Fakultät für Informatik
Lehrstuhl für Echtzeitsysteme und Robotik

Scene-aware and Social-aware Motion Prediction for Autonomous Driving

Alfred Nguyen

Practical Course *Motion Planning for Autonomous Vehicles* WS 2023/2024

Advisor:	Dr. Di Liu
Supervisor:	Prof. Dr.-Ing. Matthias Althoff
Submission:	20. February 2024

Scene-aware and Social-aware Motion Prediction for Autonomous Driving

Alfred Nguyen
Technische Universität München
Email: alfred.nguyen@tum.de

***Abstract*—The abstract goes here.**

I. Problem Statement

In our previous approach, we relied on a method known as ballistic integration. However, a significant challenge arises due to the inherent discretization error associated with working with discrete data. Specifically, when we rearrange the integration formula to assess its fit with our data,

we observe errors between the predicted acceleration and the ground truth. These errors will furthermore be explored in the results section.

The differences in the accelerations affect the performance of the neural network, as the integration results are used as inputs to the network.

II. Concept Overview

As we are working with a discrete dataset, there is no way to integrate or derive over a continuous function which describes the motion of a car.

Thus, a discrete integration method to describe the motion of a car is needed. In previous attempts, other discrete integration methods were tested such as the ballistic integration model,

$$s(k+1) = s(k) + dt \cdot v(k) + \frac{dt^2}{2} a(k) \tag{1}$$
$$v(k+1) = v(k) + dt \cdot a(k) \tag{2}$$

where *dt* describes the time interval between each data point. The main problem here was, that this integration model had issues with the accuracy of the acceleration. After rearranging both formulas to the acceleration parameter

$$a(k) = \frac{2}{dt^2}\Big(s(k+1) - s(k) - dt \cdot v(k)\Big) \tag{3}$$
$$a(k) = \frac{1}{dt}\Big(v(k+1) - v(k)\Big) \tag{4}$$

the resulting formulas do not calculate the same acceleration. Detailed results will be shown in the results section.

Our method revolves around the use of a linear model to estimate the acceleration of vehicles. By equating and rearranging the formula, we specifically ensure that the predicted accelerations remain consistent for both equations. This will be further elaborated in the methods section.

The motivation behind choosing a linear model stems from the hope of better performance compared to alternative models tested during our experimentation. Through rigorous testing, we found that the selected linear model consistently outperformed others in terms of accuracy in the equality of both accelerations from both formulas. Additionally, the simplicity and interpretability of the linear model make it a good choice for motion prediction tasks, as we can comfortably use established methods to solve these systems.

To validate the effectiveness of our approach, we plan to compare the results obtained with our linear model against the old Ballistic Integration method, particularly focusing on the accuracy and consistency of the predicted accelerations. Additionally, we will evaluate the performance of our model using standard linear regression metrics such as R-value and MSE. Furthermore, we aim to extend our analysis by rearranging the model to predict velocity and distance, allowing us to assess its predictive capabilities. In the end, we will visualize our results using the existing drone-dataset-tool repo, which was provided.

III. Method

In this section, we detail the methodology employed to determine the optimized integration model along with the implementation steps undertaken to validate our approach.

A. Dataset Description

For our experimentation, we utilized the inD, exiD, and rounD datasets provided by the ika of RWTH Aachen University. The datasets offer vehicle trajectories recorded at German intersections, highway exits, and entries, and roundabouts, respectively. Additionally, the datasets were provided with a tool that allowed us to visualize them for better understanding.

B. Model Selection Process

Our model selection process involved a systematic trial and error approach with a total of 8 different models. Each model underwent an evaluation process to assess its ability to accurately predict vehicle acceleration across various scenarios. Ultimately, the following linear model emerged as the most suitable choice based on its superior performance metrics.

$$a(k) = \bar{c}_1\big(s(k+1) - s(k) - v(k)\big) - \bar{c}_2 a(k-1) \tag{5}$$
$$a(k) = \bar{c}_3\big(v(k+1) - v(k)\big) - \bar{c}_4 a(k-1) \tag{6}$$

This linear model is then solved by linear regression. After training the model on the acceleration set from our dataset,

