This is a preprint version.The final authenticated publication is available at https://doi.org/10.1109/ARSO64737.2025.11124958

Performance Validation of PT-DCD Using a Path Tracking Algorithm: A Predictor for Teleoperation under Dynamic Communication Delays*

Hwanhee Kang[1], Hyunrok Cha[2], Myeonghwan Hwang[2], Seungha Yoon[2], and Eugene Kim[*]

***Abstract*— Teleoperation is a technology that enables remote control of robots, rovers, and vehicles, and it is widely used in various fields such as surgical robotics, lunar exploration, and unmanned shared vehicle relocation. However, teleoperation systems rely on wireless communication to exchange control commands and sensor data, which introduces communication delays that can degrade control stability. While various approaches have been studied to mitigate the effects of such delays, they continue to pose a major challenge in teleoperation systems. In previous research, we proposed PT-DCD (Predictor for Teleoperation under Dynamic Communication Delay), a data-driven, model-free predictor based on a deep learning network, specifically the Long Short-Term Memory (LSTM). PT-DCD showed the potential to mitigate the effects of communication delay by predicting real-time control commands. Specifically, when the PT-DCD was validated under varying outlier ratios, its delay-reduction performance was consistently observed across all environments. Therefore, in this paper, we evaluate whether PT-DCD can perform effectively in environments it has not been trained on, assessing its generalization capability. Performance is measured using three metrics: average trajectory error, goal point error, and acceptable error ratio. Consequently, experimental results show that PT-DCD effectively reduces the impact of communication delays and improves teleoperation stability.**

I. INTRODUCTION

Teleoperation was initially developed using mechanical robotic arms and is now utilized in various fields, including robotic surgery, space teleoperation, Unmanned Ground Vehicles (UGVs), and remotely operated vehicles. In robotic surgery, haptic technology has been actively studied to enhance the operator's response speed and mitigate the effects of communication delay [1]. Furthermore, studies have also been reported incorporating haptic technology alongside deep learning-based simulation [2] and machine learning techniques [3]. Space teleoperation involves remotely controlling robots or machines in outer space and is applied in space exploration, space station maintenance, and remote medical operations. Recent studies have proposed methods to reduce communication delays using physics-based deep learning networks to improve the maneuverability of lunar exploration rovers [4]. Additionally, predictive force feedback utilizing Smith Predictive Control [5] and predictive algorithms combining multiple deep learning networks [6] have been explored. UGVs are employed for rescue missions in disaster sites and military operations, leading to extensive research in this area. For instance, studies have investigated teleoperation techniques using visual feedback [7], assistance through path prediction based on 3D point clouds [8], and prediction methods using mathematical modeling [9]. For remotely operated vehicles, a study was conducted to compensate for communication delays using Model Predictive Control (MPC), with simulation experiments demonstrating its effectiveness. Moreover, research utilizing digital twins based on ego-vehicle tracking algorithms has been introduced [10]. Other approaches include adaptive control methods using artificial neural networks [11], [12], reflecting ongoing efforts to address communication delays in teleoperation systems. Fig. 1 shows the overall procedure of our study and the current phase. In our previous study, we developed PT-DCD (Predictor for Teleoperation under Dynamic Communication Delay), an LSTM-based prediction model to reduce communication delays in teleoperation systems[13]. We employed an LSTM network because this data-driven approach can learn system dynamics without elaborate modeling and is more effective than classical predictors such as MPC and the Kalman filter in handling nonlinear systems. PT-DCD predicts control commands in teleoperation systems without prior modeling, reducing communication delays. Specifically, PT-DCD was designed to predict steering commands in real time, enabling more immediate control. Experimental results demonstrated that PT-DCD reduced delay by over 30 %. In this paper, we evaluate the performance of PT-DCD by investigating whether it effectively reduces delays for unseen steering command patterns. To achieve this, we implemented a vehicle path-following algorithm in a simulation environment with applied communication delays and analyzed the performance of PT-DCD. Through experiments, we aim to verify the stability and effectiveness of PT-DCD across various steering command scenarios.

