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

William Fowler, Kate Keahey (advisor), Alicia Esquivel Morel (advisor)

Tufts University, The University of Chicago | Argonne National Laboratory, University of Missouri - Columbia

Data Collection

Chi@Edge Autonomous Cars

Rationale

Why is Autonomous Driving Important?

✓ Self-driving cars can reduce human fatalities
✓ Accurate driving requires fast computation on edge devices
✓ Worst-case performance must be safe

Why does Self-Driving need HPC?

✓ For on-road driving, extremely large data sets required
✓ Fast computation needed for accurate inferencing

Reproducing End-to-end Principles
All data collected from human driving and Training a CNN [1]

Pi-Racer:
Raspberry Pi 4 & Camera

NVIDIA

✓ Scale-model car
✓ Raspberry Pi instead of Nvidia Drive PX
✓ 30 mins data instead of 72 hours
✓ Testing on indoor, oval track and smaller inference model

Donkey Car:

✓ Open-source project for self-driving RC cars
✓ Utilizes Tensorflow

End-to-End Outcomes

✓ Education/Reproducibility: Extensive documentation of project
✓ **Trovi:** reproducible artifact
✓ Educational module for classrooms
✓ Testbed for Improvements: End-to-end compatibility with additional sensors
✓ Connection to Chameleon's Chi@Edge
✓ Low cost; fast experimentation

Research Question

How can real-time speed data improve self-driving consistency?

✓ Problem: Comparing models is hard because of inconsistent speed
✓ Different battery life = different speed
✓ Performance varies on different surfaces

Proposed Solution

KerasVelocity Model

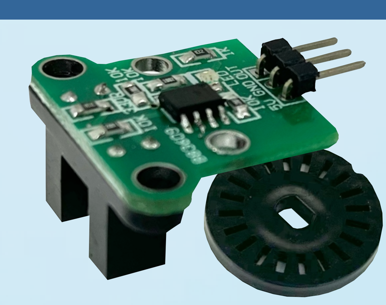
✓ Collect data with encoder speed
✓ Model predicts velocity instead of throttle

KerasLinear Model

✓ Model predicts throttle percentage based solely on image input

Expected Results:

✓ Model performance should be independent of battery percentage
✓ Improved performance on different surfaces compared to control
✓ Model should learn to be more cautious: less errors



