

# A Study of Path Prediction Algorithm for Camera-Based Rider Assistance System

Shoma Hasegawa, Takumi Takeda, Taro Onoue, Akinori Shinagawa  
Yamaha Motor Co., LTD. Japan.

## 1 Introduction

The development and spread of driving assistance systems in four-wheel vehicles has been expanding to motorcycles<sup>(1)</sup>. Among rider assistance systems, camera-based systems feature object recognition and have the potential to respond to traffic conditions more accurately. Consequently, the present research focuses on a motorcycle riding assistance system that uses a camera to inform the rider when the preceding vehicle is close. The behavior of motorcycles differs from that of four-wheeled vehicles, and must be taken into account during the development of functions. One such behavior is banking when cornering, which results in the preceding vehicle appearing inclined in the camera image. Accordingly, a function that appropriately detects the preceding vehicle based on the bank angle has been envisioned. Another motorcycle-specific behavior is the high degree of flexibility with respect to the driving trajectory within a lane. Relative vehicle positions relying on lane markings therefore cannot be used to distinguish the relevant preceding vehicle from other vehicles. This has led to envisioning a function that identifies oncoming or other non-preceding vehicles based on the behavior of other vehicles. Both the ego and other vehicles path prediction algorithms required to realize those functions have already been proposed<sup>(3)</sup>. This study examines whether the proposed algorithms select the appropriate object and avoid selecting the wrong one in straight line, cornering, and lane changing scenarios.

## 2 Functional Overview

This section provides an overview of the path prediction algorithm functionality<sup>(3)</sup>.

### 2.1 System functional structure

Figure 1 shows the functional structure of the rider assistance system used in this study. The system takes camera images, as well as acceleration, angular velocity, and vehicle speed as inputs, and uses a buzzer sound and a flashing LED as outputs. The functional structure of the system consists of object detection, ego vehicle path prediction, target selection, time-to-collision (TTC) estimation, and human machine interface (HMI) output control. Of the potential targets detected in the camera images, only the one selected as a focus based on path prediction is subject to TTC estimation and output decision. This paper describes both the ego and other vehicle path prediction algorithm functions, as well as the target selection function.

### 2.2 Ego vehicle path prediction function

This function for the ego vehicle takes angular velocity and vehicle speed as inputs, calculates the turning radius, and then predicts the path of the ego vehicle on the camera image. Figure 2 shows the predicted path of the ego vehicle as it follows the preceding vehicle and goes by an oncoming vehicle. Using the estimated roll angle to correct the path enables appropriate prediction that accounts for the banking of the vehicle body even during cornering.

### 2.3 Other vehicle path prediction function

This function calculates the movement vector (direction, magnitude) on the image based on the bounding box (BB) coordinates of the detected object and its previous coordinates, and predicts the path of other vehicles on the camera image as shown in Fig. 3. Limiting the input to the BB data makes it possible to predict paths irrespective of white lines, and without depending on the ego vehicle's position.

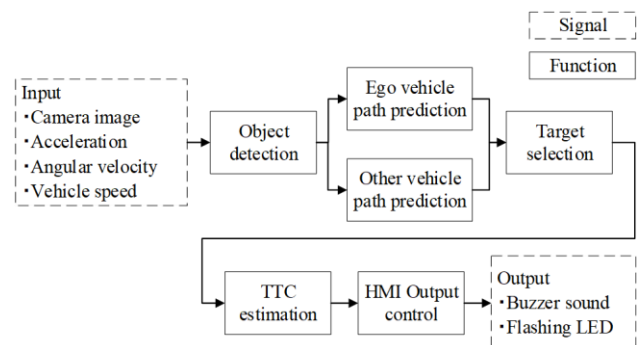


Fig. 1: System functional structure

## 2.4 Target selection function

This function assigns a selection priority to target vehicles along the predicted path in order of proximity to the ego vehicle. It then calculates the probability of a collision and of a change in path based on the predicted path of other vehicles, and excludes vehicles determined to present no risk of collision with the ego vehicle. That process leads to selecting the target to focus on, as shown in Fig. 4.