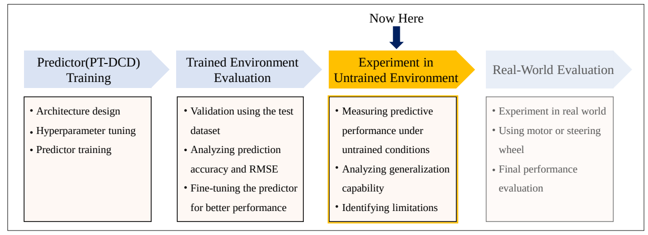


Fig. 1: Description of the overall predictor development procedure for teleoperation.

*This work was supported by "Development of Core Technologies for a Working Partner Robot in the Manufacturing Field" (KITECH EO-250005)

[1]Hwanhee Kang is with Robot Engineering, Korea National University of Science and Technology, Daejeon 34113, South Korea hwan@kitech.re.kr

[2]All the authors are with the Purpose-Based Mobility group, Seonam Division, Korea Institute of Industrial Technology, Gwangju 61012, South Korea egkim@kitech.re.kr

This is a preprint version.The final authenticated publication is available at https://doi.org/10.1109/ARSO64737.2025.11124958

II. METHODOLOGY

PT-DCD is a data-driven, model-free predictor using LSTM, which is highly effective for time series forecasting. Conceptually, it is designed to receive delayed steering commands and predict real-time steering commands. The communication delay model was implemented according to the method proposed in [14]. This approach revealed that in real-world teleoperation, communication delays follow a mixture of two normal distributions: $\mathcal{N}(0.03, 0.01^2)$ and $\mathcal{N}(0.113, 0.05^2)$, with a mixing ratio of 9:1. Among them, the distribution with a mean of 0.113 represents occasional outliers. The proportion of these outliers was defined as the contamination ratio (ρ), which was 0.1. Using this delay model, random delays were generated from the mixed normal distribution with a contamination ratio of 0.1 and applied to the steering commands. PT-DCD consists of two LSTM layers. The input $\tilde{\boldsymbol{X}}_k$ and output $\hat{\boldsymbol{Y}}_k$ can be expressed as follows:

$$\tilde{\boldsymbol{X}}_k = [\tilde{\boldsymbol{x}}_k, \tilde{\boldsymbol{x}}_{k-1}, \ldots, \tilde{\boldsymbol{x}}_{k-w_\mathrm{i}+1}]^\mathrm{T} \quad (1)$$
$$\hat{\boldsymbol{Y}}_k = [\hat{\boldsymbol{x}}_k, \hat{\boldsymbol{x}}_{k-1}, \ldots, \hat{\boldsymbol{x}}_{k-w_\mathrm{o}+1}]^\mathrm{T}, \quad (2)$$

where w_i, w_o indicate predetermined input and output window sizes of the dataset. w_i, w_o were set to 4 and 1, respectively, meaning that the model predicts the current steering command based on the most recent four input data points. The input vector $\tilde{\boldsymbol{x}}_k$ for PT-DCD is defined as:

$$\tilde{\boldsymbol{x}}_k = [\tilde{\theta}_k,\, d(k),\, \tilde{T}_\mathrm{steer}(k)]^\mathrm{T}, \quad (3)$$

where $\tilde{\theta}_k$ is delayed steering command, $d(k)$ is magnitude of delay, and $\tilde{T}_\mathrm{steer}(k)$ is delayed steering torque. The non-linear prediction function g_p, parameterized by Θ, and the prediction error $\boldsymbol{\epsilon}_\mathrm{p}(k)$ are described as follows:

$$g_\mathrm{p}(\tilde{\boldsymbol{X}}_k|\Theta) = \hat{\boldsymbol{Y}}_k \quad (4)$$
$$\boldsymbol{\epsilon}_\mathrm{p}(k) := \boldsymbol{x}(k) - g_\mathrm{p}(\tilde{\boldsymbol{X}}_k)|_n, \quad (5)$$

where $g_\mathrm{p}(\tilde{\boldsymbol{X}}_k)|_n$ is the n-th element of PT-DCD output vector. The output is the real-time steering command, i.e., a delay-compensated steering command. For training, we used the commaSteeringControl dataset from Comma.ai [15]. This dataset consists of vehicle state data sampled at 0.1 s intervals. A total of 2,975 sequences, each 60 s long, were used for training, and the PT-DCD was trained for 183 epochs.