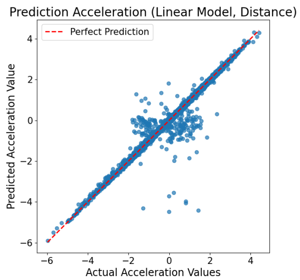

Fig. 1. Caption for Figure 1

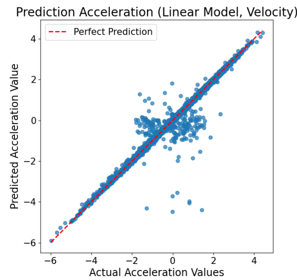

Fig. 2. Caption for Figure 2

Fig. 3. Caption for the entire figure

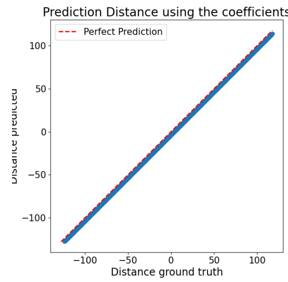

Fig. 4. Caption for Figure 1

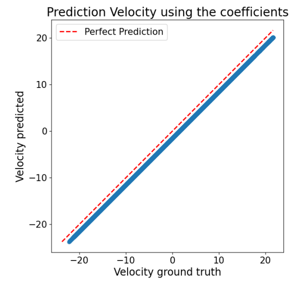

Fig. 5. Caption for Figure 2

Fig. 6. Caption for the entire figure

we can rearrange the formulas to determine the distance and velocity formulas

$$s(k+1) = s(k) + v(k) + c_1 a(k) + c_2 a(k-1) \tag{7}$$

$$v(k+1) = v(k) + c_3 a(k) + c_4 a(k-1) \tag{8}$$

The constants can then be determined through these calculations:

$$c_1 = \frac{1}{\bar{c}_1} \tag{9}$$

$$c_2 = \bar{c}_2 \cdot c_1 \tag{10}$$

$$c_3 = \frac{1}{\bar{c}_3} \tag{11}$$

$$c_4 = \bar{c}_4 \cdot c_3 \tag{12}$$

By rearranging the model as such, we specifically ensure that for both $s(k)$ and $v(k)$ we receive the same acceleration, as we are training both models on the same acceleration set. Thus, we receive a proper integration method for the data set on which we trained the model, which solves the problems of the mismatched acceleration in previous attempts.

C. Training and Testing Procedure

For training the model, we used the $LinearRegression()$ function, provided by the sci-kit-learn Python library to train our linear model. The $train\_test\_split()$ function was utilized, with a test size of 0.3, to ensure a sufficient amount of data for evaluation.

D. Evaluation Metrics and Results

The performance of our linear model was assessed using common metrics for linear regression, including R-squared (r2) and Mean Squared Error (MSE). Detailed evaluation results, including comparisons with the old method, will be provided later in the report, offering insights into the model's predictive capabilities.

IV. Results

1) Results - Linear Model: Subsubsection text here.

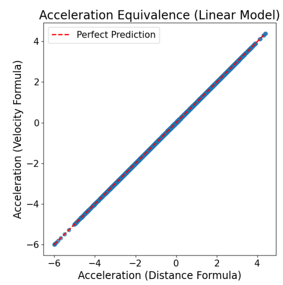

Fig. 7. Caption for Figure 1

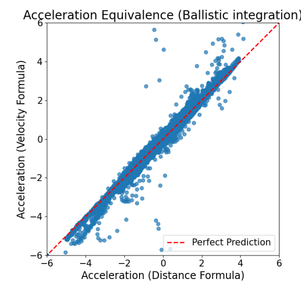

Fig. 8. Caption for Figure 2

Fig. 9. Caption for the entire figure

A. Notable Findings and Observations

During the evaluation process, several notable findings were observed. Most notably, after rearranging the formula for distance and velocity, a constant error was detected. However, increasing the size of NumPy arrays appeared to mitigate this error, indicating a potential floating-point precision issue.

B. Future Directions

Based on our implementation experience, future research and improvement areas include enhancing the precision of NumPy arrays by increasing their size and expanding the dataset to encompass a more extensive range of scenarios for comprehensive model training and testing.

V. Conclusion and Future Work

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

VI. Introduction

This demo file is intended to serve as a "starter file" for IEEE conference papers produced under LaTeX using IEEEtran.cls version 1.8 and later. I wish you the best of success.

mds

December 27, 2012

A. Subsection Heading Here

Subsection text here.

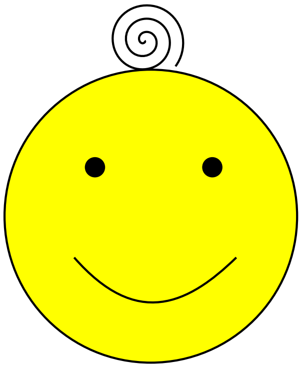

Fig. 10. A vector graphic loaded from a PDF file

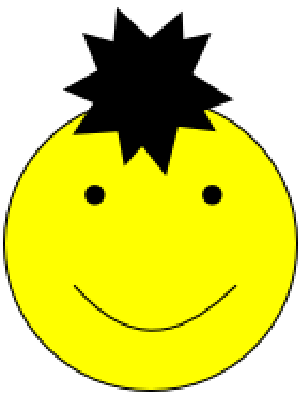

Fig. 11. A bitmap graphic loaded from a PNG file

1) Subsubsection Heading Here: Subsubsection text here.

VII. CONCLUSION

The conclusion goes here.