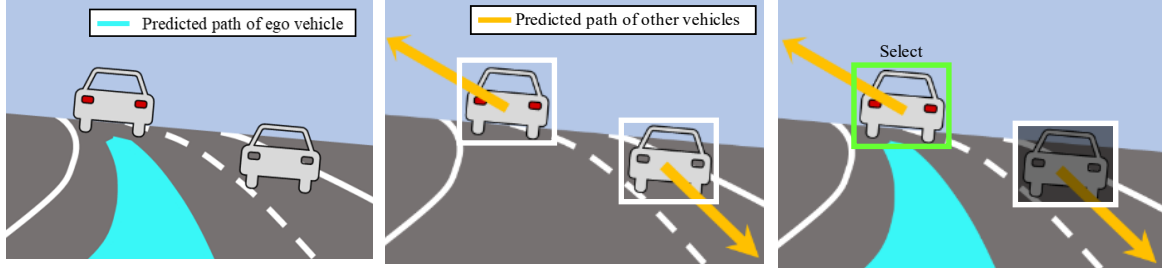


Fig. 2: Predicted Path of Ego Vehicle

Fig. 3: Predicted Path of Other Vehicles

Fig. 4: Selected Vehicle Based on Predicted Path

## 3 The Algorithms

### 3.1 Ego vehicle path prediction algorithm

This algorithm involves calculating the path, converting it to camera coordinates, and correcting the roll. As shown in Fig. 5, the algorithm establishes a geographic coordinate system using the position of the camera on the ego vehicle as the origin, and setting the course of the ego vehicle  $x$  axis, and the left side of the course parallel to the ground as the  $y$  axis. In addition,  $(x_g, y_g)$  is used to indicate an arbitrary point in the geographic coordinates. As shown in Fig. 6, the algorithm establishes a camera image coordinate system using the top left of the image as the origin, using the width direction as the  $x$  axis and the height direction as the  $y$  axis.

In addition,  $(x_s, y_s)$  is used to indicate an arbitrary point in the camera image coordinates. The path of the ego vehicle is calculated in the geographic coordinate system shown in Fig. 5 with the equation below, which uses the  $r$ [m] turning radius obtained from the vehicle speed and yaw rate.

$$y_g = r - \sqrt{r^2 - x_g^2} \quad (1)$$

The following equation is used to convert that path into coordinates in the camera coordinate system from Fig. 6 and calculate the predicted path.

$$x_s = \frac{W_s}{2} - y_g \frac{W_s}{2x_g \tan\left(\frac{\phi}{2}\right)} \quad (2)$$

In this equation,  $W_s$ [pix] is the screen width, and  $\phi$ [deg] is the horizontal angle of view. At the same time, the roll angle  $\theta_x$ [deg] estimated from factors such as the roll rate is used to rotate the coordinates to the center of the screen and correct the predicted path. The coordinates after correction are indicated by  $(x_s', y_s')$ .

$$\begin{pmatrix} x_s' \\ y_s' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \frac{W_s}{2} \\ 0 & 1 & \frac{H_s}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos \theta_x & \sin \theta_x & 0 \\ -\sin \theta_x & \cos \theta_x & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -\frac{W_s}{2} \\ 0 & 1 & -\frac{H_s}{2} \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_s \\ y_s \\ 1 \end{pmatrix} \quad (3)$$

In this equation,  $H_s$ [pix] is the screen height. To avoid being unable to capture the preceding vehicle due to excessive variation in the path, the straight line state is determined and the

predicted path is fixed to straight ahead of the camera when moving in a straight line. The yaw rate is used to determine the straight line state.

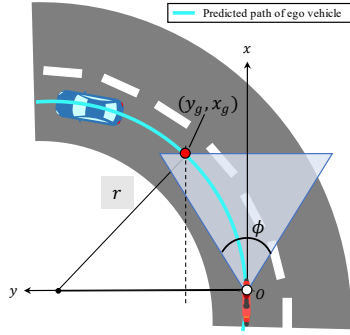


Fig. 5: Top View of Ego Vehicle's Predicted Path

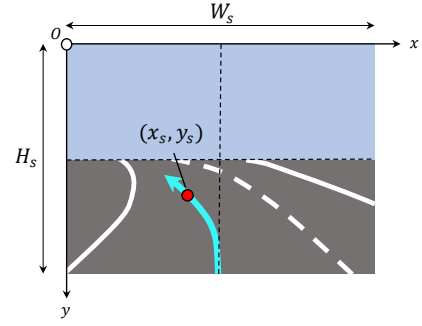


Fig. 6: Camera image of Ego Vehicle's Predicted Path

### 3.2 Other vehicle path prediction algorithm

This algorithm involves calculating the movement vector and deriving path coordinates. As shown in Fig. 7, the direction and magnitude of vehicles are, respectively, obtained by detecting the BB coordinates of the current frame and from the difference with the detected BB coordinates of a given previous frame.