III. EXPERIMENT

In this experiment, we employed the Linear-Quadratic Regulator (LQR) speed and steering control method from PythonRobotics [16] to generate control commands. Communication delays were introduced only to the steering control commands; afterwards, PT-DCD compensated for these delays. The speed control commands, in contrast, were transmitted without any delay. LQR is widely used in autonomous driving control as it enables precise trajectory tracking and smooth, optimized steering while being relatively simple to implement in simulations. Furthermore, this method was selected as it allows for a detailed comparison between cases with a relatively small communication delay (average 0.03 s) and those where PT-DCD is applied. The experiment was conducted under two conditions:

- **Delayed**: Dynamic communication delay is applied to steering command without compensation.
- **Predicted**: Steering command is corrected using PT-DCD.

Fig. 2 illustrates the original LQR control and the altered steering patterns affected by dynamic communication delays and corrected using PT-DCD. Each condition was repeated 100 times, and the distance error was compared between the **Delayed** and **Predicted**. The distance error was calculated as the perpendicular distance between the vehicle's position and the nearest point on the reference path. It was recorded at 0.1 s intervals and calculated using the Root Mean Square Error (RMSE). The RMSE of the distance error was evaluated using the following metrics:

- Path-Averaged Distance Error
- Goal Point Error
- Acceptable Error Ratio

The following assumptions were adopted for this experiment.

- The exact communication delay magnitude can be accurately measured.
- Steering torque can be approximated by differentiating the steering command.
- Speed control and steering control communications are independent, and delay only affects the steering command transmission.

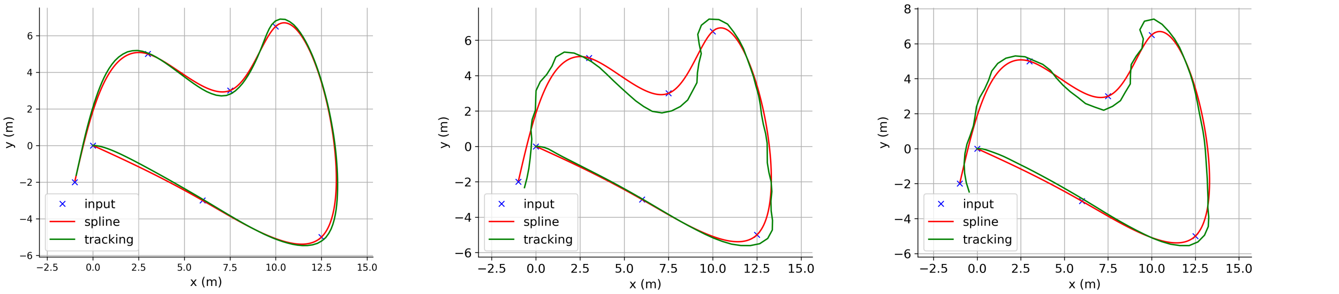
IV. RESULTS AND DISCUSSTION

This section presents the experimental results. All results are obtained by averaging the values over 100 repeated experiments. The factors that vary in these repeated experiments are the magnitude of dynamic communication delay and the corresponding predicted values of PT-DCD. To evaluate the robustness of PT-DCD under various communication delay conditions, experiments were conducted by varying the contamination ratio (ρ) of the communication delay. Specifically, $\rho = 0.1$ denotes the trained scenario, while $\rho = 0$, 0.3, and 0.5 denote untrained scenarios.

A. Evaluation of Path-Averaged Distance Error

Table I shows the computed distance error between the given reference path and the actual driving trajectory of the vehicle. The perpendicular distance between the vehicle and the reference path was calculated each time the LQR controller generated control commands at 0.1 s intervals. RMSE was then computed from the collected errors obtained over 100 repeated experiments. With $\rho = 0.1$, the RMSE of the **Delayed** case was 0.449 m, whereas that of the **Predicted** case was 0.405 m, indicating that PT-DCD reduced RMSE by approximately 9.78 %. In addition, the results for $\rho = 0, 0.3, 0.5$ indicate that applying PT-DCD led to a reduction in RMSE across all tested conditions. It was initially expected that as the contamination ratio increased, communication delay outliers would occur more

