Amirkabir Journal of Mechanical Engineering

Amirkabir J. Mech. Eng., 53(Special Issue 4) (2021) 577-580
DOI: 10.22060/mej.2020.18008.6708

A comparison study of deep neural controllers and classic controllers in self-driving car application

A. Mohammadi[1], K. Jamshidi[1*] , H. Shahbazi[2], M. Rezaei[1]

[1] Department of Computer Engineering, University of Isfahan, Isfahan, Iran
[2] Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran

ABSTRACT: In this paper deep neural controller is evaluated in self-driving car application which is one of the most important and critical among human-in-the-loop cyber physical systems. To this aim, the modern controller is compared with two classic controllers, i.e. proportional–integral–derivative and model predictive control for both quantitative and qualitative parameters. The parameters reflect three main challenges: (i) design-time challenges like dependency to the model and design parameters, (ii) implementation challenges including ease of implementation and computation workload, and (iii) run-time challenges and parameters covering performance in terms of speed, accuracy, control cost and effort, kinematic energy and vehicle depreciation. The main objective of our work is to present comparison and concrete metrics for designers to compare modern and traditional controllers. A framework for design, implementation and evaluation is presented. An end-to-end controller, constituting six convolution layers and four fully connected layers, is evaluated as the modern controller. The controller learns human driving behaviors and is used to drive the vehicle autonomously. Our results show that despite the main advantages of the controller i.e. being model free and also trainable, in terms of important metrics, this controller exhibits acceptable performance in comparison with proportional–integral–derivative and model predictive controllers.

Review History:

Received: Feb. 27, 2020
Revised: Aug. 01, 2020
Accepted: Oct. 31, 2020
Available Online: Nov. 23, 2020

Keywords:

Cyber physical systems
Human-in-the-loop system
Deep learning
End-to-end control
Self-driving car

1- Introduction

Nowadays, the presence of cyber physical systems in modern devices especially safety-critical applications, draws great attention from researchers. There are broad range of domains to consider, including medical and health care/assistance devices, transportation, robotics, intelligent houses, communication systems, game and entertainment and autonomous driving. Due to dynamic behavior of both computation and communication counterpart of a cyber-physical system, control of such systems can be increasingly complicated [1].

Human-In-The-Loop control systems (HITL) can be defined as cyber physical systems with dynamic and unpredictable intervention of human in control loop [2]. The HITL control concept is known as a solution in challenging tasks such as autonomous vehicles. Especially in autonomous cars, efficient design and implementation of such controllers is the focus of many studies. There are a lot contemporary research devoted to solving autonomous driving, intelligent path planning, collision avoidance and also driver mental and physical health monitoring problems. Two different approaches have been considered to attach these problems, the classic and the modern control solutions (mainly machine learning-based approach). Classic control solutions may not be applicable in complicated and dynamic systems (and their surroundings) due to the presence of hard-to-model dynamics [1, 3].

End-To-End (ETE) training is one of the well-known machine learning methods widely used in deep neural networks, as the controller. In this research, we employ an ETE controller for autonomous cars in both simulation and small size real plant. To have a comprehensive and deep evaluation, we further compare the ETE controller with PID and MPC controllers, which are top popular classic controllers. To the best of our knowledge, this paper is the first experimental and comprehensive study and comparison of the ETE with the mentioned classic controllers.

The rest of this paper is organized as follows. First, our methodology is explained. In section3 our results will be presented and discussed. Finally, we conclude our work in section4.

2- Methodology

The main issue in design and implementation of ETE controller is to choose efficient and suitable architecture of a deep neural network. Our proposed method is based on PilotNet architecture [4, 5] with some minor modifications for our experimental setup adaptation. As it is depicted in Fig. 1, the architecture constitutes of 9 convolution and fully-connected layers.

The network directly maps the input image to angle of wheel as output.

Our methodology constitutes training the deep network that can be used as controller in both simulation and real small-scale autonomous car. The overview of our method is

*Corresponding author's email: jamshidi@eng.ui.ac.ir

Copyrights for this article are retained by the author(s) with publishing rights granted to Amirkabir University Press. The content of this article is subject to the terms and conditions of the Creative Commons Attribution 4.0 International (CC-BY-NC 4.0) License. For more information, please visit https://www.creativecommons.org/licenses/by-nc/4.0/legalcode.

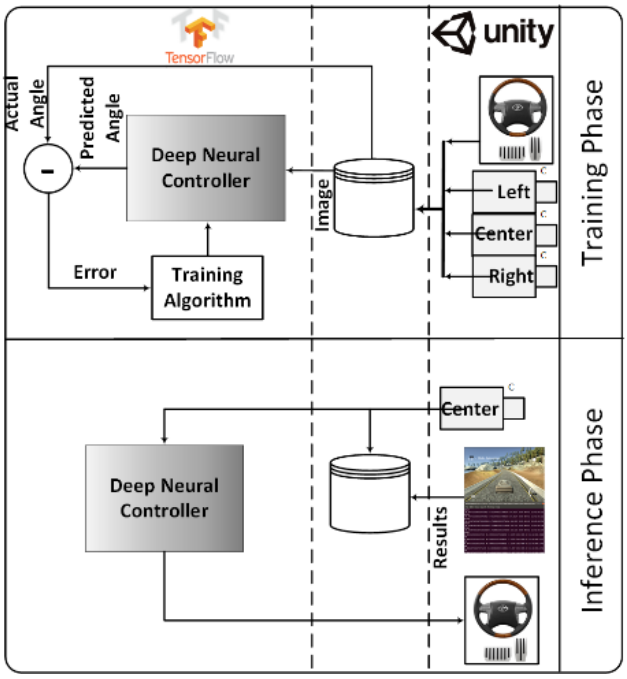


Fig. 2. Proposed method for ETE training and inference

Table 1. Training configuration

Parameter	Value
Optimization	Adam
Initial Learning Rate	1e-4
Dropout Rate	0.5
Maximum Epoch	2000
Batch Size	100

shown in Fig. 2. Obviously, the method can be divided into two main phases, namely training and inference. In addition to the two main phases, the comparison experimental setup is also the capability of our method.

During the training phase, the driving data including images from the installed camera, wheel angle, accelerate and brake values altogether collected and stored while a person controls the car in both simulations (in unity) and the real environment.

Both training algorithm and structure of network, implemented in Tensorflow, import the collected dataset proceeded by the training of the network which will be started in order to tune the network parameters (weights and biases). The configuration of the training phase is presented in Table 1.

For the inference phase, we have gotten two environments, the unity-based simulator and our real testbed i.e. a small-scale self-driving car, shown in Fig. 3. In this step, the trained network is loaded and the input images from both real and modeled camera applied to the network and final output will be analyzed in order to generate commands for steering angle, brake and accelerate of the real and modeled car.

In this research, we compare the ETE controller with the classic PID and MPC controllers in different aspects for evaluation. Out comparison is based on parameter dependence, statistical distribution, computation load, steering angle, speed control, brake and accelerate control, error, safety, and travelled distance. The details of experiments and results can be found in our main paper for interested readers.

3- Discussion and Results

In this section, the achieved results are explained. In terms of statistical distribution, PID controller issues steep angles and high amount in the rate of changes. This can cause car Depreciation and bad feelings to passengers. According to this parameter, MPC performs better than PID and ETE performs moderately between PID and MPC.

In terms of speed control, despite the fact that MPC shows the best control among controllers, but ETE causes less high variance moves that impose less stress and imbalance on car and its passengers. This fact is also proved when we consider the maximum acceleration in the brake as a metric. From this point of view, ETE performs better than MPC but worse than PID.

Error is another comparison factor that is equal to the mean square error between human decision and controller decision. ETE performs better in terms of acceleration changes and brakes but ranks second in terms of steering angle and speed.

Safety can be also determined as another comparison factor. We determine a controller as safe whenever it does not touch the border of the road. Based on the definition the ETE controller is safe while the rest has crossed the borderline of the road.

As different controllers show different trajectories for the same journey from start point to end, it is important to determine which controller has a shorter travel distance. Our experiments show that ETE has a shorter travel distance.

4- Conclusions

In this paper, we aim at providing a comprehensive evaluation and comparison among ETE controller and classic PID and MPC controllers for a self-driving car in both simulation and real environment. Our experimental results can be used as a guideline for researchers and designers. Although MPC generally shows better performance in terms of some parameters, but the complicated design and implementation of MPC can become problematic for real-world problems. Being drastically model and parameter-dependent as well as having computation load based on design parameters, can be an important challenge for both PID and MPC. In addition, better accuracy performance and also learning capability of ETE controllers make this approach interesting and popular for designers.

Since in this paper the base algorithm for PID, MPC and ETE have considered for experiments as future work, we are going to implement and compare optimized version of the controllers to have a more advanced comparison. Furthermore, different deep architecture e.g. deep LSTM network and also hybrid methods will be considered as future works.

References

[1] S.K. Khaitan, J.D. McCalley, Design techniques and applications of cyberphysical systems: A survey, IEEE Systems Journal, 9(2) (2014) 350-365.

[2] M. Jirgl, Z. Bradac, P. Fiedler, Human-in-the-loop issue in context of the cyber-physical systems, IFAC-PapersOnLine, 51(6) (2018) 225-230.

[3] I. Lenz, R.A. Knepper, A. Saxena, DeepMPC: Learning deep latent features for model predictive control, in: Robotics: Science and Systems, Rome, Italy, 2015.

[4] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, U. Muller, K. Zieba, Visualbackprop: visualizing cnns for autonomous driving, arXiv preprint arXiv:1611.05418, 2 (2016).

[5] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L.D. Jackel, M. Monfort, U. Muller, J. Zhang, End to end learning for self-driving cars, arXiv preprint arXiv:1604.07316, (2016).

HOW TO CITE THIS ARTICLE

A. Mohammadi, K. Jamshidi, H. Shahbazi, M. Rezaei, A comparison study of deep neural controllers and classic controllers in self-driving car application, Amirkabir J. Mech. Eng., 53(Special Issue 4) (2021) 577-580.

DOI: 10.22060/mej.2020.18008.6708

This page intentionally left blank

نشریه مهندسی مکانیک امیرکبیر

نشریه مهندسی مکانیک، دوره ۵۳، شماره ویژه ۴، سال ۱۴۰۰، صفحات ۲۴۳۹ تا ۲۴۵۸
DOI: 10.22060/mej.2020.18008.6708

طراحی و پیاده‌سازی کنترل‌کننده‌های عصبی یادگیرنده عمیق در مقایسه با کنترل‌کننده‌های کلاسیک در کاربرد ماشین خودران

عباس محمدی[۱]، کمال جمشیدی[۱*]، حامد شهبازی[۲]، مهران رضایی[۱]

۱- مهندسی کامپیوتر، دانشگاه اصفهان، اصفهان، ایران
۲- هندسی مکاترونیک، دانشگاه اصفهان، اصفهان، ایران

خلاصه: در این مقاله کنترل‌کننده‌های عصبی یادگیرنده عمیق در کاربرد ماشین خودران که یکی از معروفترین سیستم‌های انسان در حلقه است بررسی شده‌است. این کنترل‌کننده مدرن با دو کنترل‌کننده معروف کلاسیک تناسبی- انتگرال- مشتقگیر و کنترل پیشبین مدل، از نظر پارامترهای کیفی وکمی مقایسه شده‌است. این پارامترها انعکاس‌دهنده چالش‌های زمان طراحی شامل وابستگی به مدل و پارامترهای طراحی؛ چالش‌های زمان پیاده‌سازی شامل سهولت پیاده‌سازی و بار محاسباتی؛ و همچنین پارامترهای مهم در زمان اجرا در کاربرد ماشین خودران شامل دقت، سرعت، هزینه یا تلاش کنترلی، استهلاک و انرژی هستند. در روش پیشنهادی، بستری فراهم آورده شده‌است که طراحان قادر باشند کنترل‌کننده‌های مدرن و کلاسیک را با یکدیگر مقایسه نمایند. در روش پیشنهادی از کنترل‌کننده عصبی یادگیرنده عمیق به صورت انتها- به- انتها که ساختار آن متشکل از لایه‌های کانولوشن و لایه‌های تماما متصل است، به عنوان کنترل‌کننده مدرن استفاده شده‌است. این کنترل‌کننده در ابتدا از رانندگی انسان می‌آموزد سپس به عنوان کنترل‌کننده خودرو مورد استفاده قرار می‌گیرد. عدم وابستگی این کنترل‌کننده به مدل سیستم کنترل‌شونده (ماشین خودران) و محیط و همچنین قابلیت یادگیری مهمترین مزیت آن است. از نقطه نظر پارامترهای مهم در کنترل ماشین خودران در مقایسه با کنترل‌کننده‌های تناسبی- انتگرال- مشتقگیر و کنترل پیشبین مدل، کارائی قابل قبولی از خود به نمایش می‌گذارد.

