A Comparison Study of Deep Neural Controllers and Classic Controllers in Self-Driving Car Application

Abbas Mohammadi[1], Kamal Jamshidi[2*], Hamed Shahbazi[3], Mehran Rezaei[4]

[1] PhD Student, University of Isfahan/Department of Computer Engineering

[2] Associate Professor, University of Isfahan/Department of Computer Engineering

[3] Assistant Professor, University of Isfahan/Department of Mechanical Engineering

[4] Assistant Professor, University of Isfahan/Department of Computer Engineering

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 presenting 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 compare with proportional–integral–derivative and model predictive controllers.

KEYWORDS

Cyber Physical Systems, Human-in-the-Loop System, Deep Learning, End-to-End Control, Self-Driving Car

[*] Corresponding Author: Email: jamshidi@eng.ui.ac.ir

1. Introduction

Nowadays, presence of cyber physical systems in modern devices especially safety-critical applications, draws great attention from researchers. These 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].

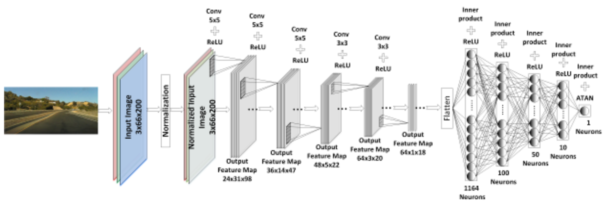
Human-in-the-loop control systems (HITL) can be defined as cyber physical systems with dynamic and unpredictable intervention of human in control loop [2]. The HITL control concept is known as a solution in challenging task such as autonomous vehicles. Especially in autonomous cars, efficient design and implementation of such controllers is the focus of many researches. There are a lot contemporary research devoted to solve 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 out work in section4.

2. Methodology

The main issue in design and implementation of ETE controller is to choose efficient and suitable architecture of 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 **Error! Reference source not found.**, the architecture constitutes of 9 convolution and fully-connected layers. The network directly maps the input image to angle of wheel as



output.

Figure 1: Deep Neural Network structure in our proposed method

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 shown in Figure 2. Obviously, the method can be divided in 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.

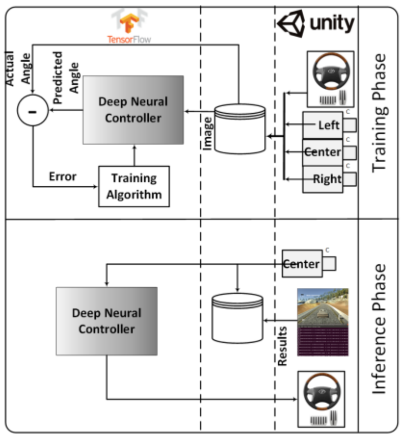


Figure 2: Proposed Method for ETE training and inference

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

Table 1: Training Configuration

Optimization	Adam
Initial Learning Rate	1e-4
Dropout Rate	0.5
Maximum Epoch	2000

Batch Size	100

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 **Error! Reference source not found.**. 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.

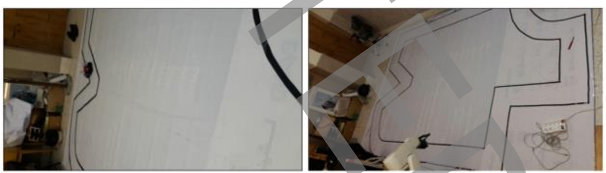


Figure 3: Our Real Testbed

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

Error is another comparison factor which 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 border of the road. Based on the definition the ETE controller is safe while the rest has crossed the border line of road.

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

4. Conclusion

In this paper, we aim at providing comprehensive evaluation and comparison among ETE controller and classic PID and MPC controllers for self-driving car in both simulation and real environment. Our experimental results can be used as 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 optimize version of the controllers to have more advance comparison. Furthermore, different deep architecture e.g. deep LSTM network and also the hybrid methods will be considered as future works.

5. 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).