

Road To Reliability: Optimizing Self-Driving Consistency With Real-Time Speed Data

William Fowler
Tufts University
Medford, MA, USA
william.fowler@tufts.edu

Kate Keahey [advisor]
University of Chicago
Chicago, IL, USA
keahey@mcs.anl.gov

Alicia Esquivel Morel [advisor]
University of Missouri
Columbia, MO, USA
ace6qv@mail.missouri.edu

Abstract

Self-driving cars can potentially improve transportation efficiency and reduce human fatalities – provided they have access to significant processing power and large amounts of data. A popular approach to the actualization of autonomous vehicles is to use end-to-end learning [1], in which a machine learning model is trained on a large data set of real human driving. This poster shows how self-driving consistency can be improved using a Convolutional Neural Network (CNN) to predict current velocity. Our approach first reproduces an end-to-end learning result and then extends it with real-time speed data as additional model input.

CCS Concepts: • Computer systems organization → Robotic autonomy.

Keywords: Autonomous driving, optimization, machine learning

ACM Reference Format:

William Fowler, Kate Keahey [advisor], and Alicia Esquivel Morel [advisor]. 2023. Road To Reliability: Optimizing Self-Driving Consistency With Real-Time Speed Data. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (SC '23)*. ACM, New York, NY, USA, 2 pages. <https://doi.org/XXXXXX.XXXXXX>

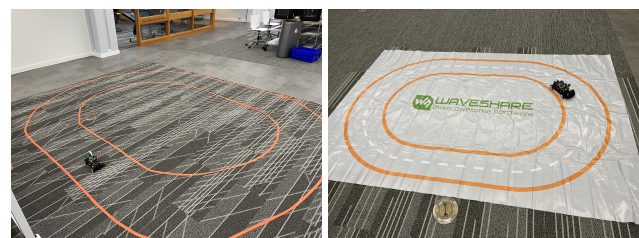
1 Introduction

Self-driving cars are expected to perform at high speed – therefore, fast inference time is necessary to react in time to avoid accidents. A current limitation of end-to-end learning is that models are often trained to only predict steering while maintaining constant velocity. This limitation can be avoided by using an optical encoder sensor tracking wheel revolutions to observe car speed and then controlling speed

by regulating throttle and power sent to the motor. However, a speed-predicting end-to-end model has only been tested in simulation [4]. The Donkey Car self-driving project [2] uses a CNN to predict steering and throttle to drive a scale model car using human-collected data. Predicting a throttle percentage is problematic, since the power sent to the motor will have a varying effect on the speed of the car depending on the road environment and the capabilities of the motor. We hypothesize that using end-to-end learning for current speed prediction will result in fewer errors and increased accuracy in varying surface conditions.

2 Methodology

Bojarski, et al. [1] showed that a CNN trained on 72 hours of real driving data could run in a real-road setting with 98% precision. We replicated this result using a 1:18 scale car equipped with a Raspberry Pi 4 as its onboard computer and collecting data by driving the car around an oval track. We achieved similar results to Bojarski, et al. Then, to extend these experiments to achieve fully autonomous driving on multiple surface environments, we equipped the car with an optical encoder sensor to read current speed.



(a) Default track

(b) Waveshare track

Figure 1. Two different oval tracks were utilized for the experiments.

Furthermore, to measure performance on different road surfaces, we created two tracks: the default track (see Figure 1a) consists of two lines of orange tape on carpet flooring, and the Waveshare track (see Figure 1b) which is an orange-lined track printed on a mat laid out on the floor. Our proposed solution to the problem of autonomous speed control is the **velocity model**, which uses the same CNN structure as the default model packaged with the Donkey Car (the Linear model), but is trained to predict the current speed of the car

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SC '23, Nov 12–17, 2023, Denver, CO

© 2023 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/XXXXXX.XXXXXX>

based on the image input, while the default Linear model predicts throttle percentage

By ensuring that the models are built using the same CNN structure, this controls for model complexity. The models are also trained on the same dataset.