تاریخچه داوری:
دریافت: ۱۳۹۸/۱۲/۰۸
بازنگری: ۱۳۹۹/۰۵/۱۱
پذیرش: ۱۳۹۹/۰۸/۱۰
ارائه آنلاین: ۱۳۹۹/۰۹/۰۳

کلمات کلیدی:
سیستم‌های فیزیکی- رایانشی
سیستم انسان در حلقه
یادگیری عمیق
کنترل انتها- به- انتها
ماشین خودران

۱- مقدمه

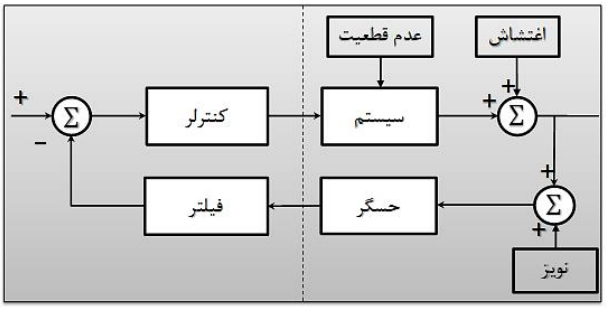
هوشمندسازی سیستم‌های فیزیکی- رایانشی[۱] در راستای برآوردن نیازمندیهای بشر از یک سو و کاربرد روزافزون این سیستم‌ها در کاربردهای بحرانی- ایمن[۲] از سوی دیگر سبب شده‌است تا این موضوع مورد توجه بسیاری از محققان قرار بگیرد. این کاربردها در حوزه‌های مختلف: پزشکی و کمک به زندگی، صنعت حمل‌ونقل، ایمنی جاده‌ها و کنترل ترافیک، سیستم‌های پیشرفته خودرو، کنترل فرآیندهای صنعتی، کنترل انرژی، محافظت از منابع محیط زیست، صنایع هوافضا، صنایع دفاعی، رباتیک، ساختمان‌های هوشمند، سیستم‌های ارتباطی و حتی صنعت بازی و سرگرمی، امروزه مطرح هستند. نیازمندیهای کنترلی سیستم‌های فیزیکی- رایانشی در اثر توسعه و پیشرفت این کاربردها، پویایی بخش محاسباتی از یک سو و بخش فیزیکی و همچنین شبکه‌ای که ارتباط این دو را برقرار می‌نماید نیز از سوی دیگر روز به روز گسترده‌تر و پیچیده‌تر می‌شود [۱].

در دسته‌ی خاصی از این سیستم‌ها، انسان به‌عنوان بخش اصلی سیستم که به‌صورت غیرقابل‌پیش‌بینی بر روی حلقه‌ی کنترلی تأثیرگذار است، در نظر گرفته می‌شود. این سیستم‌ها به سیستم‌های انسان در حلقه[۳] معروف هستند [۲]. ماشین خودران به دلیل نقش

1 Cyber Physical Systems
2 Safety Critical
3 Human in The Loop (HITL)

* نویسنده عهده‌دار مکاتبات: jamshidi@eng.ui.ac.ir

حقوق مؤلفین به نویسندگان و حقوق ناشر به انتشارات دانشگاه امیرکبیر داده شده است. این مقاله تحت لیسانس آفرینندگی مردمی (Creative Commons License) در دسترس شما قرار گرفته است. برای جزئیات این لیسانس، از آدرس https://www.creativecommons.org/licenses/by-nc/4.0/legalcode دیدن فرمائید.



شکل ۱. دیاگرام سیستم کنترلی

Fig. 1. Control system diagram

مهم خودرو در زندگی روزمره انسان‌ها یکی از مهمترین سیستم‌های فیزیکی- رایانشی انسان در حلقه است که طراحی و پیاده‌سازی کنترل‌کننده‌های کارآمد برای این سیستم‌ها بسیار مورد توجه است. مسائل مختلفی در این راستا مطرح است از جمله رانندگی خودکار، مسیریابی هوشمند، اجتناب از تصادف و همچنین بررسی رفتار و شرایط روحی و جسمی راننده که محققان روش‌های مختلفی برای حل آن‌ها ارائه نموده‌اند. دو رویکرد برای حل این مسائل مطرح است، رویکرد اول استفاده از روش‌های کلاسیک کنترلی و رویکرد دوم استفاده از روش‌های مدرن مبتنی بر یادگیری ماشین می‌باشد. استفاده از رویکردهای کنترلی کلاسیک برای حل چنین مسائلی با توجه به پویایی پیچیده و غیرخطی که مدل‌سازی ریاضی آن یا بسیار دشوار یا غیر ممکن یا با خطا روبروست، با چالش‌های جدی مواجه است [۱, ۳].

از این رو توجه طراحان را به سمت روش‌های مبتنی بر یادگیری ماشین و هوش مصنوعی هدایت نموده است. یکی از روش‌های معروف کنترل مبتنی بر یادگیری ماشین، استفاده از شبکه‌های عصبی مصنوعی عمیق با رویکرد یادگیری انتها- به- انتها[1] است. در این مقاله استفاده از این نوع خاص کنترل‌کننده بر روی دسته خاصی از سیستم‌های انسان در حلقه یعنی ماشین‌های خودران هم در محیط شبیه‌سازی و هم در نمونه واقعی با ابعاد کوچک مورد تحلیل و بررسی قرار گرفته است. لازم به ذکر است در این تحقیق از کنترل‌کننده‌های تناسبی- انتگرال- مشتقگیر[2] و کنترل پیشبین مدل[3] که از معروفترین کنترل‌کننده‌های کلاسیک هستند، برای مقایسه با روش کنترلی انتها- به- انتها استفاده شده‌است. بررسی‌های ما نشان می‌دهد که این مقاله اولین مطالعه مقایسه‌ای کنترل‌کننده‌های ذکرشده در کاربرد ماشین خودران است.

در این مقاله در بخش ۲ پیش‌زمینه و کارهای مرتبط مورد بررسی قرار داده شده‌است. در بخش ۳ روش پیشنهادی شرح داده شده‌است و در بخش ۴ نتایج بدست‌آمده از آزمایش‌ها ارائه شده‌است. در پایان در بخش ۵ نتیجه‌گیری و جمع‌بندی ارائه شده‌است.

۲- پیش‌زمینه و کارهای مرتبط

۲-۱- کنترل کلاسیک

یکی از مهمترین اهداف در طراحی کلاسیک سیستم‌های کنترلی مدل‌سازی عدم قطعیت‌ها و اغتشاشات فیزیکی است و روش‌های متنوعی را شامل می‌شود. اما با گسترش سطح مجتمع‌سازی سیستم‌های کنترلی با فن‌آوری اطلاعات از یک سو و همچنین انسان محور بودن سیستم‌های انسان در حلقه از سوی دیگر، سیستم‌های کنترلی با عدم قطعیت از جانب رفتار انسان، دنیای فیزیکی و دنیای رایانشی مواجه است.

مسئله تحلیل و طراحی سیستم‌های کنترل کلاسیک به شکل‌های متفاوتی مطرح می‌گردد. این تفاوت بر اساس مؤلفه‌های تعداد ورودی- خروجی، دیدگاه داخلی از سیستم، خطی یا غیرخطی‌بودن عناصر سیستم، بهینگی در سیستم، پارامترها و میزان معلوم و نامعلوم بودن سیستم است. امروزه سیستم‌های فیزیکی- رایانشی که عمدتاً سیستم‌های نامعلوم با پارامترهای نامعین و متغیر هستند، بسیار مورد توجه قرارگرفته‌اند و راهکارهای کنترلی که برای این سیستم‌ها پیشنهاد می‌گردند شامل کنترل مقاوم ، کنترل تطبیقی ، هوشمند

1 End to End
2 proportional–integral–derivative (PID)
3 Model predictive control (MPC)

و کنترل تصادفی است. گسترش استفاده از این سیستم‌ها از یک سو و لزوم ارائه کنترل‌کننده مناسب برای آن‌ها از سوی دیگر سبب شده‌است که روزبه‌روز توجه بیشتری را هم از نظر تئوری و هم از نظر عملی به خود جلب نماید. از میان روش‌های ذکرشده روش‌های کنترلی تطبیقی یا خودتنظیم‌شونده یا خودکنترل‌شونده که مبتنی بر کنترل حلقه‌بسته دارای بازخورد هستند، بهتر از هر روش دیگری نیازمندی‌های سیستم‌های موردبحث را فراهم می‌نمایند [۴].

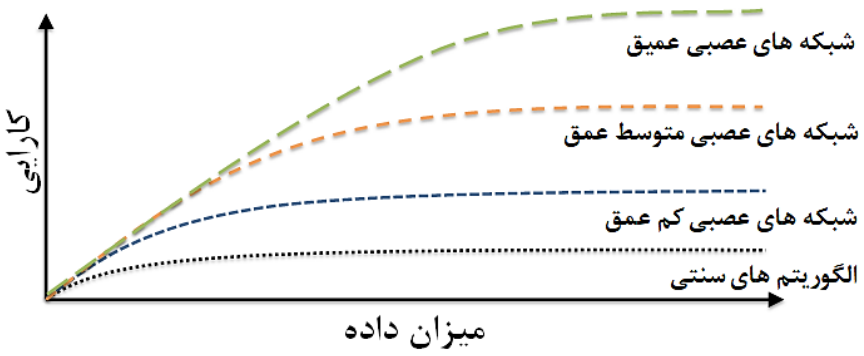
با توجه به شکل ۱، هدف سیستم‌های کنترلی خودکنترل‌شونده، کنترل سیستم‌های نامعلوم با پارامترهای نامعین و متغیر است. مطالعات تئوری مؤلفه‌های پایداری، مقاومت، کارایی و همگرایی الگوریتم پیشنهادی را موردتوجه و بررسی قرار می‌دهند. درحالیکه مطالعات عملی به پتانسیل آن به‌عنوان یک ابزار کنترلی ساده و کارآمد می‌پردازند. هر دو این مطالعات در کنار هم سبب ایجاد روشی مناسب برای کنترل می‌شوند. در گام اول بیشتر مطالعات بر روی کنترل تطبیقی سیستم‌های خطی صورت پذیرفته است. کنترل تطبیقی در اثر بررسی و حل مسائل ایده‌آل تحت شرایط ایده‌آل به وجود آمد که اثبات می‌کند که روش‌های کنترل تطبیقی تحت شرایط ایده‌آل پایدار هستند. پس‌ازآن مسئله مقاومت کنترل‌کننده تطبیقی با توجه به شرایط غیرایده‌آل مانند اغتشاشات خارجی و دینامیک مدل نشده، عدم قطعیت و همچنین نویز موردتوجه قرارگرفته که سبب پیدایش تعداد زیادی از روش‌های مختلف کنترل تطبیقی شده‌است. در همین زمان تلاش‌های بسیاری در جهت کاهش وابستگی به دانش قبلی از سیستم کنترل‌شونده و همچنین افزایش کارایی گذرا سیستم صورت پذیرفته است. در سال‌های اخیر کنترل تطبیقی سیستم‌های متغیر با زمان و غیرخطی در جهت رسیدن به هدف ارائه سیستم‌های خودکنترل‌شونده برای کاربردهای دنیای واقعی بسیار مورد توجه قرار گرفته است [۵].

کنترل یادگیرنده یکی از مهم‌ترین روش‌های پیاده‌سازی سیستم‌های خودکنترل شونده است، جایی که در آن به تقلید فرآیند یادگیری انسان پرداخته می‌شود. اگرچه خیلی از جنبه‌های یادگیری انسان هنوز ناشناخته باقی‌مانده اما پیشرفت‌های بسیاری در این زمینه بدست آمده است. به عنوان مثال در سیستم حرکتی یک ربات، یادگیری انسان به‌گونه‌ای تقلید می‌گردد که ربات توانایی اجرای یک حرکت پیچیده را با یادگیری قدم‌به‌قدم مهارت‌های لازم برای آن حرکت به دست آورد. لازم به ذکر است کنترل‌کننده‌های یادگیرنده برای سیستم‌هایی که عملیات یکسانی را تحت شرایط عملیاتی یکسان اجرا می‌کنند مطرح می‌شوند.

