \int_0^t v_x \dot{e}_2 dt. \tag{4}$$

Here,

- e_1 is the distance error of the vehicle, from its c.g. to the road centerline.
- e_2 is the orientation error of the vehicle with respect to the direction of the road.

In order to obtain e_1 and e_2, we have constructed a wooden wall along a straight road section and installed two Time-of-Flight (ToF) laser-ranging sensor on the car that measure the distance of the car from the wall at two different locations (see Fig.3 and Fig.4).

Fig. 3. The wooden wall built along a straight road section (at the left-hand side)

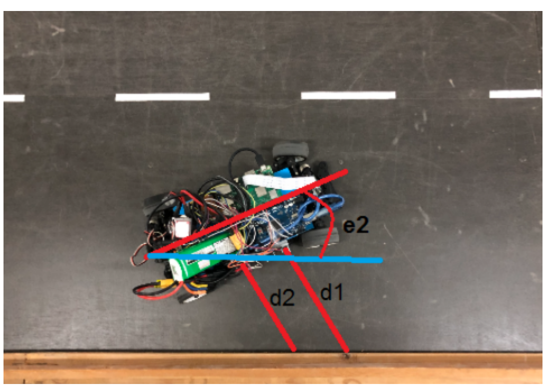

Fig. 4. The car with two ToF sensors, measuring the distances to the wall d_1 and d_2, respectively.

With this setup and the model of [25], e_1 is the average of d_1 and d_2 (i.e., the two sensors' readings, see Fig. 4) and $e_2 = \arctan((d_1 - d_2)/d_{sensors})$ with $d_{sensors}$ as the distance between the two ToF sensors (which can be measured easily). Then, rewrite (4) with discrete time integral approximation,

$$v_y = \dot{e}_1 - \sum_0^t v_x \dot{e}_2 \Delta T,$$

where ΔT is the sampling time, and use the following,

$$\dot{e}_1 = \frac{e_1 - previous(e_1)}{\Delta T}, \tag{5}$$

$$\dot{e}_2 = \frac{e_2 - previous(e_2)}{\Delta T}. \tag{6}$$

(with $previous(e_1)$ and $previous(e_2)$ as the values at the previous sampling interval) to obtain practical estimation of the errors rate.

Now, with the known longitudinal velocity v_x (taken from the driving motor encoder) and the estimation of v_y, the vehicle side-slip angle is calculated directly by (1). Zigzag driving and slow longitudinal speed allow driving with relatively high slip angles, which excites the car's nonlinear dynamics.

6) Features and data: From the parameters of the linear model in [33], the needed features (i.e, estimator inputs) are, longitudinal velocity, steering wheel angle, yaw rate, lateral acceleration, previous longitudinal velocity, previous steering wheel angle, and previous yaw rate. The yaw rate r and the lateral acceleration are measured by an onboard IMU (an UM7 Orientation Sensor). The longitudinal velocity is calculated from quadrature encoder measurements of the driving motor rotation, and the steering wheel angle is from the Logitech SDK. Because of the limited length of the straight road section (620 cm), the zigzag driving at this section was repeated, to obtain a sufficient set of samples.

The general structure of the algorithm that runs at the driving station (on a PC: Intel Core i7-8700 CPU, 3.20GHz, 16GB RAM) is described in Fig.5.

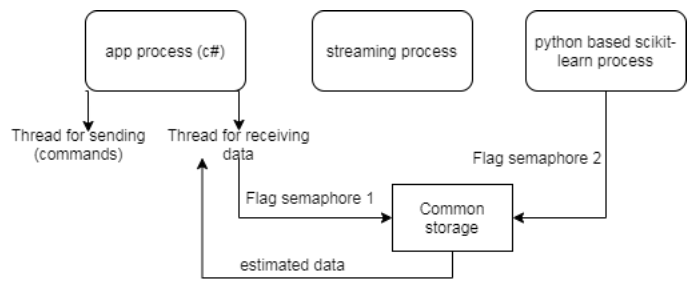

Fig. 5. Program running at the driving station. Two semaphore systems are embedded to synchronize the incoming data from the main process and a python based regression process, while the video streaming works as an independent process

IV. Results and Discussion

In the following, the testbed is shown first; then, the results of the slip angle estimation, which are also an input of the torque feedback algorithm, are discussed. Eventually, the torque feedback mechanism is evaluated.

A. Testbed

Fig.6 demonstrates the onboard camera view that the driver sees at the driving station. This photo was taken while performing the zigzag driving, and the wooden wall can be seen at the right-hand side.