$$\angle \mathbf{A} = \tan^{-1} \left( \frac{y_t - y_{t-L}}{x_t - x_{t-L}} \right) \quad (4)$$

$$|\mathbf{A}| = \sqrt{(x_t - x_{t-L})^2 + (y_t - y_{t-L})^2} \quad (5)$$

In these equations,  $\angle \mathbf{A}$  is the direction of the movement vector,  $|\mathbf{A}|$  is its magnitude,  $L$  is the difference in the number of frames,  $(x_t, y_t)$  represents the coordinates of the center of the base of the detected BB at time  $t$ [frame], and  $(x_{t-L}, y_{t-L})$  represent the coordinates for  $L$ [frame] earlier. In addition, the BB coordinates used to calculate the movement vector cancel out the ego vehicle movement component to extract only the other vehicle movement component. The predicted path of other vehicles is calculated on the premise that it will keep extending in a straight line in the direction of the vector on the screen.

$$y_s = (x_s - x_c) \frac{\sin(\angle \mathbf{A})}{\cos(\angle \mathbf{A})} + y_c \quad (6)$$

In the equation,  $(x_c, y_c)$  represents the center coordinates of the detected BB.

### 3.3 Target selection algorithm

This algorithm consists of determining target selection priority based on the predicted path of the ego vehicle, calculating the probability of collision based on the predicted path of other vehicles, calculating the probability of a change in the trajectory of other vehicles, and selecting the target vehicle.

#### 3.3.1 Determining the target selection priority

Figure 8 illustrates how target selection priority is determined. After narrowing down potential targets in the selection range, the target selection is set in order of the smallest to largest values of  $P$  obtained from the equation below.

$$P = k_p d_y + (1 - k_p) d_x \quad (7)$$

In the equation,  $d_x$ [pix] is the distance in the  $x$  axis direction between the BB base center

coordinates and the ego vehicle's path,  $d_y[\text{pix}]$  is the distance in the  $y$  axis direction between the BB base center coordinates, and  $k_p$  is a weight coefficient that determines whether to prioritize proximity to the vehicle or proximity to the predicted path. The target selection range is defined as a range consisting of the width of the ego vehicle centered on the predicted path combined with a range extended vertically from the end of the predicted path. Differences in target vehicle sizes are taken into account by considering objects that overlap with the target selection range and the BB base to be within the target selection range.

### 3.3.2 Calculating the collision probability

Figure 9 illustrates how the probability of a collision is calculated. The Gaussian distribution obtained from the predicted path and the integral of the  $W$  ego vehicle width range are used to calculate the collision probability. The average  $\mu$  of the Gaussian distribution represents the  $x$  coordinates at the intersection of the predicted path and the bottom edge of the screen, while the standard deviation  $\sigma$  defines the reciprocal of the magnitude of the movement vector based on the BB center coordinates.

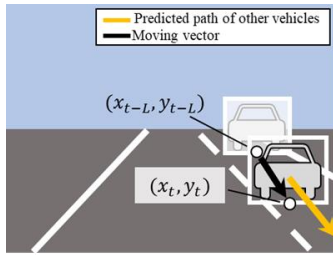


Fig. 7: Movement Vector and Other Vehicle's Predicted Path

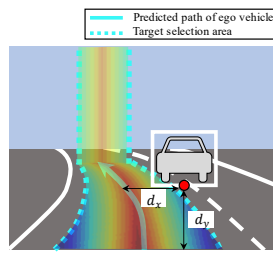


Fig. 8: Target Selection Priority Determination

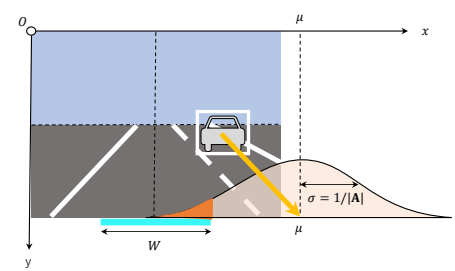


Fig. 9: Collision Probability Calculation

### 3.3.3 Calculating the probability of a change in trajectory

Two probabilities of a change in trajectory are calculated based on the magnitude of the movement vector and on the continuity of its direction. The probability of a change in trajectory based on the magnitude of the movement vector decreases according to the premise that a change in trajectory is unlikely as the magnitude of the vector increases. The probability of a change in trajectory based on the continuity of the direction of the movement vector decreases according to the premise that a change in trajectory is unlikely while the variation in direction remains within a certain range.

### 3.3.4 Selecting the target

Vehicles requiring caution are selected in order of highest priority among the vehicle inside the target selection range. In addition, vehicles with both a collision probability below the threshold and one of the two probabilities of a change in trajectory below the threshold are excluded from the selection targets.