HC-020K Speed Measuring Sensor Module

```
"enc/speed:" 0.28349513655905856
```

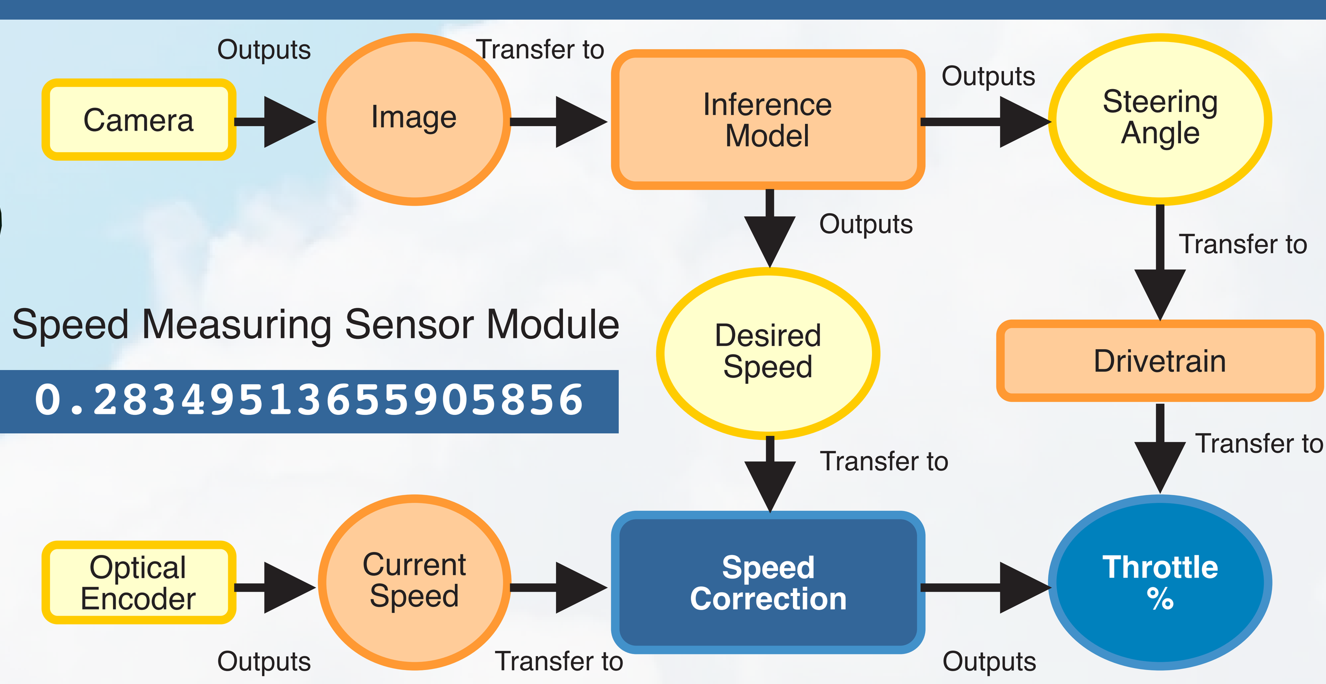
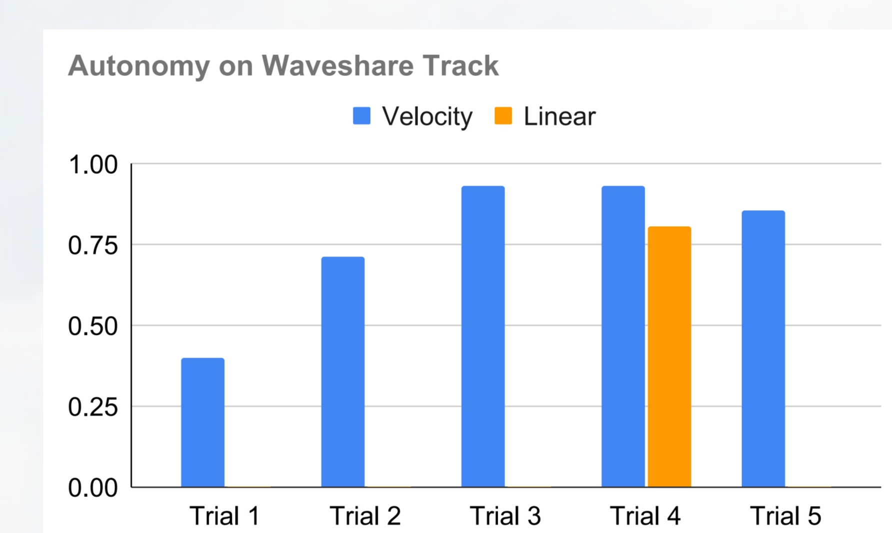
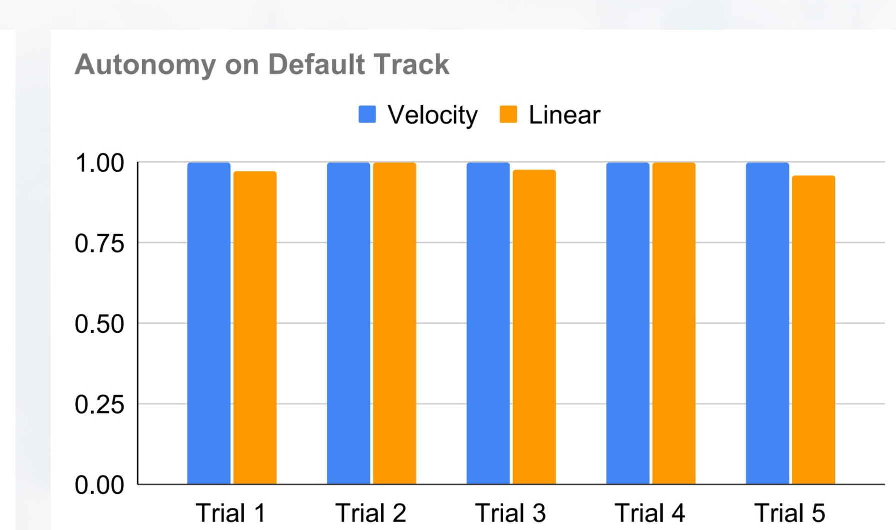


Fig. 1: Solution Architecture, I/O structure for Velocity model

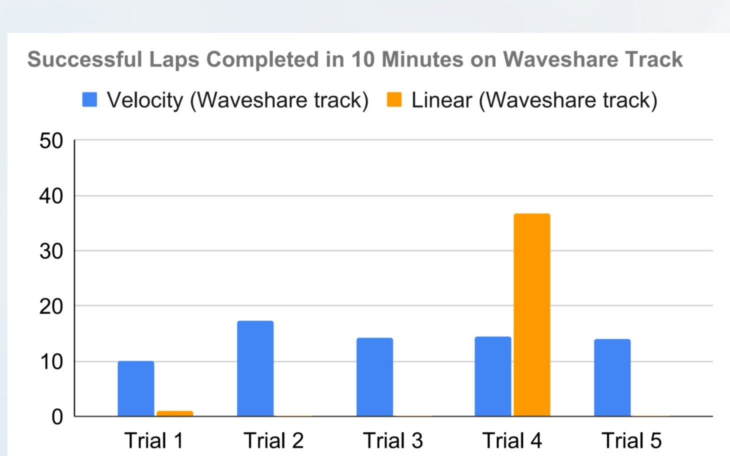
Autonomy Score: 1 - (number of errors / number of laps)