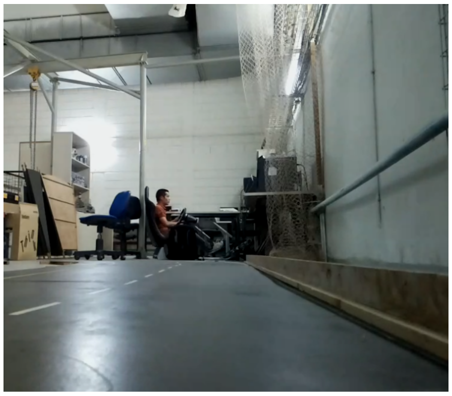

Fig. 6. The onboard camera view

A demonstration of zigzag driving, seen from the driver camera's view, is found in demo_link (https://www.youtube.com/watch?v=jOtccs8pgaY).

B. Side-slip angle estimation

We have conducted 90 rounds of zigzag drivings along a road section of 3940 samples for the training stage. Four more zigzag drives (185 samples) have been done to collect data for the testing stage. As the data we collected is in Radian with small values, RMSE may not be sufficient (since its values suppose to be small also), we therefore calculated R^2 as the percentage of accuracy, which is a relative measure of RMSE that depends on the specific situation.

TABLE 1 shows the performances of different models and how they fit the theoretical prediction. The decision tree alone results with over-fitting (perfect training but worst testing), bagging improves the variance and consequently reduces the over-fitting, linear-based RidgeCV cannot represent the non-linear characteristics of high slip angles and lastly, the combined regression solution provides the best testing outcome. The over-fitting of a single decision trees may also be observed here. The others methods result with close RMSE but with a significantly different R^2, i.e., they have the same small average error but different percentages of accuracy.

We have compared our result with Abdulrahim [23] since they have shown estimations of the same scale of slip angles using a neural-network based estimator. However, they considered neither R^2 nor RMSE but the mean of error instead. We, therefore, may intuitively conclude from the graphs that they have a better RMSE (Fig.7). Nevertheless, our obtained model can still capture the curve trend well (e.g., the points in time when the slip angle changes direction). The maximum error is 0.42 rad or about 24 deg, which is close to [23]. Fig. 7 also indicates some instability of our hardware configuration that results in different outcomes between the tests. This issue has influenced the amount of data collected for training.

In order to evaluate how good the regression model is, we implemented a Kalman filter based on the linear observer in Aoki, et.al [33], but without the calculation of a robust gain matrix (due to unknown physical parameters in our system). The necessary parameters were found by randomly picking a bunch of samples and solving the linear equations. As presented in Fig. 8, the estimator can recover the behavior trend of the reference signal but with a phase lag. The regression model suggested in this paper has shown better performance for the same conditions.

C. Torque feedback generation

Balachandran, et.al [1] suggests evaluating the torque feedback from knowledge of the effective torque stiffness, the steering sensitivity and the signal of the steering-wheel. Here, since the velocity did not change much during the experiments, only the effective torque stiffness is taken into account.

As Fig. 9 presents, we can find the effective torque stiffness region (if the steering-wheel angle goes out of the region, the torque is steady or even dropping down). As mentioned by [1] (and demonstrated in the figure) the relationship out of effective region may not be monotonic, due to the low fidelity of the spring model mentioned in [1]. Fig. 9 also shows asymmetric behavior, representing the mechanical asymmetry of the testbed steering system. An empirical assessment of the steering torque is not found here, nor in [1], and can be a future research direction.

V. Conclusion and future works

In this study, we presented an experimental setup based on a small-scale vehicle and a road. The testbed allows an affordable, practical, and safe environment for research in automotive topics, such as vehicle dynamic and driver behavior in standard or mixed (automated with manually driven vehicles) traffic scenarios. The testbed incorporates a remote driving station with low latency video streaming to provide the remote driver with a view captured from the car's windshield. To allow a realistic driving feel and because of the physical decoupling between the steering wheel and the actual steering system in the car, torque feedback was added artificially by a torque feedback motor. Calculating the needed torque requires knowledge of the vehicle's side-slip angle, and for this purpose, we have proposed a data-driven estimator. Various machine learning techniques were combined, and dedicated road infrastructure was used to learn the estimator model. Future research will include the subjective experience of participants (i.e., drivers) and comparison with standard, completely software-based, driving simulators.

VI. Acknowledgments

This research was supported in part by the Helmsley Charitable Trust through the Agricultural, Biological and Cognitive Robotics Initiative at Ben-Gurion University of the Negev.

