Fakultät für Informatik
Lehrstuhl für Echtzeitsysteme und Robotik

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

Alfred Nguyen, Baris Sozudogru

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, Baris Sozudogru
Technische Universität München
Email: alfred.nguyen@tum.de, baris.sozudogru@tum.de

***Abstract*—Self-driving technology holds the promise to make travel safer and more efficient. One of the most important steps to improving self-driving cars is to understand how humans behave using vehicles. As highlighted in the study by [1], understanding the principles and rules of dynamic interaction among human drivers in complex traffic scenes is essential for generating diverse social driving behaviours, predicting future states of a scene with moving objects, and creating realistic driving simulators. While some paths and trajectories are physically possible, some are socially unacceptable [2]. People inherently respect social norms such as yielding right-of-way or respecting personal space. This paper builds on a method to enhance the prediction of vehicles using a neural network to introduce social forces between vehicles. These social forces aim to mimic the social interaction between human-driven vehicles. In particular, we focus on improving the discrete integration method used in the neural network, reducing the error between prediction and ground truth acceleration.**

I. Problem Statement

In addressing the complexities of autonomous driving systems, our enhanced approach combines the refinement of vehicle behavior detection through filtering techniques with the critical evaluation of integration models, specifically addressing the challenges observed in ballistic integration. Previously, the discrete nature of ballistic integration led to discretization errors, impacting the accuracy of predicted accelerations when compared against ground truth data. This discrepancy not only undermined the performance of neural networks utilized in motion prediction but also highlighted the need for a methodological overhaul. By integrating a filtering module that detects nuanced vehicle behaviors—such as entering, merging, and yielding—with an improved integration model, we aim to mitigate the inaccuracies stemming from discretization errors, leveraging insights from [3], who underscore the complexity of human motion and the necessity of accounting for multimodal paths in crowded spaces. This dual-focused strategy ensures a balanced emphasis on both the detection of vehicle dynamics and the precision of mathematical models. Through this approach, we seek to address the limitations of previous methodologies by offering a more accurate, reliable, and comprehensive framework for understanding and predicting vehicle behaviour in complex driving scenarios. Supporting our methodology, [4] demonstrate the significance of considering social and environmental factors in predictive modeling, emphasizing the need for a comprehensive behavioral model that incorporates these elements for improved tracking performance.

Fig. 1. Socially Interacting Vehicles

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. Specifically, after rearranging both formulas to the acceleration parameter like such,

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

$$a(k) = \frac{1}{dt}\Big(v(k+1) - v(k)\Big) \tag{4}$$

the resulting formulas do not calculate the same acceleration.

Our method revolves around the use of a linear model to estimate the acceleration of vehicles. By equating and rearranging both formulas for the acceleration, we ensure that the predicted accelerations remain consistent for both equations. Through rigorous testing, we found that the selected linear model outperformed the ballistic integration model in terms of accuracy and the equality of both accelerations from both formulas.

To test our results we will be comparing our model to the previoulsy used ballistic integration method. Additionally, we will evaluate the performance of our model using standard

linear regression metrics such as mean square error (MSE), mean absolute error (MAE), and R-squared (R^2) score.

Furthermore, we aim to extend our analysis by rearranging the model to predict velocity and distance, allowing us to assess if our model can be used as a valid discrete integration model. In the end, we will visualize our results using the existing drone-dataset-tool repo, which was provided.

To complement this methodological foundation, our approach incorporates a filtering module, designed to accurately detect and analyse nuanced vehicle behaviours critical for motion prediction, such as merging, yielding, and hard braking. This module employs advanced data processing techniques to refine the discrete dataset, ensuring the identification of meaningful patterns and behaviours. The integration of this filtering module with our enhanced linear modelling technique establishes a comprehensive framework for motion prediction in autonomous driving systems. This synergy not only mitigates the limitations of previous integration methods but also improves the accuracy and reliability of predicting vehicle dynamics. Our approach, therefore, represents an advancement in the field of intelligent transportation systems, offering a deeper understanding of vehicle interactions and dynamics within complex urban environments. This integrated methodology underscores our commitment to developing innovative solutions that enhance the predictability, safety, and efficiency of autonomous driving technologies. As shown in Fig. 2, the general pipeline overview illustrates the approach taken in this study, integrating both the advanced filtering techniques and the linear modelling for a thorough analysis and prediction of vehicle behavior in discrete datasets.