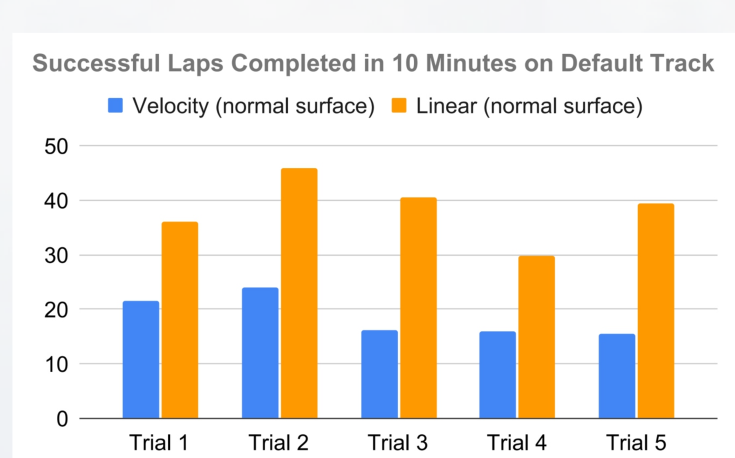




Autonomy shows how well models could complete laps without errors. If no successful laps completed or if score below 0, autonomy considered to be 0

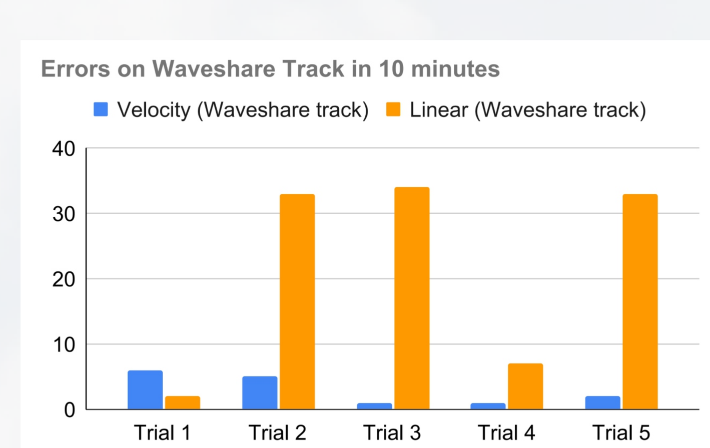
Completed Laps

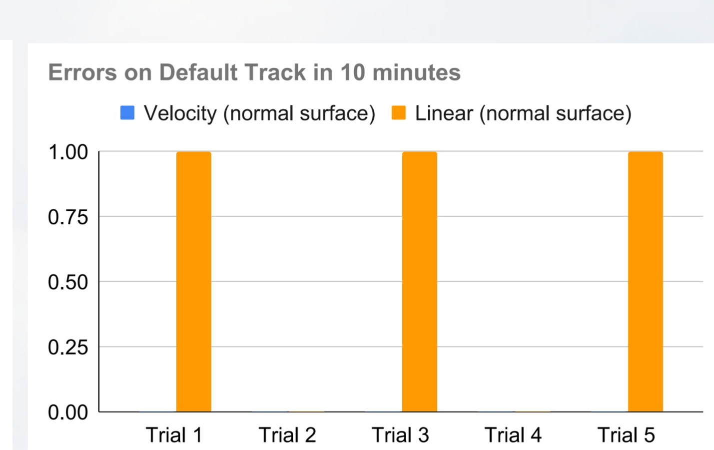




Successful laps are full laps w/o error. Linear model made 0 successful laps on the Waveshare track in trials 2, 3, and 5

Errors





Errors defined as both wheels completely off the track or stuck for over 5 seconds. Velocity model made no errors on the normal track

Conclusion and Future Work

Default Track:

✓ Velocity autonomy marginally better than Linear **(1.6%)**
✓ Velocity makes marginally fewer errors **(0 vs. 3)**
✓ During successful laps, Linear drives **2.11 times faster** than Velocity

Waveshare Track:

✓ Velocity autonomy **473% better** than Linear
✓ Velocity makes **7.27 times fewer errors** than Linear
✓ During successful laps, **Linear drives 2.85 times faster** than Velocity

✓ These results show that a velocity-dependent model is better suited for safety in autonomous driving

✓ **As Future Work**, more accurate encoder, testing on more surfaces. Calculate the trade offs between speed and accuracy

[1] Bojarski, Mariusz, et al. "End to end learning for self-driving cars." arXiv preprint arXiv:1604.07316 (2016)