References

[1] A. Balachandran, J. C. Gerdes, "Designing steering feel for steer-by-wire vehicles using objective measures," IEEE/ASME Trans. Mechatronics 20, 373–383 (2015).

[2] A. Balachandran, "Applications of Force Feedback Steering for Steer-by-wire Vehicles with Active Steering," P.hd Dissertation, Stanford University, 2015, Retrieved from: https://ddl.stanford.edu/sites/g/files/sbiybj9456/f/publications/Thesis_Balachrandran.pdf

TABLE I
PERFORMANCES OF DIFFERENT REGRESSION MODELS

	decision tree	bagging of decision trees	RidgeCv	combination of bagging of decision trees and RidgeCV
Accuracy (from R^2)(%): training	100	92.7	55.4	79.5
Accuracy (from R^2)(%): testing	44.4	74.4	73.3	76.6
RMSE: training	0	0.09	0.22	0.15
RMSE: testing	0.26	0.18	0.18	0.17

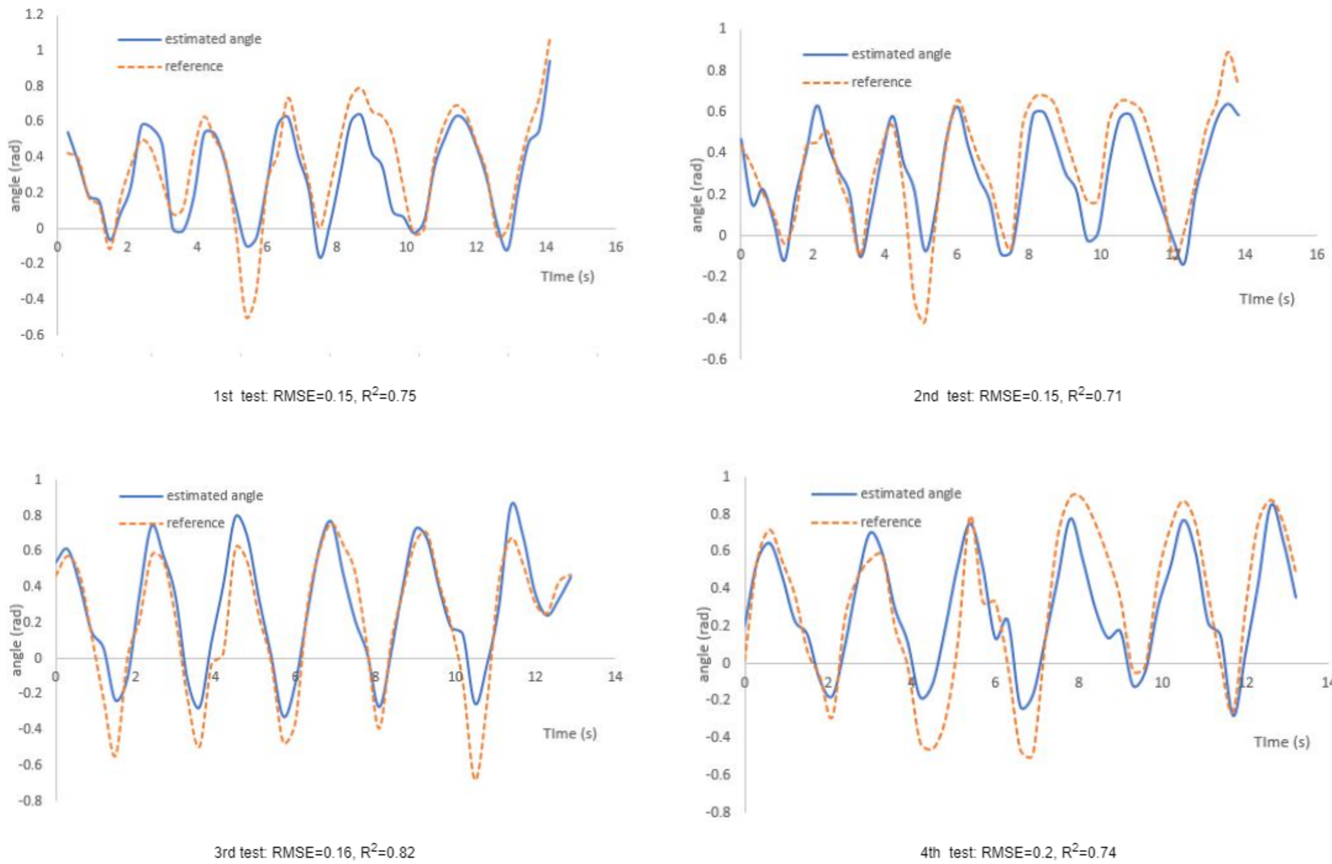

Fig. 7. Testings of the combined model (4 rounds of zigzag drive)

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," In Conference on Robot Learning (CoRL), 2017.