## 4 Testing

Table 1 shows the items that were validated for the path prediction algorithm.

Table 1: Validation Items

No.	Validation item
A	The correct object is selected as the target vehicle.
B	No incorrect object is selected as the target vehicle.

### 4.1 Test device

The experiment was conducted using a camera system implementing all of the functions shown in Fig. 1. The device was equipped with a camera and an inertial measurement unit (IMU) and

installed on a motorcycle, enabling the real time execution of the proposed algorithm. Individual video frames, angular velocity, and vehicle speed can also be retrieved ahead of time, making it possible to enter that information from an external source and validate it analytically. In this experiment, the validation scenarios were conducted by driving on a closed course and the information retrieved was verified through analysis.

#### 4.2 Test conditions

The validation scenarios, validation sections, index, and collection method used as test conditions are described below.

##### 4.2.1 Validation scenarios

Table 2 shows the validation scenarios for validation item A.

Table 2: Scenarios to Confirm the Correct Object Is Selected (Validation Item A)

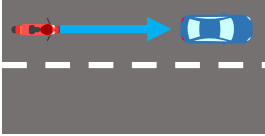

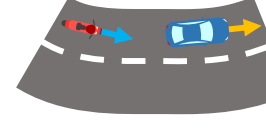
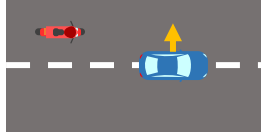
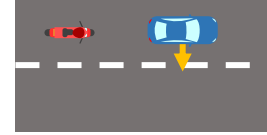
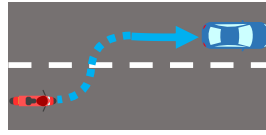
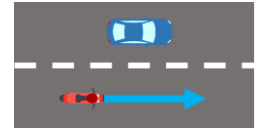
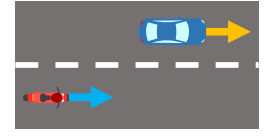
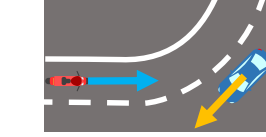
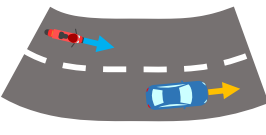
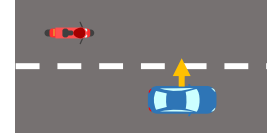
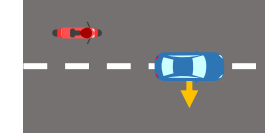
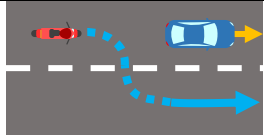
Scenario No.	A00	A01	A10
Details	Approaching a stationary vehicle ahead	Following the preceding vehicle	Following the preceding vehicle while cornering
Overview			
Scenario No.	A20	A21	A22
Details	The preceding vehicle cuts into the lane	The preceding vehicle moves into the other lane	The ego vehicle cuts into the lane
Overview			

Table 3 shows the validation scenarios for validation item B.

Table 3: Scenarios to Confirm No Incorrect Object Is Selected (Validation Item B)

Scenario No.	B00	B01	B02
Details	Overtaking a stationary vehicle in the other lane	Maintaining distance from the vehicle in the other lane while driving	Moving past a vehicle cornering ahead while driving straight
Overview			
Scenario No.	B10	B20	B21
Details	Maintaining distance from the vehicle in the other lane while cornering	The preceding vehicle cuts into the lane	The preceding vehicle moves into the other lane
Overview			

Scenario No.	B22		
Details	The ego vehicle moves into the other lane		
Overview			

#### 4.2.2 Validation sections

The validation sections were defined as a frame in which the ego vehicle drives in accordance with the conditions and the target vehicle is correctly detected. However, in the lane changing scenes, different validation sections were set based on the driving state, as shown in Figs. 10 and 11.