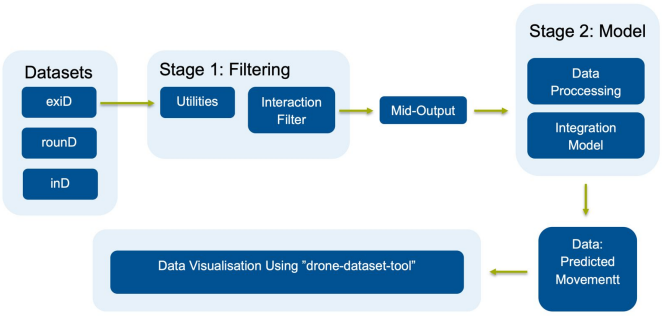

Fig. 2. Overview of the General Pipeline

III. Method

In this section, we detail the methodology used to formulate filtering module and determine the discrete integration model along with the implementation steps to validate our approach.

A. Dataset Description

For our experimentation, we utilized the inD [5], exiD [6], and rounD [7] datasets provided by the *Institut für Kraftfahrzeuge (ika) 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 (drone-dataset-tool) [8] that allowed us to visualize them for better understanding. The datasets include 18 columns, spanning from vehicle identifiers and lifetimes of vehicles to various columns of motion-related data points.

Our analysis primarily targets key data points: x_{Center}, y_{Center}, x_{Velocity}, y_{Velocity}, $x_{\text{Acceleration}}$, for the integration model. The filtering module, however, utilizes these columns alongside additional data, allowing for a detailed examination of vehicle dynamics and the identification of specific driving behaviors crucial for enhancing predictive accuracy.

The columns x_{Center} and y_{Center} represent the current position of the object's center in the coordinate coordinate system. Preprocessing was done such that the columns describe the distance to the origin of the vehicle.

Velocity information is captured through columns x_{Velocity} and y_{Velocity} representing velocities in the x-axis and y-axis respectively.

Acceleration data are represented in columns $x_{\text{Acceleration}}$ and $y_{\text{Acceleration}}$ which provide information on the acceleration in the x-axis and y-axis respectively.

For an illustrative example of traffic scenarios and vehicle behaviors captured in the exid and round datasets, see Fig. 3 and Fig. 4, respectively.

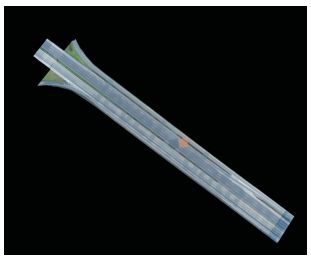

Fig. 3. exiD dataset

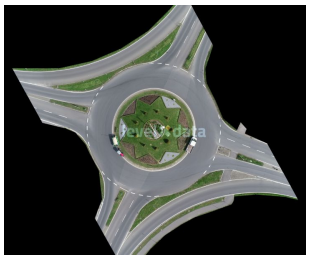

Fig. 4. rounD dataset

B. Filtering Module

In addition to the rigorous dataset description already outlined, our methodology incorporates a filtering module designed to refine the datasets further for enhanced predictive modeling. This module, integral to our approach, employs algorithms to process the raw data meticulously, identifying and extracting meaningful patterns indicative of specific vehicle behaviors. These behaviors, crucial for accurate motion prediction, include merging, yielding, and hard braking, among others.