کنترل‌کننده غیریادگیرنده از خطا در هر مرحله سودی نمی‌برد درحالی‌که کنترل‌کننده یادگیرنده از این خطا به‌عنوان اطلاعاتی ارزشمند برای بهبود کارایی کنترل‌کننده در تکرارهای بعدی عملکرد موردنظر، بهره می‌برد. بدین ترتیب کنترل‌کننده باوجود عدم قطعیت بالای مدل و همچنین اغتشاشات می‌تواند به کارایی بالایی برسد. کنترل یادگیرنده یک تفاوت بنیادی با دیگر استراتژی‌های کنترل مانند کنترل تطبیقی و شبکه عصبی کنترلی دارد. کنترل تطبیقی و شبکه عصبی کنترلی در طول فرآیند یادگیری در جهت رسیدن به کارایی بالاتر به دست‌کاری و تنظیم پارامترهای کنترلی سیستم کنترل‌کننده می‌پردازند درحالیکه کنترل یادگیرنده از طریق تغییر ورودی‌های کنترلی که از جنس سیگنال هستند عملیات کنترل را انجام می‌دهد [۶].

چالش عمده در مسائل کنترلی مطرح است که برای بشر اجرای آن آسان است اما بیان رسمی و فرمول‌بندی آن بسیار مشکل است. این مسائل ادراکی به‌صورت خودکار و سریع توسط انسان حل می‌شوند (مانند تشخیص چهره‌ها در یک تصویر). راه‌حل کامپیوتری این مسائل این است که به کامپیوترها اجازه داده شود از تجارب درس بگیرند و جهان را بر مبنای سلسله مراتبی از مفاهیم درک کنند که این مفاهیم بر اساس روابطشان با مفاهیم ساده‌تر تعریف می‌گردند. همین امر این روش را از ارائه بیان رسمی و فرمول‌بندی یک مسئله تا حد زیادی بی‌نیاز می‌نماید. مسائل کنترلی در دنیای واقعی که روزمره با آن سروکار داریم بیشتر موضوعی و ادراکی هستند و همچنین دارای دینامیک پیچیده هستند و تحقیقات نشان داده است که استفاده از یادگیری عمیق در جهت پیاده‌سازی سیستم کنترلی مستقل از مدل کارایی خوبی ارائه می‌نماید [۷]. در بخش بعدی به طور مفصل در مورد یادگیری عمیق بحث خواهد شد.

به‌طور خلاصه با توجه به پیچیدگی روزافزون سیستم‌های فیزیکی-رایانشی، کنترل دستی این سیستم‌ها بسیار دشوار و غالباً غیرممکن است. بنابراین روش‌های کنترل خودکار در این شرایط پیشنهاد گردید. برای کنترل خودکار این سیستم‌ها اولین راهکار این است که به مدل ریاضی آن دست یافت و از طریق مدل، طراحی کنترل‌کننده



شکل ۲. نمودار بررسی مقیاس پذیری کارایی در ازای میزان داده بر گرفته از مرجع [۹]

Fig. 2. Performance scalability of algorithms with respect to training data set size [9]

و کنترل آن صورت پذیرد. اما این روش به دلیل پیچیدگی‌ها و عوامل بسیاری که در زمان اجرا در فرآیند کنترل تأثیرگذار هستند، در اکثر موارد امکان‌پذیر نیست. ازاین‌رو تمایل و گرایش مهندسان به سمت روش‌هایی از کنترل خودکار است که فرآیند کنترل را بی‌نیاز از تجربه، دانش و مدل از پیش تعیین‌شده سیستم تحت کنترل می‌نماید. از طرفی با توجه به نیازمندی‌های کاربردهای امروزی سیستم‌های انسان در حلقه، استفاده از روش‌های پیاده‌سازی سخت‌افزاری باقابلیت‌های بالای محاسباتی، دقت، انعطاف‌پذیری، قابلیت مقیاس‌پذیری و بهینگی در مصرف انرژی و اتکاپذیری بسیار موردتوجه است.

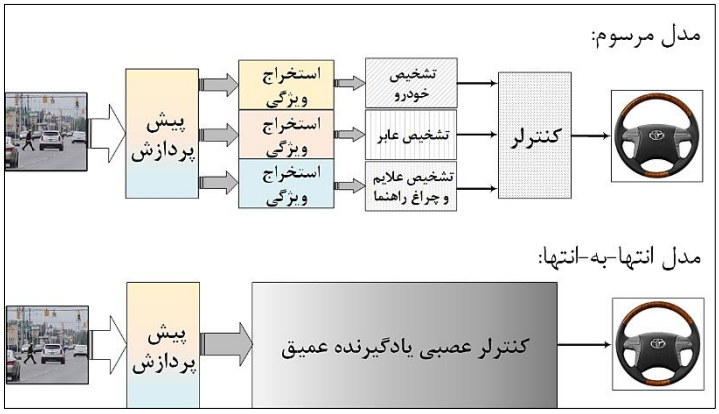
۲-۲- یادگیری عمیق

ایده اولیه یادگیری عمیق، در اواخر قرن بیستم با الهام از ساختار سلسله‌مراتبی سیستم ادراک و تولید گفتار انسان مطرح گردید. جهش بزرگ در یادگیری عمیق در سال ۲۰۰۶ زمانی رخ داد که آقای هینتون ساختار نوینی برای یادگیری عمیق به نام شبکه باور عمیق[1] ارائه نمود. در این ساختار فرآیند یادگیری متشکل از دو مرحله پیش آموزش و تنظیم دقیق است. در مرحله اول بردار مشاهده به‌عنوان ورودی در جهت پیش آموزش و تولید بردار داده میانی و همچنین از همه مهم‌تر تولید وزن‌های اولیه برای مرحله بعدی است. در مرحله بعدی (تنظیم دقیق) که در ساده‌ترین حالت از سه لایه تشکیل‌شده‌است، هسته اصلی تفاوت این نوع یادگیری با روش‌های قبلی یادگیری می‌شود و از آنجائیکه مبتنی بر یادگیری بدون نظارت است فاقد داده مقصد است و از طرفی مسئول تولید مجموعه مقادیر اولیه است به‌گونه‌ای که سبب یادگیری بهتر شبکه شود [۸].

اما سوال مهمی که مطرح است این است که با وجود اینکه ایده اولیه شبکه‌های عصبی عمیق در حدود ۴ دهه پیش مطرح شده چه عاملی سبب ظهور دوباره و پیشرفت های چشمگیر در این زمینه شده‌است. اولین و مهمترین پاسخ به این سوال توجه به این حقیقت است که کارایی نه تنها به میزان داده موجود در مرحله آموزش، بلکه به ساختار شبکه نیز وابسته است. سیر مقیاس‌پذیری کارایی به ازای افزایش داده در شکل۲ آورده شده‌است[۹]. همانطور که مشاهده می‌شود افزایش میزان داده به سرعت سبب بهبود کارایی همه روش‌ها می‌شود. اما در روش‌های سنتی به همان سرعت کارایی آن‌ها با افزایش بیشتر میزان داده به اشباع می‌رسد. از طرفی استفاده از شبکه‌های عصبی با تعداد لایه بیشتر سبب بهبود کارایی در ازای افزایش میزان داده است.

پاسخ دیگری که در راستای سوال مطرح‌شده لازم به ذکر است، پیشرفت های چشمگیر در ساخت بسترهای سخت‌افزاری پیاده‌سازی شبکه‌های عصبی عمیق است که بتوانند تسریع مناسبی نه تنها در فاز آموزش بلکه در فاز اجرا هم فراهم آورند. چرا که شبکه‌های عصبی عمیق بوضوح ازنظر محاسباتی بسیار هزینه‌بر و سنگین هستند. از طرف دیگر سخت‌افزار و نرم‌افزار پیاده‌سازی مدل یادگیری عمیق در کارایی آن نقش بسیار مهمی را ایفا می‌کند. تعداد نرون‌ها، تعداد اتصال‌های بین نرون‌ها، توپولوژی، پیکربندی و عملکرد هر یک از نرون‌ها در شبکه عصبی مصنوعی ازجمله مهم‌ترین ملاحظات طراحی

1 Deep Belief Network (DBN)



شکل ۳. مقایسه معماری مدل کلاسیک کنترل با مدل ETE برگرفته از [۱۱]

Fig 3. Classic control model vs. ETE model (adapted from [11])

هستند که تعیین‌کننده اصلی کارایی و دقت می‌باشند. از طرف دیگر تطبیق و تغییر شبکه در زمان اجرا نیز نیازمندی دیگری است که باید مورد توجه قرار بگیرد. مجموع این عوامل توجه خاص طراحان را معطوف به انتخاب دقیق روش پیاده‌سازی و همچنین چگونگی پیاده‌سازی می‌نماید که بتواند کارایی لازم را تضمین کند [۱۰].

به طور خلاصه، در روش‌های ارائه‌شده مبتنی بر یادگیری عمیق، روندهای زیر مشاهده می‌شوند که عبارت‌اند از:

• یادگیری عمیق در مسیر تاریخچه طولانی و غنی خود با اسامی مختلفی که برگرفته از نقطه‌نظرهای متفاوت فلسفی است مطرح شده‌است.

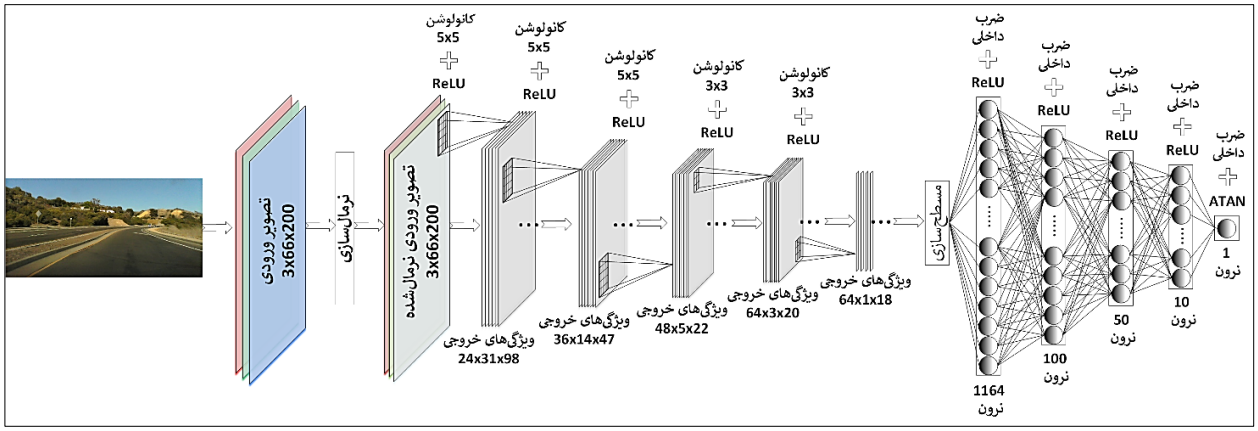
• یادگیری عمیق در جایی سودمندی بیشتری خواهد داشت که مقدار داده‌های آموزشی موجود افزایش یابد. لازم به ذکر است میزان داده مورد نیاز وابسته به نوع کاربرد و نوع داده آن و همچنین ساختار شبکه عصبی عمیق متفاوت است.

• مدل‌های یادگیری عمیق با بهبود زیرساخت‌های سخت‌افزاری و نرم‌افزاری، ازنظر سایز روزبه‌روز رشد می‌نماید. این بهبود در سخت‌افزارهایی چون واحد پردازنده گرافیکی[1]، واحد پردازشی تنسور[2] و آرایه‌های منطقی برنامه‌پذیر[3] با معرفی معماری‌های بهینه‌شده برای کاربرد یادگیری عمیق و در نرم‌افزار با ارائه ابزارهایی چون تنسورفلو[4]، کراس[5] و متلب[6] برای پیاده‌سازی الگوریتم‌ها و شبکه‌های یادگیری عمیق دیده می‌شود.