This is a preprint version.The final authenticated publication is available at https://doi.org/10.1109/ARSO64737.2025.11124958



(a) LQR steering control (b) **Delayed** (c) **Predicted**

Fig. 2: LQR steering control simulation: (a) Original LQR steering control, (b) **Delayed** case with applied dynamic communication delays, and (c) **Predicted** case where communication delays are compensated using PT-DCD. The red line represents the given path, the green line represents the actual path taken by the vehicle, and the 'x' markers indicate target points during driving.

frequently, which would in turn, reduce the effectiveness of RMSE improvement. However, the experimental results did not show a clear correlation between changes in ρ and the degree of RMSE reduction.

TABLE I: RMSE Comparison of Path-Averaged Distance Error

ρ	0	0.1	0.3	0.5
Delayed (m)	0.377	0.449	0.523	0.634
Predicted (m)	0.365	0.405	0.515	0.579
RMSE reduction (%)	3.226	9.778	1.475	8.672

B. Evaluation of Goal Point Error

Table II presents the final position error of the vehicle at the goal point upon completion of the driving trajectory. When $\rho = 0.1$, the RMSE values for the **Delayed** and **Predicted** cases were 0.485 m and 0.420 m, respectively, demonstrating a 13.45 % reduction in error when PT-DCD was applied. However, the effect of varying ρ was inconsistent across the experiments: while RMSE decreased by 10.38 % at $\rho = 0.3$, it increased at both $\rho = 0$ and $\rho = 0.5$. This phenomenon can be interpreted by considering that the goal point is merely a randomly assigned position on the path and does not possess any distinct characteristics from a control algorithm's perspective. As a result, while PT-DCD reduces the overall path-following error, this effect may not necessarily be reflected in the final position error at the goal point. In addition to the randomness of the goal point, performance degradation can also arise from a mismatch between the training and test delay distributions. Under $\rho = 0$ (no delay), PT-DCD trained on a $\rho = 0.1$ may suffer from prediction mismatch; whereas under $\rho = 0.5$ (high outlier frequency), extreme delays can trigger overcompensation, resulting in increased error at the goal point.

TABLE II: RMSE Comparison of Goal Point Error

ρ	0	0.1	0.3	0.5
Delayed (m)	0.364	0.485	0.699	0.668
Predicted (m)	0.368	0.420	0.627	0.737
RMSE reduction (%)	-1.105	13.446	10.376	-10.339

C. Evaluation of Acceptable Error Ratio

Table III compares the proportion of trajectory errors that fall within a tolerable lateral deviation range (± 0.5 m) between the **Delayed** and **Predicted** cases. While numerous studies have explored control methods that minimize lateral error in path following, a universally defined criterion for acceptable lateral error in stable driving has not been established. Instead, Autonomous vehicle steering assist systems—namely, Lane Departure Warning (LDW), Lane Keeping Assist (LKA), and Lane Following Assist (LFA)—all operate based on lane markings [17], [18], [19]. Generally, when a vehicle remains within its lane, it is considered that proper path tracking is achieved. That is, as long as the vehicle does not cross the lane boundaries, it meets at least the minimum requirements for driving stability. On typical roads, the standard lane width is approximately 3 m, while most passenger vehicles have a width of around 2 m. Therefore, a vehicle can maintain its trajectory within a 1 m range from the lane center, and a lateral deviation of ± 0.5 m can be defined as an acceptable range. Based on this criterion, we calculated the proportion of trajectory errors that fell within this range. The acceptable error ratio for the **Delayed** case was 81.71 %, while for the **Predicted** case using PT-DCD, it increased to 85.19 %, representing an improvement of 3.48 %p. Additionally, experiments conducted under different values of ρ confirmed that, in all tested conditions, the acceptable error ratio of **Predicted** exceeded that of **Delayed**, as shown in Fig. 3. Notably, the most significant improvement was observed in the training condition of PT-DCD ($\rho = 0.1$). These results suggest that applying PT-DCD increases the duration for which the vehicle remains within

This is a preprint version.The final authenticated publication is available at https://doi.org/10.1109/ARSO64737.2025.11124958

the lane, ultimately enhancing overall driving stability.