The filtering module operates by first preprocessing the vehicle data to ensure consistency and reliability in the input data. This step involves normalizing the data points to a common reference frame, enabling a more accurate analysis of vehicle trajectories. For instance, the preprocessing involve a transformation formula for normalizing vehicle positions, such as:

$$\text{Normalized Position} = \frac{\text{Position} - \text{Min Position}}{\text{Max Position} - \text{Min Position}} \quad (5)$$

Following this, the module applies a series of algorithms aimed at detecting specific behaviors. For instance, the detection of entering vehicles is based on a set of predefined conditions such as interaction distance threshold, relative speed threshold, and yielding behavior, which are crucial for understanding vehicle dynamics at intersections and roundabouts. For detecting entering vehicles and other behaviours, a combination of velocity and acceleration thresholds might be applied, defined as:

$$\text{Entering Condition} = \begin{cases} \text{True} & \text{if } \Delta v > \text{speed threshold} \\ & \text{and } \Delta d < \text{distance threshold} \\ \text{False} & \text{otherwise} \end{cases} \tag{6}$$

Where Δv represents the change in velocity, and Δd signifies the relative distance to another vehicle.

The algorithms within the filtering module make extensive use of the vehicle trajectory data, focusing on parameters such as x_{Center}, y_{Center}, x_{Velocity}, y_{Velocity}, $x_{\text{Acceleration}}$, $y_{\text{Acceleration}}$. By analyzing these parameters over time, the module can detect changes in vehicle behavior that signal merging, yielding, or hard braking. This detection is further enhanced by comparing the mean of the first frames (indicative of vehicle entry) and the last frames (indicative of vehicle exit) against established thresholds, thereby identifying vehicles that are entering or exiting the roadway.

Furthermore, the module incorporates functions to detect lane changes, filter vehicles in the same lane, and assess the relative velocity and distance between vehicles to identify potential interactions. These functions are critical for understanding the complex dynamics of vehicle behavior in dense traffic conditions and are instrumental in improving the accuracy of our discrete integration model. For example, lane change detection involves analysing the lateral displacement and comparing it against a lane width parameter to determine if a vehicle has moved sufficiently across lanes:

$$\text{Lane Change Detected} = \begin{cases} \text{True} & \text{if } |\Delta x| > \frac{\text{Lane Width}}{2} \\ \text{False} & \text{otherwise} \end{cases} \tag{7}$$

The module's algorithms also utilize parameters directly from the datasets, such as odrLaneId for lane identification and laneChange flags, to facilitate precise behavioural analyses. These elements are crucial for the module's ability to detect driving behaviours like yielding and hard braking, through the calculation of relative velocities and distances, applying thresholds that have been determined to reflect driving scenarios.

By integrating this filtering module into our methodology, we enhance the quality of the dataset used for modeling. This not only improves the accuracy of our predictive models but also ensures that the models are trained on data that accurately reflect the complexities of real-world driving scenarios. The application of these advanced data processing techniques contributes to the overall robustness and reliability of our approach, setting a new approach for predictive modeling in autonomous driving systems.

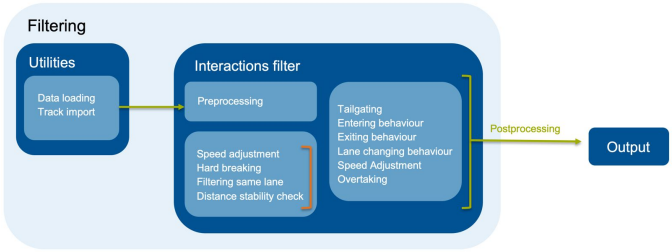

Fig. 5. Overview of the Filtering Module

The Fig. 5 provides a detailed visualization of the structure and functionality of the filtering module, comprising two principal components: Utilities and Interactions Filter. The Utilities package is tasked with the loading and configuration of the dataset, preparing it for analysis. The Interactions Filter, highlighted with an orange bracket, initially focuses on identifying micro behaviors such as speed adjustment and hard braking. It advances to recognize macro behaviors by synthesizing these micro observations, including vehicles' entering, exiting, and overtaking actions. Postprocessing refines the data further, preparing it for the integration module.