• یادگیری عمیق به حل دقیق‌تر مسائلی می‌پردازد که روزبه‌روز پیچیده‌تر می‌شوند و این دقت روزبه‌روز بیشتر می‌شود.

یکی از قدیمی‌ترین و پرکاربردترین ساختار ارائه‌شده برای شبکه عصبی عمیق، شبکه‌های عصبی کانولوشنی[7] است. این ساختار به طور کلی متشکل از لایه‌های استخراج‌کننده ویژگی در کنار لایه‌های تماما متصل یا طبقه‌بند است. لایه‌های استخراج‌کننده ویژگی، متشکل از هسته‌های مربعی با ابعاد فرد است (۳×۳، ۵×۵، ۷×۷ و ...) و تعداد متفاوت است. اعمال این هسته‌ها همانند اعمال فیلتر بر روی تصاویر، منجر به شکلگیری ویژگی‌های مختلف استخراج‌شده می‌گردند. مقادیر وزن‌های این هسته‌ها در زمان آموزش براساس داده‌های آموزش تعیین می‌شوند اما تعداد و ابعاد را طراحان در زمان تعیین ساختار شبکه مشخص می‌نمایند. تفاوت در ساختار بکارگیری این لایه ها در کنار استفاده از تعدادی از لایه‌های تماما متصل با تعداد نرون متفاوت در هرلایه، سبب شکلگیری شبکه های عصبی عمیق برای کاربردهای متفاوتی از جمله دسته‌بندی و قطعه‌بندی تصاویر می‌شود. کاربرد مورد نظر ما در این مقاله استفاده از شبکه عصبی عمیق برای کاربرد رگرسیون است یعنی نگاشت مستقیم تصاویر جاده به زاویه فرمان خودرو که برای انجام این مهم تغییراتی در شبکه

1 Graphics Processing Unit (GPU)
2 Tensor Processing Unit (TPU)
3 Filed Programmable Gate Arrays (FPGA)
4 TensorFlow
5 Keras
6 MATLAB
7 Convolutional Neural Network (CNN)



شکل ۴. ساختار شبکه عصبی عمیق استفاده شده در روش پیشنهادی

Fig. 4. Deep neural network structure utilized in our proposed method

عصبی عمیق دسته‌بند تصاویر صورت می‌پذیرد که در بخش روش پیشنهادی توضیح داده شده‌است.

۲-۳- یادگیری انتها- به- انتها

استفاده از روش‌های مبتنی بر ادراک برای حل مسئله ماشین خودران در گذشته گسترش چشمگیری داشته است. همانطور که در شکل ۳ نمایش داده شده‌است، در این روش‌ها مسئله به چندین زیر مسئله مانند تشخیص خودرو، تشخیص عابر، تشخیص علایم و چراغ راهنما و همچنین برنامه‌ریزی مسیر و کنترل سطح پائین شکسته می‌شود و این اجزا با همدیگر تشکیل یک خط لوله محاسباتی می‌دهند [۱۱]. اما نقاط ضعفی که ناشی از عدم یادگیری و تطبیق، پیچیدگی طراحی و خاص منظوره بودن استخراج‌کننده‌های ویژگی است سبب کاهش محبوبیت این روش‌ها و توجه بیشتر به سمت روش‌های انتها- به- انتها شده‌است. این روش از طریق تولید مستقیم خروجی‌های کنترلی از روی ورودی سنسورها، خط لوله محاسباتی مورد نیاز را به طور چشمگیری ساده‌سازی می‌نماید. سادگی مفهوم و همچنین الگوی محاسباتی روش‌های انتها- به- انتها سبب موفقیت این روش‌ها در کاربردهای مختلف هدایت وسائل نقلیه در محیط‌های داخلی و خارجی مانند اجتناب از موانع، سیستم هدایت خودکار خارج از جاده و رانندگی خودکار شده‌است [۱۲]. وظیفه کنترل با استفاده از این روش شامل بارکاری آموزش و بارکاری اجراست. بیشتر توجه در اکثر کارهای تحقیقاتی معطوف به بار کاری اجراست. مرحله آموزش به صورت برون- خطی بر روی پردازنده‌های گرافیکی انجام می‌شود با توجه به این مسئله که در این مرحله میزان برون‌دهی بستر سخت‌افزاری مورد توجه است. مرحله اجرا از نظر محاسباتی سبک‌تر از مرحله یادگیری است اما پارامتر تاخیر حتی بیشتر از میزان برون‌دهی اهمیت دارد بویژه در کاربردهایی که دارای قیدها و نیازمندی‌های اکید بلادرنگی هستند [۱۳].

خط لوله مربوط به بارکاری اجرا متشکل از مراحل زیر است:

- در قدم اول می‌بایست از طریق حسگرهای مختلف قادر به درک محیط پیرامون خود باشد
- در قدم بعدی می‌بایست داده‌های درک‌شده را در راستای استخراج ویژگی‌های مورد نیاز پردازش نماید
- پس از آن سیستم در نقش کنترل‌کننده موتورها به برنامه‌ریزی و کنترل رانندگی وسیله نقلیه می‌پردازد

این روش در بسیاری از مسائل کنترلی بلادرنگ شامل رباتیک وکاربردهای دیگر سیستم‌های فیزیکی- رایانشی و بحرانی- ایمن مورد بررسی قرار گرفته است [۱۲].

۲-۴- ساختار کنترل‌کننده عصبی یادگیرنده عمیق

در طراحی کنترل‌کننده‌های عصبی یادگیرنده عمیق مهمترین مسئله انتخاب ساختار شبکه مناسب جهت کاربرد مورد نظر است. در کاربرد کنترل ماشین‌های خودران، در روش انتها- به- انتها لازم است تا یک شبکه عصبی عمیق به طور مستقیم از رفتار انسان در حین

رانندگی یاد بگیرد و در زمان اجرا از طریق نگاشت تصاویر دوربین به زاویه چرخش فرمان، میزان فشار پدال گاز و میزان قدرت ترمز گرفتن، وسیله نقلیه را کنترل و مسیریابی نماید.

استفاده از روش یادگیری انتها- به- انتها در کاربرد ماشین خودران ابتدا در اواخر سال ۱۹۸۰ ارائه گردید که در آن نویسندگان سعی داشتند که از شبکه‌های عصبی مصنوعی کاملا متصل سه لایه برای هدایت‌کردن وسیله نقلیه استفاده کنند [۱۴]. پس از آن در قالب پروژه وسیله نقلیه خودکار در آژانس پروژه‌های تحقیقاتی پیشرفته دفاعی ایالات متحده از شبکه‌های عصبی کانولوشنی شش لایه برای کنترل انتها- به- انتها استفاده شد [۱۵]. در سال ۲۰۱۶ در شرکت انویدیا درپروژه‌ای از ساختار ۹ لایه (شکل۴) به شکلی استفاده شد که به صورت مستقیم تصاویر دوربین به دستورات کنترلی برای موتورها تبدیل شود و کلیه لایه‌های میانی سنتی که به صورت دستی طراحی می‌شدند حذف شدند [۱۶-۱۸].

این ساختار ۹ لایه با استفاده لایه‌های کانولوشنی به استخراج ویژگی‌های تصاویر ورودی می‌پردازد و سپس عملیات دسته‌بندی در لایه‌های تماما متصل شکل می‌گیرد. تک نرون خروجی نتیجه این عملیات را به زاویه خروجی تبدیل می‌کند. عملکرد مناسب این شبکه کنترلی برای کاربرد ماشین خودران و از سوی دیگر ساختار نه‌چندان پیچیده آن سبب شده‌است تا در روش پیشنهادی از آن استفاده شود. در روش پیشنهادی بخش‌هایی به این شبکه اضافه شدند که در بخش بعدی توضیح داده می‌شوند.

۳- روش پیشنهادی

همانطورکه در بخش قبل توضیح داده شد، ساختار شبکه عصبی عمیق کنترلی بر پایه شبکه پایلوت[1] است. علاوه بر این در راستای بهبود عملکرد و همچنین تطبیق آن با بستر آزمایش تغییراتی بر روی آن صورت گرفته‌است. در مرحله اول بر روی تصاویر ورودی پیش‌پردازش انجام می‌شود. این پیش‌پردازش می‌تواند شامل کاهش ابعاد ورودی، اعمال فیلتر، کوانتیزه‌کردن، کاهش حجم داده و تغییر در مقادیر پیکسل و... باشد. در روش پیشنهادی از نرمال‌سازی مقادیر پیکسل‌ها به مقدار میانه برای نرمال‌سازی توزیع مقادیر آن‌ها استفاده شده‌است. در این ساختار با استفاده از ۳ لایه کانولوشن با اندازه هسته ۵×۵ و پس از آن از ۲ لایه کانولوشن ۳×۳ به دنبال یافتن ویژگی‌های مختلف تصویر مانند لبه به صورت خودکار هستیم. ویژگی‌های تولیدشده طی عملیات مسطح‌سازی به صورت تک بعدی وارد لایه اول شبکه تماما متصل با ۱۱۶۴ نرون می‌شوند. لایه‌های بعدی شبکه تماما متصل به ترتیب ۱۰۰، ۵۰ و ۱۰ نرون محاسباتی دارند که عملیات ضرب داخلی ورودی‌ها و وزن‌ها را انجام می‌دهند. تابع فعالیت همه لایه‌ها تابع واحد یکسوشده‌ی خطی[2] است به جز تک نرون خروجی که تابع آرکتانژانت[3] است (به این تک نرون در خروجی لایه رگرسیون نیز گفته می‌شود). دلیل این امر این است که بازه مقادیر تابع واحد یکسوشده‌ی خطی از ۰ تا ۱ است اما آرکتانژانت می‌تواند در بازه ۹۰- تا ۹۰+ در خروجی عدد تولید نماید. اضافه‌شدن این تک نرون به انتهای شبکه در حقیقت شبکه دسته‌بند تصاویر را به شبکه رگرسور تبدیل می‌کند که قادر به انجام عملیات رگرسیون برای تولید مستقیم تصاویر جاده به زاویه فرمان خودرو است.

روش کار استفاده‌شده شامل آموزش شبکه‌های عصبی عمیق برای استفاده در کنترل انتها- به- انتها ماشین خودران، اجرا در محیط واقعی و مقایسه با روش‌های سنتی در محیط شبیه‌سازی و همچنین بهینه‌سازی روش پیشنهادی است. روال کلی کار در شکل ۶ نمایش داده شده‌است. همانطور که در شکل نمایش داده شده‌است، این روال از دو فاز اصلی آموزش و اجرا تشکیل شده‌است. پس از اجرای این دو فاز اصلی، در روش پیشنهادی قابلیت پیاده‌سازی و مقایسه تعبیه شده‌است که در ادامه به شرح آن می‌پردازیم.