TABLE III: Comparison of Acceptable Error Ratio

ρ	0	0.1	0.3	0.5
Delayed (m)	86.635	81.713	77.931	72.183
Predicted (m)	87.406	85.194	79.333	74.492
Increment (%)	0.771	3.480	1.402	2.309

V. Conclusion

Teleoperation is utilized across various fields, and its potential applications could expand significantly if the issue of communication delay is effectively addressed. In this study, experimental results confirmed that PT-DCD can effectively mitigate the impact of dynamic communication delay in teleoperation systems. Performance was evaluated using path-averaged distance error, goal point error, and acceptable error ratio, comparing the results between the **Delayed** and **Predicted** cases. The findings showed that PT-DCD had the most significant impact in reducing communication delay effects in the training condition ($\rho = 0.1$). Furthermore, even in untrained communication delay conditions ($\rho = 0, 0.3, 0.5$), PT-DCD improved control performance in most cases, demonstrating its generalizability. This suggests that further optimization of PT-DCD based on the specific characteristics of communication delay in a given environment could yield even greater improvements. Therefore, applying PT-DCD in real-world teleoperation scenarios with dynamic communication delay is expected to enhance driving stability. Future research will focus on designing and training an additional network for speed prediction, enabling the simultaneous prediction of both steering and speed to further improve the robustness of teleoperationf systems.

Acknowledgment

This work was supported by the "Development of Core Technologies for a Working Partner Robot in the Manufacturing Field" (KITECH EO-250005).

References

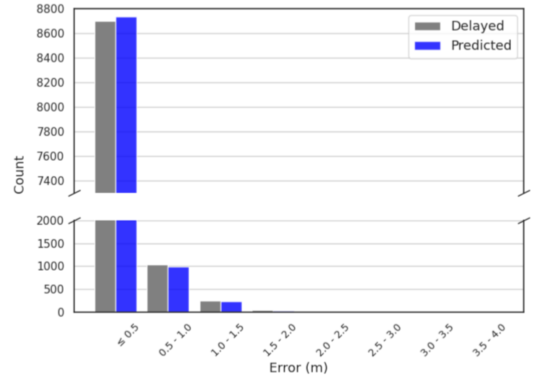
[1] R. V. Patel, S. F. Atashzar, and M. Tavakoli, "Haptic feedback and force-based teleoperation in surgical robotics," *Proceedings of the IEEE*, vol. 110, no. 7, pp. 1012–1027, 2022.

[2] G. Gonzalez, M. Agarwal, M. V. Balakuntala, M. Masudur Rahman, U. Kaur, R. M. Voyles, V. Aggarwal, Y. Xue, and J. Wachs, "Deserts: Delay-tolerant semi-autonomous robot teleoperation for surgery," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12693–12700, 2021.

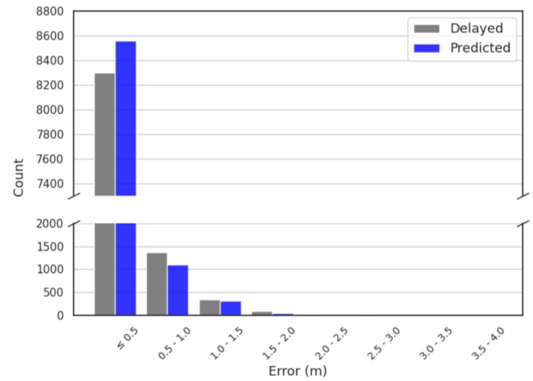
[3] F. Boabang, A. Ebrahimzadeh, R. H. Glitho, H. Elbiaze, M. Maier, and F. Belqasmi, "A machine learning framework for handling delayed/lost packets in tactile internet remote robotic surgery," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4829–4845, 2021.

[4] A. Abubakar, Y. Zweiri, A. Haddad, M. Yakubu, R. Alhammadi, and L. Seneviratne, "Physics-informed lstm-based delay compensation framework for teleoperated ugvs," *arXiv preprint arXiv:2402.16587*, 2024.