C. 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 evaluate its ability to accurately predict vehicle acceleration. Ultimately, the following two linear models emerged as the most suitable models

$$a_{dis}(k) = \bar{c}_1\big(s(k) - s(k+1) - v(k)\big) - \bar{c}_2 a(k-1) \tag{8}$$

$$a_{vel}(k) = \bar{c}_3\big(v(k) - v(k+1)\big) - \bar{c}_4 a(k-1) \tag{9}$$

These linear models are then solved by linear regression. After training the model on the acceleration set from our dataset, we can rearrange the formulas using the newly found coefficients $(\bar{c}_1, \bar{c}_2, \bar{c}_3, \bar{c}_4)$ to determine the distance and velocity formulas, similar to the ballistic integration

$$s(k+1) = s(k) + v(k) + c_1 a_{dis}(k) + c_2 a(k-1) \tag{10}$$

$$v(k+1) = v(k) + c_3 a_{vel}(k) + c_4 a(k-1) \tag{11}$$

The coefficients can then be determined through these calculations:

$$c_1 = \frac{1}{\bar{c}_1} \tag{12}$$

$$c_2 = \bar{c}_2 \cdot c_1 \tag{13}$$

$$c_3 = \frac{1}{\bar{c}_3} \tag{14}$$

$$c_4 = \bar{c}_4 \cdot c_3 \tag{15}$$

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 on, which solves the problems of the mismatched acceleration in previous attempts. As seen in the section III-A, we can see, that the columns for distance, velocity and acceleration are split into their x- and y-components. Our linear models will look like this:

Distance Model (Acceleration from distance formula):

$$\begin{bmatrix} a_x(k) \\ a_y(k) \end{bmatrix}_{\text{dis}} = \begin{bmatrix} s_x(k) - s_x(k+1) - v_x(k) & -a_x(k-1) \\ s_y(k) - s_y(k+1) - v_y(k) & -a_y(k-1) \end{bmatrix} \begin{bmatrix} \bar{c}_1 \\ \bar{c}_2 \end{bmatrix} \quad (16)$$

Velocity Model (Acceleration from velocity formula):

$$\begin{bmatrix} a_x(k) \\ a_y(k) \end{bmatrix}_{\text{vel}} = \begin{bmatrix} v_x(k) - v_x(k+1) & -a_x(k+1) \\ v_y(k) - v_y(k+1) & -a_y(k+1) \end{bmatrix} \begin{bmatrix} \bar{c}_3 \\ \bar{c}_4 \end{bmatrix} \quad (17)$$

Afterward, we will compare the accelerations derived from displacement, denoted as $\begin{bmatrix} a_x(k) \\ a_y(k) \end{bmatrix}_{dis}$, with those derived from velocity, denoted as $\begin{bmatrix} a_x(k) \\ a_y(k) \end{bmatrix}_{\text{vel}}$, to assess the effectiveness of the distance model's acceleration estimation relative to that of the velocity model.

Using the coefficients determined from linear regression from the linear models (16) and (17), we can determine the distance and velocity using the following matrices:

Distance formula

$$\begin{bmatrix} s_x(k+1) \\ s_y(k+1) \end{bmatrix} = \begin{bmatrix} a_x(k) & a_x(k-1) \\ a_y(k) & a_y(k-1) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} s_x(k) + v_x(k) \\ s_y(k) + v_y(k) \end{bmatrix} \quad (18)$$

Velocity formula

$$\begin{bmatrix} v_x(k+1) \\ v_y(k+1) \end{bmatrix} = \begin{bmatrix} a_x(k) & a_x(k-1) \\ a_y(k) & a_y(k-1) \end{bmatrix} \begin{bmatrix} c_3 \\ c_4 \end{bmatrix} + \begin{bmatrix} v_x(k) \\ v_y(k) \end{bmatrix} \quad (19)$$

Hence, we obtain a discrete integration model with matching accelerations for the distance and velocity formula. For a detailed visualization of the integration module's pipeline, including how these coefficients are applied within the broader framework of our model, please refer to Fig. 5 below.