۳-۱- فاز آموزش

با توجه به اینکه روش کنترلی انتها- به- انتها داده- محور است، برای آموزش کنترل‌کننده عصبی عمیق لازم است تا ابتدا داده‌های آموزش تولید شوند. در این مرحله برای تولید داده‌ها از محیط شبیه‌سازی مبتنی بر موتور بازی‌سازی یونیتی[4] استفاده شد. این محیط دارای قابلیت‌ها و ابزار لازم جهت رانندگی خودکار/غیرخودکار، ضبط رویدادها و همچنین تولید خروجی‌های لازم از قبیل تصویر سه دوربین (چپ، راست، وسط)، زاویه ماشین، سرعت و میزان ترمز است [۱۹]. در فاز آموزش پس از جمع‌آوری داده‌های رانندگی انسان،

1 Pilot Net

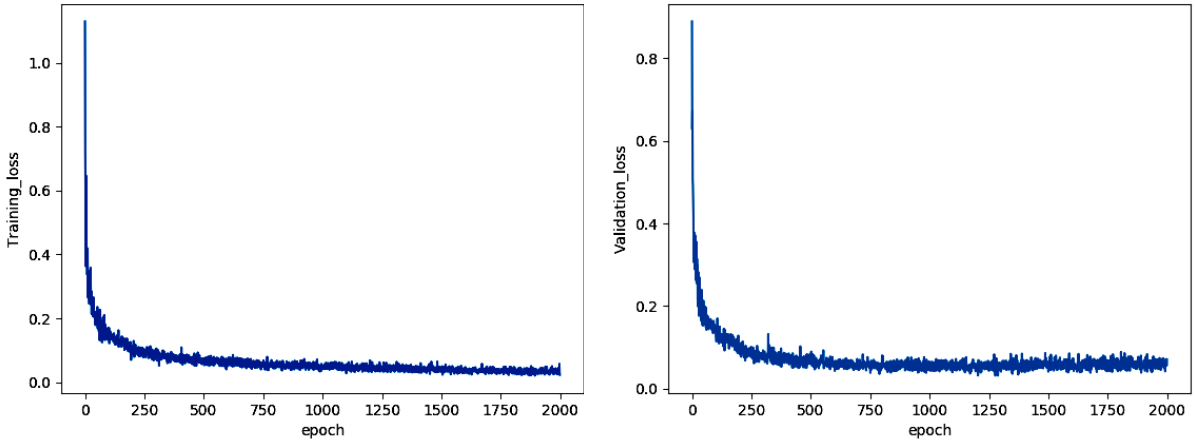
2 ReLU
3 Arctangent
4 Unity

جدول ۱. پیکربندی فاز آموزش

Table 1. Parameters of training phase

پارامتر	مقدار
الگوریتم بهینه‌سازی	آدام[1]
نرخ آموزش اولیه	۱×۱۰^{-۴}
نرخ حذف تصادفی	۰/۵
حداکثر تعداد دوره[2] آموزش	۲۰۰۰
اندازه دسته[3]	۱۰۰

[1] Adam



شکل ۵. نمودار مقادیر loss فاز آموزش

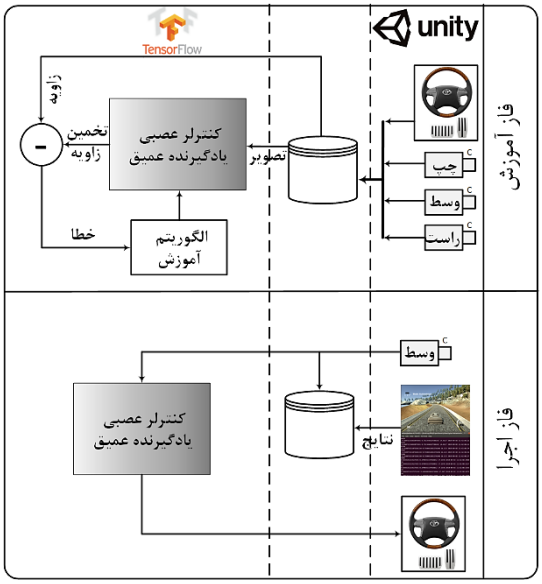
Fig. 5. Training and validation loss

این داده‌ها در فرمت مورد نظر خروجی گرفته می‌شوند و سپس بر روی دیسک ذخیره می‌شوند. لازم به ذکر است داده‌هایی که انسان بدون خطا مسیر درنظرگرفته‌شده را طی می‌کند برای آموزش شبکه کنترلی انتخاب می‌شوند. این داده‌ها شامل تصویر، زاویه فرمان ومیزان گاز و ترمز است. در مرحله بعدی با استفاده از ابزار یادگیری عمیق تنسورفلو (نسخه ۲) ابتدا ساختار مورد نظر پیاده‌سازی و سپس الگوریتم آموزش شبکه عمیق برنامه‌نویسی می‌شوند.

پس از این مرحله الگوریتم آموزش، تصاویر و زاویه‌های جمع‌آوری‌شده از رانندگی انسان را بارگذاری می‌کند و پارامترهای شبکه عصبی عمیق را در راستای حداقل‌سازی خطا آموزش می‌دهد. در این مرحله می‌توان با بهره‌گیری از پارامترهای مختلف شامل الگوریتم بهینه‌سازی، نرخ یادگیری، تعداد تکرار آموزش و همچنین نحوه دسته‌بندی داده‌های آموزش فرآیند آموزش را در راستای بدست‌آوردن دقت و سرعت آموزش مورد نظر تنظیم نمود. پیکربندی مرحله آموزش در جدول ۱ آورده شده‌است. یکی از مهمترین تفاوت‌های ساختار شبکه در فاز آموزش و اجرا استفاده از لایه حذف تصادفی[1] است. به این معنی که به عدد تعیین‌شده برای این پارامتر احتمال حذف برای نرون‌ها در فاز آموزش در نظر گرفته می‌شود و نرون مذکور نادیده گرفته می‌شود. محل قرارگرفتن این لایه در ساختار استفاده‌شده در این مقاله، بعد از اولین لایه تماما متصل است. لازم به ذکر است بیشتر مقادیر پیکربندی با استفاده از سعی و خطا و پارامترهای پیشنهادی در مقالات تعیین شده‌است.

در فاز آموزش ماشین خودران سایز کوچک و در محیط واقعی،

1 Dropout



شکل ۶. روش پیشنهادی برای آموزش و اجرای کنترل‌کننده ETE

Fig. 6. Proposed model for both training and inference phases of ETE control system

ماشین خودران سایز کوچک توسط انسان طی ۳۱ مرحله بر روی مسیر نمایش‌داده‌شده در شکل ۱۰، هدایت شده و داده‌های مربوط به دوربین و هدایت انسان برروی حافظه برد پردازشی ذخیره می‌شود. لازم به ذکر است در هر بار اجرای این مرحله، نقطه شروع و زاویه حرکت به صورت تصادفی و غیرتکراری تعیین شده‌است.

پس از اتمام این مرحله، فایل‌های ذخیره‌شده بر روی حافظه برد پردازشی به کامپیوتر برای اجرای فاز آموزش انتقال داده می‌شوند. این فایل‌ها به سه دسته شامل آموزش، تست و ارزیابی تقسیم می‌شوند. نسبت درنظرگرفته برای این تقسیم‌بندی به نسبت کل داده‌ها به ترتیب عبارتست از: ۷۰، ۲۰ و ۱۰ درصد. پس از آن الگوریتم آموزش شبکه عصبی عمیق در ابزار تنسورفلو اجرا، و وزن‌ها و پارمترهای شبکه تعیین می‌شوند. لازم به ذکر است پیکربندی آزمایشها شامل تعداد و نوع لایه‌های شبکه عصبی عمیق و همچنین پارامترهای الگوریتم آموزش مانند آزمایش‌های بخش ۴ در نظر گرفته شده‌است. الگوریتم آموزش پس از طی ۱۰ گام، به دقت آموزش ۹۹ درصد و دقت ارزیابی ۹۸ درصد رسید.

فاز آموزش بر روی کامپیوتر با مشخصات جدول ۹ پیاده‌سازی و اجرا شده‌است. نکته قابل توجه در مورد مرحله آموزش این است که در این مرحله در نقطه‌های مختلف، در بازه‌های زمانی مشخص، از مقادیر بدست‌آمده نمونه‌برداری می‌شود و در انتها بهترین نمونه به عنوان مقادیر نهایی پارامترهای شبکه ذخیره می‌شوند. نمودار مقادیر تابع هزینه[1] برای آموزش و مرحله اعتبارسنجی[2] در شکل ۵ آورده شده‌است. پس از اتمام این مرحله، پارامترهای تعیین‌شده در مرحله آموزش می‌تواند در فاز اجرا مورد استفاده قرار بگیرد.

۳-۲- فاز اجرا

در این مرحله شبکه عمیق آموزش‌دیده در محیط تنسورفلو بارگذاری می‌شود و تصویر دوربین وسط به آن اعمال می‌شود و زاویه تخمین‌زده‌شده و همچنین میزان فشار پدال گاز و ترمز به محیط شبیه‌سازی ارسال می‌شود. محیط شبیه‌سازی براساس این زاویه و سرعت تعیین‌شده توسط کنترل‌کننده عامل ماشین خودران را در مسیر تعیین‌شده به حرکت در می‌آورد. این مسیر دارای اشیاء و مناظر مختلف و همچنین جاده با الگوهای متفاوت است.

در مورد اجرای شبکه بر روی بستر واقعی آزمایش، در این مرحله شبکه آموزش دیده به برد پردازشی جهت اجرا و ارزیابی بر روی مسیر واقعی منتقل می‌شود. در فاز اجرا، ماشین خودران در تکرارهای

1 Loss Function
2 Validation

مختلف در نقاط مختلف مسیر و زاویه شروع به حرکت مختلف قرار داده می‌شود و فرمان شروع از طریق نرم افزار واسط که به زبان پایتون نوشته شده‌است، صادر می‌شود. برد پردازشی شبکه عصبی آموزش‌دیده را از طریق ابزار تنسورفلو نصب‌شده بر روی سیستم عامل خود اجرا و نرم‌افزار واسط تصاویر ورودی از دوربین را به شبکه اعمال و خروجی شبکه را به راه‌انداز موتورها ارسال می‌نماید. نرم‌افزار واسط به صورتی طراحی شده‌است که می‌تواند کنترل ماشین خودران را از حالت خودکار به حالت دستی و هدایت انسان تغییر دهد. علاوه بر تغییر نقطه و زاویه شروع حرکت، سرعت‌های مختلف نیز برای حرکت ماشین نیز در نظر گرفته شده‌است.

۳-۳- مقایسه

در مرحله مقایسه از دو کنترل‌کننده کلاسیک پرکاربرد یعنی کنترل‌کننده‌های تناسبی- انتگرال- مشتقگیر و کنترل پیشبین مدل استاندارد استفاده شده‌است. از آنجائیکه این کنترل‌کننده‌ها بدون قابلیت یادگیری هستند در مرحله اجرا به جای استفاده از تنسورفلو و کنترل‌کننده عصبی یادگیرنده عمیق، از آن کنترل‌کننده‌ها استفاده شده‌است. در ادامه به تعریف این کنترل‌کننده‌ها و چالش‌هایی که در استفاده از آن‌ها وجود دارد پرداخته می‌شود.

۳-۳-۱- کنترل‌کننده تناسبی- انتگرال- مشتقگیر

یکی از متداولترین انواع کنترل‌کننده بازخوردی موجود کنترل‌کننده تناسبی- انتگرال- مشتقگیر است. این کنترل‌کننده به صورت پیوسته یک تابع خطا محاسبه می‌کند و به سیستم کنترل‌شونده به گونه‌ای فرمان‌های اصلاحی ارسال می‌کند تا خطا به حداقل برسد. براساس رابطه (۱) این کار براساس تجمیع سه جمله تناسب (P) ، انتگرال (I) و مشتق (D) صورت می‌پذیرد.که در آن e بیانگر خطا و u خروجی کنترل‌کننده است.

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \qquad (۱)$$

لازم به ذکر است رفتار این کنترل‌کننده به مقدار تعیین‌شده برای ضرایب یا بهره‌های K_P, K_I, K_D بستگی دارد. در ادامه در بخش نتایج به چگونگی این وابستگی پرداخته می‌شود.