[4] A. Kemeny and F. Panerai, "Evaluating perception in driving simulation experiments," Trends Cogn. Sci., vol. 7, no. 1, Jan. 2003, pp. 31-37

[5] A. Liu and S. Chang, "Force feedback in a stationary driving simulator," in Proc. IEEE Int. Conf. Syst., Man Cybern., 1995, vol. 2, pp. 1711–1716.

[6] A. Nishio, K. Tozu, H. Yamaguchi, K. Asano, and Y. Amano, "Development of vehicle stability control system based on vehicle sideslip angle estimation," SAE International, Warrendale, PA, USA, SAE Tech. Paper Series, 2001

[7] B. A. C. Forsyth, and K. E. Maclean, "Predictive haptic guidance: Intelligent user assistance for the control of dynamic tasks," IEEE Trans. Vis. Comput. Graph., vol. 12, no. 1, Jan./Feb. 2006, pp. 103–113

[8] B. Vedder, J. Vinter, and M. Jonsson, "A Low-Cost Model Vehicle Testbed with Accurate Positioning for Autonomous Driving," Journal of Robotics, vol. 2018, Nov. 2018, pp. 1–10

[9] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and Regression Trees. CRC Press; 1984.

[10] C.E.Beal and S.Brennan, "Modeling and friction estimation for automotive steering torque at very low speeds," Int. Journal of Vehicle Mechanics and Mobility, Vol. 59, Issue.3, 2021, pp. 458-484

[11] C. Lv, J. Zhang, and Y. Li, "Extended-Kalman-filter-based regenerative and friction blended braking control for electric vehicle equipped with axle motor considering damping and elastic properties of electric powertrain," Vehicle Syst. Dyn., vol. 52, no. 11, 2014, pp. 1372–1388

[12] D. Fenyes, B. N´emeth, M. Asszonyi, and P. Gaspar, "Side-slip angle estimation of autonomous road vehicles based on big data analysis," in 26th Mediterranean Conference on Control and Automation, pp. 849–85, June 2018

[13] E.Vasconcelos Filho, N.Guedes, B.Vieira, M.Mestre, R.Severino, B.Gonçalves, A.Koubaa, and E.Tovar, "Towards a Cooperative Robotic Platooning Testbed," 2020 IEEE Int. Conference on Autonomous Robot Systems and Competitions (ICARSC), April.2020, pp. 332 - 337

[14] G. Phanomchoeng, R. Rajamani, and D. Piyabongkarn, "Nonlinear observer for bounded Jacobian systems, with applications to automotive slip angle estimation," IEEE Trans. Autom. Control, vol. 56, no. 5, pp. 1163–1170, May 2011

[15] H. Fang, L. Dou, J. Chen, R. Lenain, B. Thuilot, and P. Martinet,

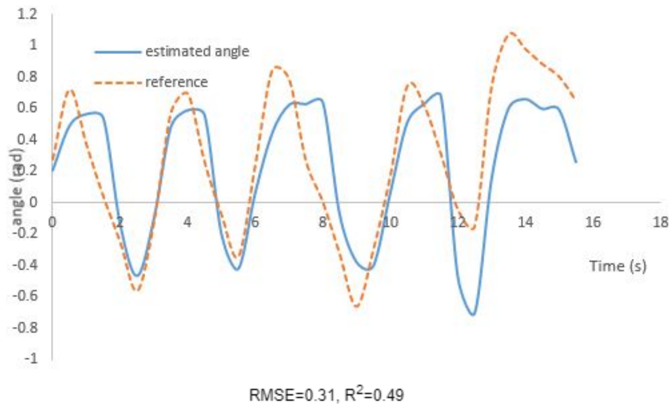

Fig. 8. Slip angle estimation using a linear observer without gain matrix

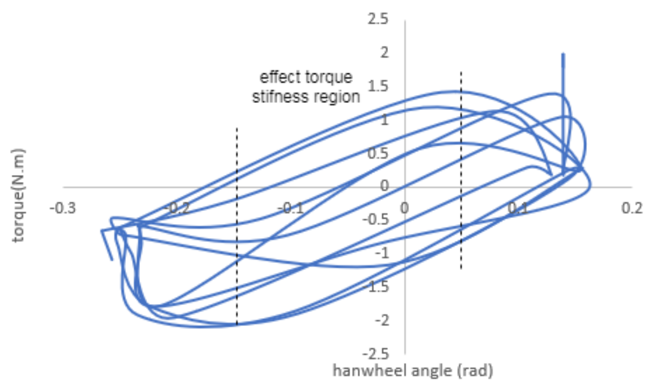

Fig. 9. Steering-wheel torque vs. the driver steering input

"Robust anti-sliding control of autonomous vehicles in presence of lateral disturbances," Control Eng. Pract., vol. 19, no. 5, 2011, pp. 468–478