3 Evaluation

We tested the models on both tracks (Figure 1). The car was allowed to drive autonomously for 10 minutes at a time for 5 trials. An error is defined as both front wheels of the car leaving the track or the car stopping for more than 5 seconds. Successful laps are full laps around the track completed with zero errors. The accuracy of the models was calculated using an autonomy score:

$$\text{Autonomy Score} = 1 - \frac{\text{Number of Errors}}{\text{Number of Laps}} \quad (1)$$

Equation 1 outputs an autonomy score that shows how well a model could complete a full lap without error. The autonomy value was at or near 1.0 for both models on the default track (Figure 2a). On the other hand, in the Waveshare track, Velocity averaged 473% higher autonomy than Linear (Figure 2b). On the default track, the models made little or no errors (Figure 3a). On the Waveshare track, Linear made 7.27× as many errors as Velocity (Figure 3b).

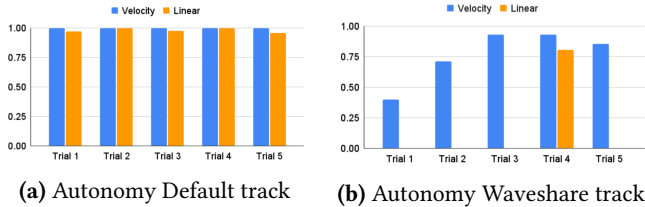


Figure 2. Results show few or no errors on the default track, whereas on the Waveshare track, the Linear model made 7.27× as many errors as Velocity. Autonomy is a percentage score showing for what percentage of a lap the car drove on the road.

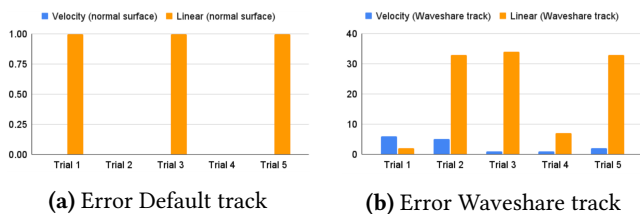


Figure 3. Errors made on default and Waveshare tracks respectively in a 10 minutes period.

On the default track, successful laps using Linear were 2.11× quicker than Velocity (Figure 4a). On the Waveshare track, this ratio was 2.85 (Figure 4b).

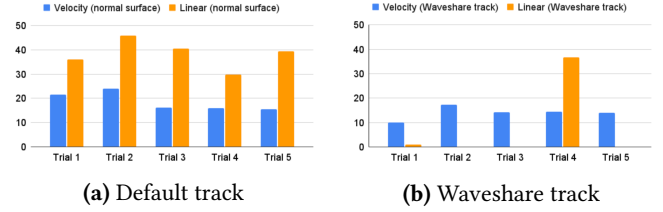


Figure 4. Successful laps completed on default and Waveshare tracks respectively in a 10 minutes period. This plot gives insight into which track allowed for faster laps.

4 Conclusions

In order to ensure safe autonomous driving, neural networks must be able to properly control speed. We presented a comparison of two different methods for controlling the steering and speed of a scale-model car with a CNN. Our proposed solution, which predicts the current speed of the car, produced 7.47× fewer errors than a model of the same structure which predicts throttle percentage. These results show that a velocity-dependent model is better suited for safety in autonomous driving. Future work would require examining the trade-offs between model accuracy and car speed [3].

References

- [1] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316* (2016).
- [2] Tawm Kramer. 2016. Donkey Car. <https://github.com/autorope/donkeycar>.
- [3] Xiaopeng Li. 2022. Trade-off between safety, mobility and stability in automated vehicle following control: An analytical method. *Transportation research part B: methodological* 166 (2022), 1–18.
- [4] Shakti N Wadekar, Benjamin J Schwartz, Shyam S Kannan, Manuel Mar, Rohan Kumar Manna, Vishnu Chellapandi, Daniel J Gonzalez, and Aly El Gamal. 2021. Towards end-to-end deep learning for autonomous racing: On data collection and a unified architecture for steering and throttle prediction. *arXiv preprint arXiv:2105.01799* (2021).