۳-۳-۲- کنترل‌کننده کنترل پیشبین مدل

این کنترل‌کننده که یکی از روش‌های کنترلی پیشرفته است، از مدل ریاضی دینامیکی سیستم تحت کنترل برای پیش‌بینی رفتار آینده آن استفاده می‌کند و دنباله‌ای از ورودی‌های کنترلی بهینه به گونه‌ای محاسبه می‌نماید که محدویت‌های داده‌شده را نقض نکند و همچنین کارایی بهینه مورد نظر را بدست آورد. مدل دینامیکی سیستم می‌تواند به صورت رابطه گسسته- زمان (رابطه ۲) در نظر گرفته شود:

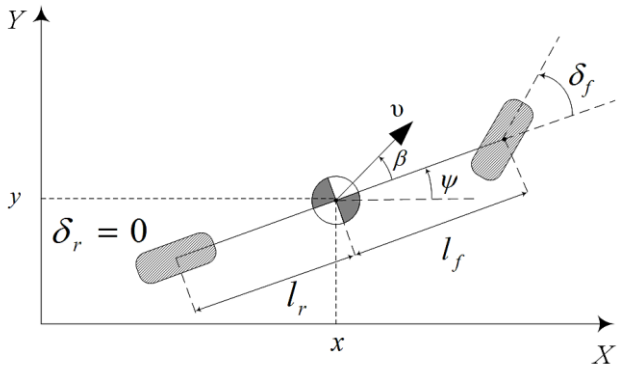
$$x_{k+1} = f(x_k, u_k) \qquad (۲)$$

که در آن $x_k \in R^n$ بیانگر وضعیت سیستم در زمان k است و $u_k \in R^m$ ورودی کنترلی است. یکی از نقاط قوت کنترل‌کننده MPC، قابلیت اعمال آن به سیستم‌های غیرخطی، درنظرگرفتن محدودیت‌های کلی و همچنین وظایف کنترلی متفاوت به نسبت سیستم‌های کنترلی کلاسیک دنبال‌کننده نقطه هدف[1] است. چالش مهم استفاده از کنترل‌کننده‌های کنترل پیشبین مدل این است که پیاده‌سازی آنها نیازمند یک حل برخط از مسئله بهینه‌سازی که در رابطه (۳) آورده شده‌است، در هر زمان نمونه‌برداری است.

$$\begin{aligned} \min_{u_k} \quad & \sum_{k=0}^{N-1} J(x_k, u_k) \\ \textit{subject to} \quad & x_{k+1} = f(x_k, u_k), x_0 = \hat{x} \\ & g(x_k, u_k) \le 0 \end{aligned} \qquad (۳)$$

در این رابطه g، توصیف‌کننده محدودیت‌های کلی و $\hat{x}$ وضعیت فعلی سیستم است که این کمیت می‌بایست به ازای هر محاسبه برای بدست‌آوردن دستور کنترلی یا اندازه گیری شود یا تخمین زده شود. معیار کارایی در حقیقت همان تابع هزینه است که با J بیان می‌شود و N پنجره پیش‌بینی است. در هر مرحله از دنباله کنترلی بهینه

1 Set Point Tracking



شکل۷. مدل خودرو

Fig. 7. Vehicle's model

محاسبه‌شده یعنی $(u_0,....,u_{N-1})$ تنها اولین عنصر u به سیستم اعمال می‌شود و در مرحله بعدی مسئله بهینه‌سازی یکسانی برای وضعیت جدید سیستم محاسبه می‌شود.

اگر چه در سال های اخیر پیشرفت‌های بسیاری در سخت‌افزارها و الگوریتم‌های پیاده‌سازی این روش کنترلی بدست آمده است، اما برای کاربردهایی که به زمان نمونه‌برداری در محدوده میکروثانیه نیاز باشد حل معادلات و رابطه‌های ذکرشده در زمان اجرا می‌تواند چالش بسیار بزرگی باشد [۳].

۴- نتایج

در این بخش در ابتدا به صورت کیفی کنترل‌کننده یادگیرنده عمیق با کنترل‌کننده‌های کلاسیک تناسبی- انتگرال- مشتقگیر و کنترل پیشبین مدل مقایسه می‌شوند. سپس براساس نتایج بدست‌آمده از آزمایش‌ها این کنترل‌کننده‌ها برای کاربرد ماشین خودران به صورت کمی با یکدیگر مقایسه می‌شوند.

۴-۱- وابستگی به مدل

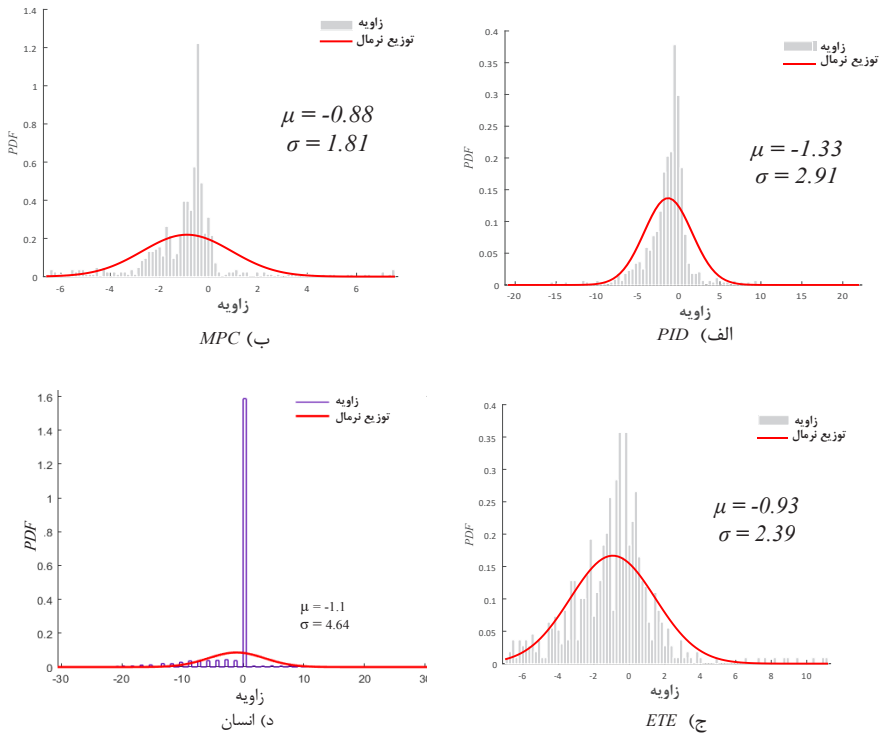
هر دو کنترل‌کننده تناسبی- انتگرال- مشتقگیر و کنترل پیشبین مدل نیازمند مدل دقیقی از سیستم کنترل شونده هستند. البته نوع وابستگی به مدل برای این دو کنترل‌کننده متفاوت است. برای کنترل‌کننده تناسبی- انتگرال- مشتقگیر مدل سیستم کنترل شونده برای تنظیم ضرائب یا بهره‌های کنترلی برای بدست‌آمدن حالت بهینه مورد نیاز است در حالیکه در کنترل‌کننده کنترل پیشبین مدل، برای پیش‌بینی رفتار آینده سیستم کنترل شونده مورد نیاز است. در این مقاله از مدل پیشنهادی [۲۰] برای خودرو استفاده شده‌است. این مدل در شکل ۷ مشخص شده است. براساس این مدل روابط غیر خطی پیوسته- زمان مربوط به حرکت خودرو عبارتند از:

$$\beta = \tan^{-1}\left(\frac{l_r}{l_r + l_f}\tan(\delta_f)\right) \qquad (۴)$$
$$\dot{x} = \upsilon\cos(\psi + \beta)$$
$$\dot{y} = \upsilon\sin(\psi + \beta)$$
$$\dot{\psi} = \frac{\upsilon}{l_r}\sin(\beta)$$
$$\dot{\upsilon} = a$$

در کاربرد ماشین‌های خودران نه تنها خودرو بلکه محیط پیرامون دارای دینامیک پیچیده و عدم قطعیت هستند که بیان ریاضی آن بسیار دشوار و معمولا غیردقیق است. این در حالیست که کنترل‌کننده انتها- به- انتها به مدل دقیقی از خودرو و محیط نیاز ندارد. از این نقطه نظر این کنترل‌کننده‌ها به دو روش کلاسیک کنترلی برتری دارند.

۴-۲- وابستگی به پارامترها

همانطور که قبلا هم بیان شد رفتار کنترل‌کننده تناسبی- انتگرال- مشتقگیر به پارامترهای K_p,K_I,K_D وابستگی دارد. این



شکل ۸. توزیع آماری خروجی زاویه کنترل‌کننده‌ها و انسان

Fig. 8. Statistical distribution functions of output angles (PID, MPC, ETE, and human)

کنترل‌کننده به صورت پیوسته بر اساس مشتق، انتگرال و جمع خطا و ضرایب آن‌ها یک فرمان اصلاحی به سیستم کنترل‌شونده ارسال می‌کند. ضریب K_p تعیین‌کننده میزان تاثیر بزرگی خطا در تغییر خروجی است. اگر میزان این ضریب به درستی تعیین نشود می‌تواند ناپایدار شود. ضریب انتگرال یعنی K_I هم در بزرگی و هم در مدت زمان خطا مشارکت دارد. این جمله باعث حذف خطای حالت پایدار باقیمانده که مشکل اساسی کنترل‌کننده‌های P نیز هست، می‌شود. ضریب مشتق خطا، K_d تعیین کننده میزان تاثیر تغییر نرخ خطا در فرمان کنترلی است. به عبارت دیگر هر چقدر به نقطه هدف مطلوب نزدیکتر شویم می‌بایست فرمان کنترلی نرمتری صادر شود تا جهش اضافه در سیستم اتفاق نیافتد. این ضریب به پایداری و زمان استقرار سیستم کمک می‌کند.

با توجه به مطالب بالا تعیین دقیق ضرایب، تعیین‌کننده کارایی کنترل‌کننده تناسبی- انتگرال- مشتقگیر است و این درحالیست که این موضوع نیازمند در اختیارداشتن مدل ریاضی دقیق از سیستم کنترل‌شونده است. تعیین این ضرایب حتی با داشتن تخصص و در اختیار داشتن مدل دقیق در صورتیکه سیستم پیچیده باشد، دشوار است.

کنترل‌کننده کنترل پیشبین مدل، مبتنی بر مدل دینامیکی سیستم کنترل‌شونده است و بر خلاف کنترل‌کننده تناسبی- انتگرال- مشتقگیر قابلیت پیشبینی رخدادهای آینده و تصمیم‌گیری کنترلی براساس آن را دارد. در حقیقت رفتار کنترلی بازه زمانی فعلی این کنترل‌کننده براساس بازه های زمانی آینده بهینه‌سازی می‌شود. رفتار این کنترل‌کننده علاوه بر مدل دینامیکی، به تعیین دقیق محدودیت‌های سیستم کنترل‌شونده مانند محدودیت پاسخ عملگرهای سیستم و همچنین یک تابع هزینه دقیق است که در جهت بهینه‌سازی رفتار این کنترل‌کننده مورد استفاده قرار می‌گیرد.

محدوه پیش‌بینی یکی دیگر از پارامترهای مهم در کارایی کنترل‌کننده کنترل پیشبین مدل است. این پارامتر از رابطه (۵) بدست می‌آید.

جدول ۲. مشخصات آماری زاویه های خروجی (برحسب درجه)

Table 2. Characteristics of output angles (in terms of degree)

کنترل‌کننده	واریانس	انحراف معیار	حداکثر	حداقل
تناسبی-انتگرال-مشتقگیر	۶/۲۹	۲/۵۰	۲۱/۹۹	۰/۰۰۱۵
کنترل پیشبین مدل	۲/۲۳	۱/۴۹	۷/۵۵	۰/۰۰۴۱
انتها-به-انتها	۳/۲۶	۱/۸۰	۱۱/۱۷	۰/۰۰۲۷
انسان	۲۱/۵۴	۴/۶۴	۳۰	۰

جدول ۳. داده‌های آماری کنترل سرعت (برحسب کیلومتر بر ساعت)

Table 3. Statistical parameters of speed control (in terms of kph)

کنترل‌کننده	واریانس	انحراف معیار	حداکثر	حداقل
تناسبی- انتگرال- مشتقگیر	۱۸/۷۱	۴/۳۲	۴۲/۴	۰
کنترل پیشبین مدل	۳۴/۶۴	۵/۸۸	۵۱/۶۶	۰
انتها- به- انتها	۱۹/۴۳	۴/۴۰	۴۲/۷۳	۰
انسان	۱۱/۲۹	۳/۳۶	۳۵/۳۲	۰

$$T = N * dt \tag{۵}$$

که در آن N تعداد بازه های زمانی و dt طول هر بازه است. افزایش بیش از اندازه dt سبب افزایش زمان بین تصمیم‌گیری کنترل‌کننده می‌شود که این سبب می‌شود که سیستم به سختی بتواند یک مسیر مرجع پیوسته را دنبال نماید. که با آن خطای گسسته سازی گفته می‌شود. اگر چه افزایش اندازه T سبب افزایش محدوده پیش‌بینی می‌شود اما در دنیای واقع پیش‌بینی زمان‌های خیلی دور، منطقی به نظر نمی‌رسد و در شبیه‌سازی انجام‌شده افزایش بیش از حد این پارامتر سبب خروج خودرو از خط می‌شود.