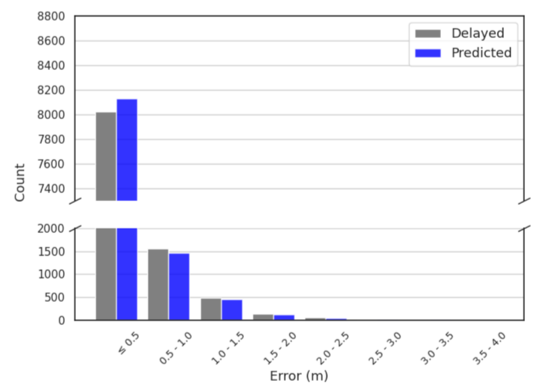
[5] H. Chen and Z. Liu, "Time-delay prediction-based smith predictive control for space teleoperation," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 4, pp. 872–879, 2021.



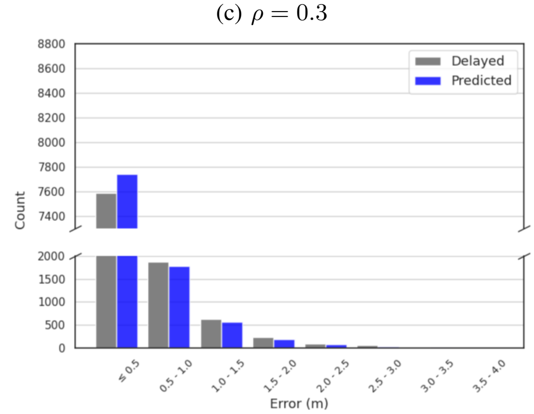
(a) $\rho = 0$



(b) $\rho = 0.1$



(c) $\rho = 0.3$



(d) $\rho = 0.5$

Fig. 3: Histograms comparing the acceptable error ratio for the **Delayed** and **Predicted** as a function of ρ. The first bin ($\leq 0.5\,\mathrm{m}$) indicates the acceptable error range.

[6] K. Wu, H. Chen, L. Li, Z. Liu, and H. Chang, "Position prediction for space teleoperation with sao-cnn-bigru-attention algorithm," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11674–11681, 2024.

[7] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, "High latency unmanned ground vehicle teleoperation enhancement by presentation of estimated future through video transformation," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 2, p. 48, 2022.

[8] Q. Zhang, Z. Xu, Y. Wang, L. Yang, X. Song, and Z. Huang, "Predicted trajectory guidance control framework of teleoperated ground vehicles compensating for delays," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 9, pp. 11264–11274, 2023.

[9] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, "Evaluation of a predictor-based framework in high-speed teleoperated military ugvs," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 6, pp. 561–572, 2020.

[10] P. Kremer, N. Nourani-Vatani, and S. Park, "Digital twin enabled teleoperated driving under network delay using ego vehicle tracking," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1943–1948, IEEE, 2023.

[11] S. Zhang, S. Yuan, X. Yu, L. Kong, Q. Li, and G. Li, "Adaptive neural network fixed-time control design for bilateral teleoperation with time delay," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9756–9769, 2022.

[12] M. Nikpour, B. Yazdankhoo, B. Beigzadeh, and A. Meghdari, "Adaptive online prediction of operator position in teleoperation with unknown time-varying delay: simulation and experiments," *Neural Computing and Applications*, vol. 33, pp. 7575–7592, 2021.

[13] H. Kang, H. Cha, M. Hwang, S. Yoon, and E. Kim, "Digital twin simulation of teleoperation: Communication delay compensation using lstm network," in *2024 24th International Conference on Control, Automation and Systems (ICCAS)*, pp. 383–388, 2024.

[14] E. Kim, H. Cha, T. Talluri, C. Jeong, H. Kim, S. Yoon, and M. Hwang, "Delay compensation and outlier removal in command signal for teleoperated vehicle using asynchronous filter," *IEEE Access*, 2024.

[15] Comma.ai, "commasteeringcontrol." https://huggingface.co/datasets/commaai/commaSteeringControl, n.d.

[16] A. Sakai, "Pythonrobotics." https://github.com/AtsushiSakai/PythonRobotics.git.

[17] "Intelligent transport systems – lane departure warning systems – performance requirements and test procedures," 2017.

[18] "Intelligent transport systems – lane keeping assistance systems (lkas) – performance requirements and test procedures," 2014.

[19] "Road vehicles – test method to evaluate the performance of lane-keeping assistance systems," 2021.