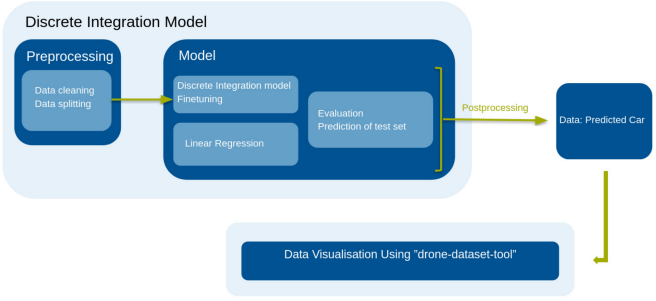

Fig. 6. Overview of the Integration Module

D. Evaluation Metrics and Results

The evaluation of our linear model's performance relies on the following metrics for linear regression: Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) score.

Mean Squared Error (MSE) measures the average squared difference between the actual and predicted values. It penalizes large errors more heavily than smaller ones, making it sensitive to outliers. A lower MSE indicates better model performance.

Mean Absolute Error (MAE) calculates the average absolute difference between the actual and predicted values and is not affected by the scale of the data. Like MSE, lower MAE values indicate better model performance.

R-squared (R^2) score quantifies the proportion of the variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, where a value closer to 1 indicates a better fit of the model to the data.

IV. Results

A. Qualitative Results for the Filtering

In our analysis of exiting behavior, the comparative visualizations between unfiltered Fig. 7 and filtered Fig. 8 datasets illuminate the impact of the filtering module. The unfiltered dataset, while comprehensive, includes noise and irrelevant data points that can obscure critical patterns of exiting behaviour. Conversely, Fig. 8, representing the filtered dataset, showcases a more refined view, where additional information is eliminated, and the relevant behavioural patterns are emphasized. This clarity enhances our ability to precisely identify and analyse exiting manoeuvres, demonstrating the filtering module's efficacy in isolating meaningful data, thus improving the accuracy of our predictive models.

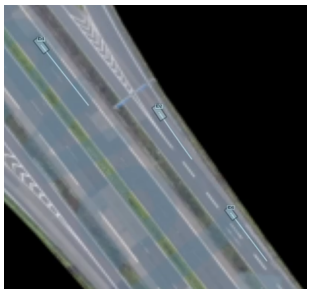

Fig. 7. Unfiltered Exiting Scenario

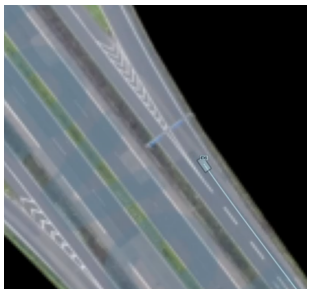

Fig. 8. Filtered Exiting Scenerio

Similar to the analysis of exiting behaviour, the comparison between unfiltered and filtered datasets for entering behaviour, depicted in Fig. 10 and Fig. 9, respectively, offers insightful observations. The unfiltered dataset, depicted in Figure Fig. 10, presents a raw, comprehensive view of vehicle movements, including a mix of relevant and irrelevant behaviours for entering scenarios. 10, on the other hand, reveals the effectiveness of the filtering module in enhancing data clarity.