۴-۳- بار محاسباتی

بار محاسباتی کنترل‌کننده‌های تناسبی- انتگرال- مشتقگیر به پارامترهای کنترلی وابستگی ندارد و تنها به نرخ نمونه‌برداری سیستم وابسته است و بار محاسباتی کمتری نسبت به دو کنترل‌کننده دیگر دارد. در حالیکه در کنترل‌کننده کنترل پیشبین مدل بار محاسباتی با افزایش تعداد متغیرهایی که می‌بایست برای بهینه‌سازی در نظر بگیرد، افزایش می‌یابد. تعداد متغیرهای بهینه‌سازی متناسب با N است و چالش کار این است که T بزرگ و dt کوچک منجر به N بزرگ می‌شود. کنترل‌کننده‌های انتها- به- انتها، از نظر بار محاسباتی وابسته به ابعاد ورودی، ساختار و ابعاد لایه‌های تشکیل‌دهنده آن هستند.

نکته مهم دیگر که از این نقطه‌نظر اهمیت دارد وابستگی کنترل پیشبین مدل به الگوریتم‌های تشخیص خط است که موجب تحمیل بارکاری اضافه علاوه بر بار کاری اجرای الگوریتم کنترلی به بستر پیاده‌سازی است. در حالیکه روش انتها- به- انتهای پایه علاوه بر محاسبات لایه‌های اصلی نیازی به محاسبات اضافه دیگری ندارد.

۴-۴- توزیع آماری زاویه خروجی

در این بخش رفتار کنترل‌کننده‌های مورد بررسی از نظر زاویه خروجی محاسبه‌شده توسط آنها مورد بررسی قرار داده شده‌است. این بررسی از جهت استهلاک خودرو، آسایش سرنشینان و ایمنی مورد توجه است. در شکل ۸ تابع چگالی احتمال زاویه خروجی کنترل‌کننده‌های

جدول ۴. استفاده از ترمز
Table 4. Utilization of brake system

کنترل‌کننده	تعداد دفعات تخطی از آستانه	حداکثر شتاب ترمز (m/s^2)	واریانس شتاب (m/s^2)
تناسبی- انتگرال- مشتقگیر	۹	۸/۰۹	۱/۸۲
کنترل پیشبین مدل	۱۷۷	۸/۴	۳۰/۷۵
انتها- به- انتها	۲۹	۸/۹۴	۱۸/۲

جدول ۵. خطای تصمیم‌گیری
Table 5. Decision making errors

کنترل‌کننده	خطای سرعت	خطای ترمز	خطای گاز	خطای زاویه فرمان
تناسبی- انتگرال- مشتقگیر	۱۶۱/۸۳	۱×۱۰^{-۴}	۰/۰۵۷۱	۰/۰۳۶۵
کنترل پیشبین مدل	۲۴۰/۰۷	۰/۲	۰/۵۲۸۲	۰/۰۲۶۵
انتها- به- انتها	۱۶۸/۷۶	۰	۰/۰۵۳۷	۰/۰۳۲۳

تناسبی- انتگرال- مشتقگیر (الف)، کنترل پیشبین مدل (ب)، انتها- به- انتها (ج) و رانندگی توسط انسان (د) آورده شده‌است.

در این شکل نزدیکترین توزیع نرمال برای تابع چگالی احتمال داده‌های خروجی کنترل‌کننده‌ها و همچنین میانگین و انحراف از معیار نیز گزارش شده‌است. مشخصات آماری زاویه‌های خروجی کنترلی برحسب درجه در جدول ۲ آورده شده‌است.

این نتایج نشان می‌دهد تناسبی- انتگرال- مشتقگیر دستورات با زاویه تند (میانه و حداکثر بیشتر) و میزان تغییرات بیشتر (انحراف از معیار بیشتر) صادر می‌کند و این امر سبب فشار و استهلاک بیشتر به خودرو، ایجاد حالت نامطلوب برای سرنشینان و همچنین ایمنی کمتر در شرایط لغزندگی جاده می‌شود. از این نقطه نظر کنترل پیشبین مدل بهترین رفتار را از خود نشان می‌دهد و کنترل‌کننده انتها- به- انتها با اختلاف کم نسبت به آن عملکرد بهتری نسبت به تناسبی- انتگرال- مشتقگیر از خود نشان می‌دهد. علاوه بر این زوایای خروجی از کنترل پیشبین مدل میانه نزدیکتری به زاویه صفر دارند و از این نظر کنترل‌کننده انتها- به- انتها عملکرد بهتری نسبت به تناسبی- انتگرال- مشتقگیر دارد.

۴-۵- کنترل سرعت

در این بخش نحوه کنترل سرعت کنترل‌کننده‌ها مورد توجه است. این پارامتر با توجه به ملاحظات ایمنی، مصرف سوخت و انرژی و همچنین زمان رسیدن به مقصد مورد توجه است. در آزمایش‌های انجام‌شده برای این مقاله حداکثر سرعت ۳۰ مایل بر ساعت (۴۸,۲ کیلومتر بر ساعت) تعیین شده‌است. همانطور که در جدول ۳ مشاهده می‌شود کنترل‌کننده کنترل پیشبین مدل این محدوده مجاز را نقض کرده است و کنترل‌کننده تناسبی- انتگرال- مشتقگیر نیز نمی‌تواند از سرعت مجاز بهره کافی ببرد و در صورت اجبار کنترل‌کننده به استفاده از سرعت های بالاتر منجر به خروج خودرو از جاده می‌شود. در حالیکه کنترل‌کننده انتها- به- انتها محدوده مجاز سرعت را نقض نمی‌کند و تا حد ممکن به حداکثر سرعت مجاز نزدیک می‌شود.

از نظر سرعت رسیدن به مقصد کنترل پیشبین مدل و بعد از آن کنترل‌کننده انتها- به- انتها بهترین رفتار را دارند این در حالیست که نوسانات شدید سرعت در کنترل پیشبین مدل هم سبب ایجاد استهلاک و مصرف سوخت بیشتر و هم ایجاد حالت نامتعادل و احساس نامطلوب در سرنشینان می‌شود. اما کنترل‌کننده انتها- به- انتها استراتژی متعادلی از این نظر اتخاذ می‌کند که هم بتواند از حداکثر مجاز سرعت بهره کافی ببر و زمان رسیدن به مقصد طولانی نشود و هم اینکه تغییرات سرعت کمتری به نسبت کنترل پیشبین مدل داشته باشد.

۴-۶- تغییرات شتاب و استفاده از ترمز

پارامتر دیگری که می‌تواند در مقایسه کنترل‌کننده‌های ماشین خودران مورد استفاده قرار بگیرد میزان و نحوه استفاده از ترمز است.

جدول ۶. مسافت طی شده کنترل‌کننده‌ها بر حسب متر

Table 6. Traveling distance of the controllers (in terms of meters)

کنترل‌کننده	مسافت طی شده
تناسبی- انتگرال- مشتقگیر	۱۱۷۰
کنترل پیشبین مدل	۱۴۲۳
انتها- به- انتها	۱۰۹۳

جدول ۷. مشخصات پیکربندی آزمایش

Table 7. Experimental setup configuration

پردازنده	اینتل i7-7700
حافظه اصلی	۱۶ گیگابایت
دیسک سخت	۲۵۶ گیگابایت
پردازنده گرافیکی	اینتل ۳۶۰
سیستم‌عامل	لینوکس ابونتو ۱۸/۰۴

در جدول ۵ داده‌های مربوط به این کمیت آورده شده‌است. با توجه به اینکه مکانیزم کنترلی انتها- به- انتها رانندگی را از انسان آموزش می‌بیند رفتار بهینه‌ای در این خصوص از خود نشان می‌دهد وسعی می‌کند با کنترل میزان گاز خودرو را تحت کنترل داشته باشد در حالیکه کنترل پیشبین مدل به وفور از ترمز استفاده می‌کند. این موضوع نشان می‌دهد که کنترل پیشبین مدل سبب اعمال استهلاک بیشتر به خودرو می‌شود. همچنین استفاده متعدد از ترمز می‌تواند سبب احساس نامطلوب در سرنشینان خودرو شود. آستانه احساس نامطلوب در اثر ترمزگرفتن برای سرنشینان حداکثر شتاب m/s^2 است [۲۱]، که در جدول ۴ تعداد دفعات تخطی از این آستانه برای کنترل‌کننده‌ها آورده شده است. البته لازم به ذکر است که این رفتار می‌تواند با تغییر پارامترهای این کنترل‌کننده‌ها تاحدی اصلاح گردد و نیازمند سعی و خطای بسیار و تجربه فراوان است. واریانس استفاده از پدال گاز و ترمز که راننده بر عکس یکدیگر از آن استفاده می‌کند نشان‌دهنده تغییرات شتاب خودرو است در جدول ۶ تحت عنوان واریانس شتاب آورده شده‌است.

۴-۷- خطا

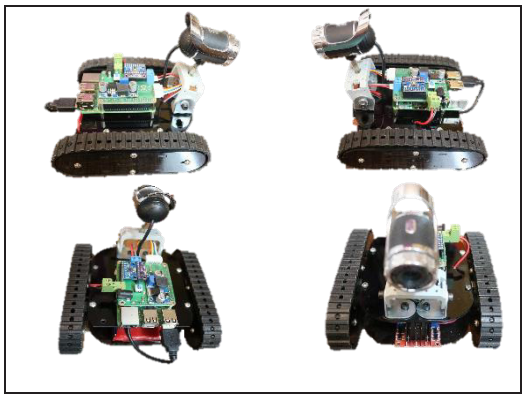
یکی از مهمترین پارامترهای برای انتخاب کنترل‌کننده بهتر درنظرگرفتن خطا در تصمیم‌گیری در مقایسه با رانندگی انسان است این تصمیم‌گیری شامل زاویه فرمان خودرو، میزان گاز و میزان ترمز است. برای مقایسه خطای کنترل‌کننده‌ها از رابطه (۶) استفاده شده‌است. که در آن میانگین مربعات اختلاف مقادیر اندازه‌گیری شده و مقدار مرجع که مربوط به رانندگی انسان است محاسبه شده است. نتایج این مقایسه در جدول ۵ آورده شده‌است.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \qquad (۶)$$

همانطور که مشاهده می‌شود، کنترل‌کننده انتها- به- انتها در مقایسه با دو کنترلر دیگر از نقطه نظر میزان ترمز و گاز عملکرد بهتری دارد. اما از نظر زاویه فرمان و سرعت در جایگاه دوم و به ترتیب پس از کنترل پیشبین مدل و تناسبی- انتگرال- مشتقگیر قرار گفته است و می بایست عملکرد آن در این زمینه بهبود یابد.

۴-۸- ایمنی

اگرچه برای ایمنی می‌توان تعاریف متعددی ارائه نمود، در این تحقیق با توجه به محیط شبیه‌سازی و همچنین صورت مسئله یعنی



شکل۹. ماشین خودران مقیاس کوچک از زوایای مختلف

Fig 9. Small scale unmanned vehicle used in this research (different perspectives)

شکل ۱۰. مسیر تعیین‌شده برای ماشین خودران

Fig 10. Our real testbed and the defined route for Small scale unmanned vehicle

دنبال‌کردن جاده، ملاک ایمن‌بودن رفتار کنترل‌کننده میزان فاصله از خطوط کنار جاده تعریف می‌شود. در آزمایش‌های انجام‌شده، خودرو تحت کنترل، در صورت استفاده از کنترل‌کننده‌های تناسبی- انتگرال- مشتقگیر و کنترل پیشبین مدل خطوط کناری جاده را در پیچ‌های تند قطع می‌کند درحالیکه در کنترل‌کننده انتها- به- انتها فاصله ایمن از خطوط کناری جاده را حفظ می‌کند.

۴-۹- مسافت

از آنجائیکه مسافت جاده‌ای که خودرو در محیط شبیه‌سازی می‌بایست طی کند برای هر سه کنترل‌کننده ثابت است، می‌توان میزان مسافتی که کیلومتر شمار خودرو نمایش می‌دهد را برای مقایسه کارآمدی کنترل‌کننده استفاده نمود. چرا که هر چقدر کنترل‌کننده مسافت کمتری طی کند، میزان سوخت و انرژی کمتری برای طی‌نمودن مسیر نیاز دارد. به صورت تقریبی اعداد بدست آمده از محیط شبیه‌سازی در جدول ۶ آورده شده‌است. همانطور که مشاهده می‌شود کنترل‌کننده انتها- به- انتها با توجه به انحراف کمتر از مرکز جاده مسافت کمتری را نسبت به دیگر کنترل‌کننده‌ها طی می‌کند.