[16] 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," Mech. Syst. Signal Pr., vol. 118, 2019, pp. 41–60

[17] H. Wang, H. Kong, Z. Man, M. T. Do, Z. Cao, and W. Shen, "Sliding mode control for steer-by-wire systems with AC motors in road vehicles," IEEE Trans. Ind. Electron., vol. 61, no. 3, Mar. 2014, pp. 1596–1611

[18] J. P. Jansch-Porto and G. E. Dullerud, "Decentralized control with moving-horizon linear switched systems: Synthesis and testbed implementation," in 2017 American Control Conference (ACC), May 2017, pp. 851–856, iSSN: 2378-5861

[19] J. P. Switkes, E. J. Rossetter, I. A. Coe, and J. C. Gerdes, "hand-wheel force feedback for lane keeping assistance: Combined dynamics and stability," J. Dyn. Syst., Meas., Contr., vol. 128, no. 3, 2006, pp. 532–542

[20] J. Subosits and J. C. Gerdes, "A synthetic input approach to slip angle based steering control for autonomous vehicles," in 2017 American Control Conference (ACC). IEEE, 2017 (in press)

[21] K. H. Guo and L. Ren, 1999, "A unified semi-empirical tire model with higher accuracy and less parameters," SAE Technical Paper 1999-01-0785, Society of Automotive Engineers, New York, pp. 37–44.

[22] K. Shabalina, A. Sagitov, K. Su, K.-H. Hsia, and E. Magid, "Avrora unior car-like robot in gazebo environment," in International Conference on Artificial Life and Robotics (ICAROB), 2019, pp. 116–119

[23] M. Abdulrahim, "On the dynamics of automobile drifting," SAE International, Warrendale, PA, USA, SAE Tech. Paper Series, 2006.

[24] R. Liu, D. Kwak, S. Devarakonda, K. Bekris, and L. Iftode, "Investigating remote driving over the LTE network," in Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, AutomotiveUI '17, (New York, NY, USA), ACM, 2017, pp. 264–269

[25] R. Rajamani, "Vehicle Dynamics and Control," New York, NY, USA: Springer Science Business Media, 2011. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2F978-1-4614-1433-9.pdf

[26] S. A. Arogeti and N. Berman, "Path following of Autonomous Ground Vehicles in the presence of sliding effects," IEEE Trans. Veh. Technol., vol. 61, no. 4, May 2012, pp. 1481–1492

[27] S. Cheng, L. Li, and J. Chen, "Fusion algorithm design based on adaptive SCKF and integral correction for side-slip angle observation," IEEE Trans. Ind. Electron., vol. 65, no. 7, Jul. 2018, pp. 5754–5763

[28] S. M. Taheri, K. Matsushita, M. Sasaki, "Virtual Reality Driving Simulation for Measuring Driver Behavior and Characteristics," J. Transp. Technol. 2017, 7, 123.

[29] T. Graeber, S. Lupberger, M. Unterreiner and D. Schramm, "A Hybrid Approach to Side-Slip Angle Estimation with Recurrent Neural Networks and Kinematic Vehicle Models," IEEE Transactions on Intelligent Vehicles, vol. 4, no. 1, 2019, pp. 39-47

[30] V. Punzo and B. Ciuffo, "Integration of driving and traffic simulation: Issues and first solutions," IEEE Trans. Intell. Transp. Syst., vol. 12, no. 2, Jun. 2011, pp. 354-363.

[31] W.-Y. Loh, Classification and regression trees, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 1 (1) (2011) 14–23.

[32] Xiong L, Xia X, Lu Y, Liu W, Gao L, Song S, and Yu Z, "Imu based automated vehicle body sideslip angle and attitude estimation aided by GNSS using parallel adaptive Kalman filters," IEEE Transactions on Vehicular Technology, Oct. 2020, pp.1–10

[33] Y. Aoki, T. Inoue, and Y. Hori, "Robust design of gain matrix of body slip angle observer for electric vehicles and its experimental demonstration," in Proc. 8th IEEE Int. Workshop Adv. Motion Control, Mar. 2004, pp. 41–52.