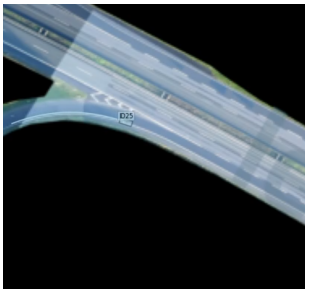

Fig. 9. Unfiltered Entering Scenario

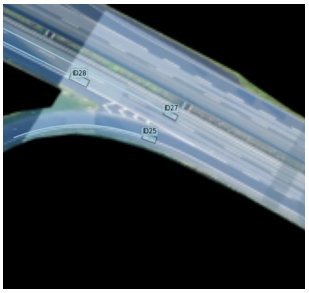

Fig. 10. Filtered Entering Scenario

B. Linear Model

The linear models that predict the acceleration derived from the distance formula (16) and the model derived from the velocity formula (17) solved using linear regression have the following test results:

TABLE I
COMPARISON RESULTS: DISTANCE MODEL AND VELOCITY MODEL

	MSE	MAE	R^2 Score
Distance Model	3.5392×10^{-3}	1.1892×10^{-2}	9.8918×10^{-1}
Velocity Model	3.5370×10^{-3}	1.1877×10^{-2}	9.8918×10^{-1}

We can see that the predictions for the test set for both models are quite accurate having a very small MSE and MAE. We can also plot our results into graphs to have a better visual understanding of our results in figure 11 and 12.

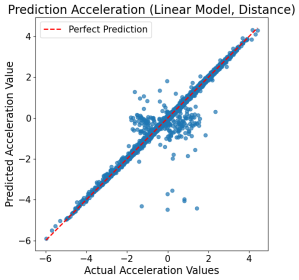

Fig. 11. Prediction results Distance Model

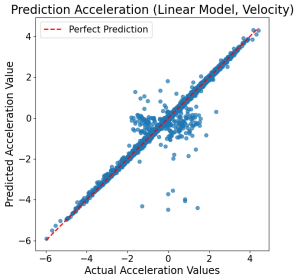

Fig. 12. Prediction results Velocity Model

The graph seems to fit our low MSE and MAE values, as all predicted points are lying close to the red line which indicates a perfect prediction. Interestingly we can see that most inaccuracies lie in the middle where the acceleration is close to zero. Our hypothesis is, that if our acceleration is close to zero the model has the least amount of information for inference, leading to a more scattered prediction in this region.

C. Equivalence of accelerations

Note that both results of the models look quite similar. This is due to the fact, that we specifically trained both models to fit the same dataset. The similarity can be seen in the following figure. The left shows how the predicted accelerations of both our linear models match up, while on the right we can see how the accelerations match up for the previously used ballistic integration method

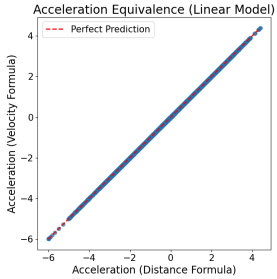

Fig. 13. Acceleration equivalence using linear model

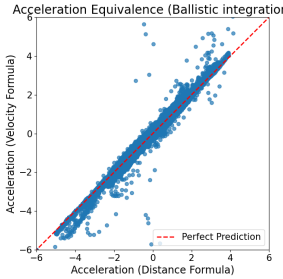

Fig. 14. Acceleration equivalence using ballistic model

Fig. 15. Caption for the entire figure

In the following, we can see the metrics of our results for the equivalence of the accelerations

TABLE II
MODEL PERFORMANCE COMPARISON

Model	MSE	MAE
Linear Model	1.8141e-06	7.5061e-04
Ballistic Integration	2.0698e+02	5.0158e-01

We can see that the accelerations from both our linear models match up almost perfectly, while the results from our previously used model are more scattered.

D. Prediction of the movement

Using the coefficients determined by the linear regression we can rearrange the model such that the model predicts the distance (18) and the velocity (19). The following shows metrics for the prediction of the distance and the velocity using the coefficients.