۵- پیکربندی سیستم شبیه‌سازی

برای انجام آزمایش‌های این مقاله از محیط شبیه‌سازی متن باز مبتنی بر موتور بازی‌سازی یونیتی استفاده شده‌است . برای پیاده‌سازی کنترل‌کننده‌ها از زبان برنامه‌نویسی پایتون استفاده شده‌است. ساختار کنترل‌کننده انتها- به- انتها با استفاده از محیط تنسورفلو پیاده‌سازی و در کد اصلی به زبان پایتون مورد استفاده قرار

جدول ۸. وضعیت رانندگی کنترل‌کننده ETE

Table 8. Driving status of ETE controlle

سرعت (m/s)	زمان طی ۱ دور مسیر (ثانیه)	تماس با خطوط کناری	خارج‌شدن از خطوط کناری	خارج‌شدن از مسیر
۰/۴۶	۲۸	۵	۲	۱
۰/۳۹	۳۳	۳	۱	۰
۰/۲۹	۴۴	۲	۰	۰
۰/۲۱	۶۱	۱	۰	۰

گرفته است. سخت‌افزاری که کدها و محیط شبیه‌سازی بر روی آن اجرا شده‌است کامپیوتر رومیزی با مشخصات جدول ۷ است. لازم به ذکر است روش پیشنهادی بر روی کامپیوترهای با کارایی و قیمت پائین‌تر و حتی کامپیوترهای نهفته نیز قابلیت پیاده‌سازی و اجرا را دارد. اما ملاحظات زمان اجرا و حافظه مورد نیاز برای پارامترها، برای شبکه‌های بزرگتر از شبکه استفاده شده در این مقاله می‌بایست مورد توجه قرار بگیرد. کدهای پیاده‌سازی شده در این مقاله در [۲۲] قابل دسترسی است.

۶- پیاده‌سازی عملی

در راستای بررسی کارآمدی عملی روش انتها- به- انتها، این روش کنترلی بر روی ماشین خودران مقیاس کوچک ساخته‌شده در آزمایشگاه با ابعاد ۱۵در۱۵ سانتی متر، پیاده‌سازی و آزمایش شده‌است. همانطور که در شکل ۹ قابل مشاهده است، این ماشین خودران از اجزا ساده و کم هزینه شامل کفی، جعبه دنده و پیچ و مهره و موتور جریان مستقیم[1] کوچک تشکیل شده‌است.

در شکل علاوه بر اجزای مکانیکی، اجزای سخت‌افزاری این بستر آزمایش که شامل برد پردازشی رزپبری پای ۲، دوربین، برد واسط، برد تغذیه و شارژ باتری و همچنین راه‌انداز موتور جریان مستقیم هستند، مشخص است. شبکه عصبی عمیق معرفی‌شده در شکل ۴، با استفاده از ابزار یادگیری عمیق تنسورفلو نسخه ۲ پیاده‌سازی، آموزش و اجرا شده‌است. لازم به ذکر است مراحل پیاده‌سازی و فاز آموزش بر روی کامپیوتر با مشخصات جدول ۷ و براساس روش بیان‌شده در شکل ۶ صورت پذیرفته ، اما فاز اجرا بر روی برد پردازشی انجام شده‌است.

1 Direct Current (DC)

نتایج به‌دست‌آمده از این آزمایش نشان می‌دهد که کنترل‌کننده انتها- به- انتها قابلیت یادگیری رانندگی انسان و رانندگی بدون دخالت انسان را داراست. سیکل زمانی حلقه کنترل در روش استفاده‌شده به طور میانگین ۴۰ میلی ثانیه است. این زمان شامل دریافت تصاویر از دوربین، اجرای شبکه عصبی عمیق کنترلی بر روی برد پردازشی و ارسال دستورها به راه‌انداز موتورهاست. برای بررسی عملکرد انتها- به- انتها طی اجراهای مختلف پارامترهای تماس با خطوط کناری مسیر تعیین شده که در شکل ۱۰ مشخص است، خارج شدن از خطوط کناری و همچنین خارج‌شدن از مسیر (در این حالت با کنترل دستی ماشین به مسیر بازگردانده می‌شود) در سعی‌های مختلف مورد بررسی قرار گرفته است. جدول ۸ وضعیت کلی رانندگی کنترل‌کننده انتها- به- انتها را در ۲ بار طی‌شدن مسیر نشان می‌دهد. براساس این نتایج عملکرد کنترلی کنترل‌کننده انتها- به- انتها با کاهش سرعت بهبود می‌یابد. دلیل این امر ایده‌آل‌نبودن سیستم مکانیکی ماشین و لغزندگی و همچنین نویز و اعوجاج بیشتر تصاویر دوربین در سرعت‌های بالاتر است. پیوند [۲۳] حاوی فیلم حرکت این ماشین است.

۷- نتیجه‌گیری و جمع‌بندی

در این مقاله سعی شد از نقطه نظرهای مختلف استفاده از شبکه‌های عصبی عمیق انتها- به- انتها در کنترل ماشین خودران با کنترل‌کننده‌های کلاسیک پرکاربرد تناسبی- انتگرال- مشتقگیر و کنترل پیشبین مدل مقایسه گردد. این مقایسه می‌تواند در انتخاب کنترل‌کننده مناسب توسط طراحان مورد استفاده قرار گیرد. اگر چه به طور کلی کنترل‌کننده کنترل پیشبین مدل رفتار بهتری به

نسبت کنترل‌کننده انتها- به- انتها از خود نشان می‌دهد اما پیچیدگی طراحی و پیاده‌سازی کنترل‌کننده‌های کنترل پیشبین مدل می‌تواند دلیل دیگری بر استقبال از کنترل‌کننده‌های انتها- به- انتها باشد. به علاوه وابستگی کنترل‌کننده کنترل پیشبین مدل به مدل دینامیکی دقیق از سیستم و همچنین وابستگی شدید عملکرد و همچنین بار محاسباتی این کنترل‌کننده به پارامترهای کنترل‌کننده چالش مهمی است که می‌بایست در نظر گرفته شود. دقت بالاتر و همچنین قابلیت یادگیری انتها- به- انتها یکی دیگر از عوامل محبوبیت این کنترل‌کننده‌ها است. این نکته حائز اهمیت است که با توجه همزمان به مقادیر واریانس شتاب، زاویه و سرعت خودرو برای ۳ کنترل‌کننده بررسی شده، می‌توان نتیجه گرفت که تلاش کنترلی کنترل پیشبین مدل بسیار بالاتر از بقیه است و کنترل‌کننده انتها-به-انتها با اختلاف بسیار در رتبه دوم قرار دارد.

در این مقاله با استفاده از محیط شبیه‌سازی، کنترل‌کننده انتها- به- انتها بر روی ماشین خودران بدون دراختیارداشتن مدل خودرو و همچنین محیط پیرامون با موفقیت و بدون خطا و خروج از خطوط کناری جاده آزمایش شده‌است. لازم به ذکر است در این مقاله از شبکه عصبی عمیق کانولوشنی به صورت مستقیم و از داده‌های تصویر دوربین به عنوان ورودی آن و بدون استفاده از هیچ ماجول دیگری برای کنترل خودرو استفاده شده‌است. این در حالیست که کنترلر کنترل پیشبین مدل به سنسورهای سرعت، زاویه و همچنین الگوریتم تشخیص خط نیز نیازمند است.

از آنجائیکه برای مقایسه از الگوریتم‌های پایه هر سه کنترل‌کننده استفاده شده‌است در کارهای آتی قصد داریم تا از الگوریتم‌های بهینه‌سازی‌شده کنترل‌کننده تناسبی- انتگرال- مشتقگیر و کنترل پیشبین مدل و همچنین ساختارهای دیگر شبکه عصبی عمیق مانند حافظه طولانی کوتاه- مدت[1] و ماجول‌های کمکی دیگر و روش‌های پیاده‌سازی سخت‌افزاری بهینه که می‌تواند در بهبود عملکرد این کنترل‌کننده‌ها تاثیر بسزایی داشته باشد استفاده و مقایسه نمائیم. رویکرد دیگر برای کارهای آتی استفاده از روش‌های ترکیبی برای بهره‌گرفتن از نقاط قوت کنترل‌کننده کنترل پیشبین مدل در کنار کنترل‌کننده انتها- به- انتها است.

1 Long Short-Term Memory (LSTM)

مراجع

[1] S.K. Khaitan, J.D. McCalley, Design techniques and applications of cyberphysical systems: A survey, IEEE Systems Journal, 365-350 (2014) (2)9.

[2] M. Jirgl, Z. Bradac, P. Fiedler, Human-in-the-loop issue in context of the cyber-physical systems, IFAC-PapersOnLine, 230-225 (2018) (6)51.

[3] I. Lenz, R.A. Knepper, A. Saxena, DeepMPC: Learning deep latent features for model predictive control, in: Robotics: Science and Systems, Rome, Italy, 2015.

[4] Q. Zhu, T. Başar, Robust and resilient control design for cyber-physical systems with an application to power systems, in: 50 2011th IEEE Conference on Decision and Control and European Control Conference, IEEE, 2011, pp. 4071-4066.

[5] P.A. Ioannou, J. Sun, Robust adaptive control, Courier Corporation, 2012.

[6] J.-X. Xu, Y. Tan, Linear and nonlinear iterative learning control, Springer, 2003.

[7] S.M. Grigorescu, B. Trasnea, L. Marina, A. Vasilcoi, T. Cocias, Neurotrajectory: a neuroevolutionary approach to local state trajectory learning for autonomous vehicles, IEEE Robotics and Automation Letters, -3441 (2019) (4)4 3448.

[8] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F.E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing, 26-11 (2017) 234.

[9] A. Zappone, M. Di Renzo, M. Debbah, Wireless networks design in the era of deep learning: Model-based, AI-based, or both?, IEEE Transactions on Communications, (10)67 7376-7331 (2019).

[10] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural computation, (7)18 1554-1527 (2006).

[11] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G.V. Hernandez, L. Krpalkova, D. Riordan, J. Walsh, Deep learning vs. traditional computer vision, in: Science and Information Conference, Springer, 2019, pp. 144-128.

[12] S. Levine, C. Finn, T. Darrell, P. Abbeel, End-to-end

training of deep visuomotor policies, The Journal of Machine Learning Research, 1373-1334 (2016) (1)17.

[13] R. Raina, A. Madhavan, A.Y. Ng, Large-scale deep unsupervised learning using graphics processors, in: Proceedings of the 26th annual international conference on machine learning, 2009, pp. 880-873.

[14] D.A. Pomerleau, Alvinn: An autonomous land vehicle in a neural network, in: Advances in neural information processing systems, 1989, pp. 313-305.

[15] U. Muller, J. Ben, E. Cosatto, B. Flepp, Y.L. Cun, Off-road obstacle avoidance through end-to-end learning, in: Advances in neural information processing systems, 2006, pp. 746-739.

[16] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, U. Muller, K. Zieba, Visualbackprop: efficient visualization of cnns, arXiv preprint arXiv:,1611.05418 2016)).

[17] M. Bojarski, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, U. Muller, K. Zieba, Visualbackprop: visualizing cnns for autonomous driving, arXiv preprint arXiv:2016) 2 ,1611.05418).

[18] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L.D. Jackel, M. Monfort, U. Muller, J. Zhang, End to end learning for self-driving cars, arXiv preprint arXiv:2016) ,1604.07316).

[19] Udacity, Public driving dataset, in, https://www.udacity.com/self-driving-car, 2020.

[20] J. Kong, M. Pfeiffer, G. Schildbach, F. Borrelli, Kinematic and dynamic vehicle models for autonomous driving control design, in: 2015 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2015, pp. 1099-1094.

[21] Z. Wu, Y. Liu, G. Pan, A smart car control model for brake comfort based on car following, IEEE transactions on intelligent transportation systems, 46-42 (2008) (1)10.

[22] A. Mohammadi, Code repository, in, Github, https://github.com/abbasmohammadi/ETE, 2020.

[23] A. Mohammadi, Short video of our self-driving car, in, https://b2n.ir/2019 ,738589.

چگونه به این مقاله ارجاع دهیم

A. Mohammadi, K. Jamshidi, H. Shahbazi, M. Rezaei, A comparison study of deep neural controllers and classic controllers in self-driving car application, AmirKabir J. Mech. Eng., 53(Special Issue 4) (2021) 2439-2458.

DOI: 10.22060/mej.2020.18008.6708

این صفحه عمداً خالی است