TABLE III
REGRESSION METRICS

	MSE	MAE	R^2 Score
Distance Formula	1.3867×10^{1}	3.7239×10^{0}	9.8926×10^{-1}
Velocity Formula	2.2635×10^{0}	1.5045×10^{0}	9.3378×10^{-1}

As we can see, all metrics are quite low for both formulas. Though, our prediction for the distance and velocity is not perfect. For a better understanding, we also visualized our findings

Again, we can see that the prediction matches up with the ground truth well. Although we can see a small offset to the red line, especially on the velocity formula, our results might be sufficient for our use case. These results can then be visualized using the drone-dataset-tool provided with the

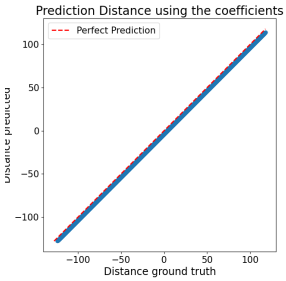

Fig. 16. Prediction of the distance using coefficients

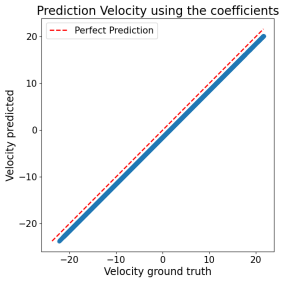

Fig. 17. Prediction of the velocity using coefficients

dataset. Results comparing the prediction of the moving cars to the ground truth can be found here.

V. Conclusion and Future Work

A. Conclusion

We demonstrated that our linear model provided us with more equal results for both the distance model and the acceleration model. This highlights the effectiveness of using a linear model as a discrete integration system. A comparison with the previously used ballistic integration method revealed significantly better performance in terms of accuracy. Finally, using the coefficients obtained from linear regression, we were able to predict both distance and velocity with high accuracy Although some small error was observed, these results might be sufficient for our neural network. Visualizations of these predictions showcased their alignment with the ground truth, further validating the effectiveness of our integration model.

Integral to enhancing our model's precision, the filtering module plays an essential role in preprocessing and refining the data for improved analysis. By systematically identifying key vehicle behaviors through advanced algorithms, this module enables a more nuanced understanding of driving dynamics. This layered approach, combining both granular observation and broader behavioral patterns, enriches our dataset, allowing for a more robust and accurate prediction model. By integrating this layered understanding of social dynamics, the filtering module not only identifies basic maneuvers but also interprets these actions within the broader context of social interaction, thereby enriching the predictive model with a more refined representation of real-world driving conditions. This enhancement significantly increases the model's ability to accurately predict vehicle behaviors, marking an improvement over traditional predictive models by incorporating a deeper comprehension of the complex interplay between individual actions and social influences in driving scenarios

B. Future Work

In future work, we plan to continue refining our integration model to achieve even better results by further fine-tuning its parameters and training procedures.

Additionally, we aim to explore the integration of our discrete integration model with the neural network developed by the previous team. By testing the combined model and comparing its effectiveness against the old method, we can gain insights into its actual performance.

Furthermore, we plan to extend the scope of vehicle behaviours our filtering module can detect, incorporating a wider range of complex driving actions and social interactions. This will involve not only enhancing the algorithmic composure of the module but also diversifying and expanding our datasets to cover more varied and challenging driving scenarios. By doing so, we aim to achieve a richer, more detailed understanding of driver behaviour, further improving the predictive accuracy and reliability of our autonomous driving systems.

References

[1] W. Wang, L. Wang, C. Zhang, C. Liu, and L. Sun, "Social interactions for autonomous driving: A review and perspectives," *Foundations and Trends® in Robotics*, vol. 10, no. 3–4, p. 198–376, 2022. [Online]. Available: http://dx.doi.org/10.1561/2300000078

[2] H. Cui, V. Radosavljevic *et al.*, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[3] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," 2018.

[4] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, "Who are you with and where are you going?" in *CVPR 2011*, 2011, pp. 1345–1352.

[5] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1929–1934.

[6] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exid dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 958–964.

[7] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The round dataset: A drone dataset of road user trajectories at roundabouts in germany," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020.

[8] ika-rwth aachen, "drone-dataset-tool," https://github.com/ika-rwth-aachen/drone-dataset-tools, 2022.