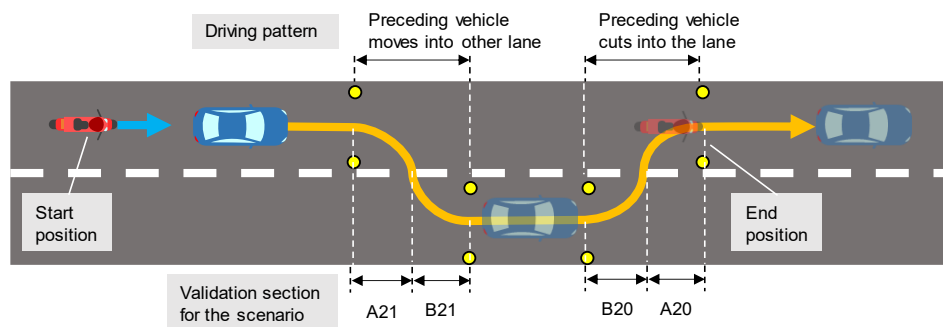


Fig. 10: Validation Section for the Preceding Vehicle Lane Change Scenario

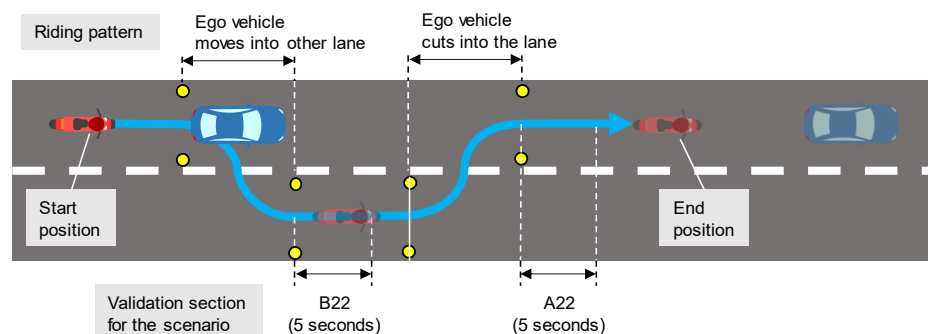


Fig. 11: Validation Section for the Ego Vehicle Lane Change Scenario

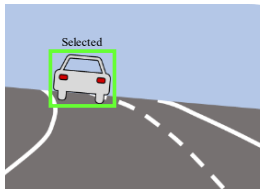
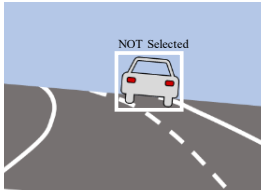
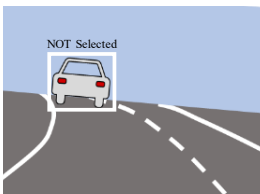
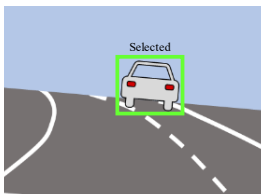
#### 4.2.3 Index

The index represents the correct vehicle selection ratio. That ratio consists of the proportion of frames where the algorithm selects the intended frame within a validation section.

#### 4.2.4 Collection method

The correct vehicle selection ratio was calculated using the number of frames in the validation section after counting the number of correct frames. The number of correct frames was counted by determining the correct vehicle selection from the images output by the system. The criteria for correctness depends on the scenario. Table 4 shows decision examples for each validation item. In the validation A scenarios where the vehicle is in the same lane as the ego vehicle, correctness was defined as the ability to select the preceding vehicle in the same lane. In the validation B scenarios where the vehicle is not in the same lane as the ego vehicle, correctness was defined as the ability to select the vehicle in the other lane. The collection process was designed to retrieve at least 10 seconds of data for each scenario.

Table 4: Examples of Correct Vehicle Selection Decision

	Validation item A	Validation item B
Correct		
Incorrect		

### 4.3 Results

This section presents the collected results for the correct vehicle selection ratio. Table 5 shows the collected results for validation item A. The A00 and A01 results confirm that the vehicle is correctly selected when driving in a straight line, while the A10 results show that the correct selection ratio drops during cornering when there is a lot of distance between the vehicles. The A20 scenario also demonstrated a drop in the correct vehicle selection when the preceding vehicle cuts into the lane. In contrast, the A21 scenario showed that when the preceding vehicle moves into the lane, the system can continue to make the selection while that vehicle is still in the same lane as the ego vehicle. In addition, the A22 confirms that the target vehicle is selected correctly after the preceding vehicle has fully entered the ego vehicle's lane.

Table 5: Collected Results for Selecting the Correct (Validation Item A)

Scenario	A00	A01	A10	A20	A21	A22
Vehicle speed	50 km/h	50 km/h	50 km/h	50 km/h	50 km/h	50 km/h
Distance between vehicles	-	12 m	40 m	12 m	40 m	-
Turning radius	-	-	-	425 m	425 m	-
Correct selection ratio	96.7%	100%	100%	98.2%	66.0%	56.1%
						87.8%
						82.9%

Table 6 shows the collected results for validation item B. The B00 and B01 results confirm that a other vehicle in the other lane is not incorrectly selected when driving in a straight line, while the B02 results confirm that going past an oncoming vehicle in a curve ahead does not incorrectly select that vehicle. In those scenarios, the correct answer cannot be obtained using only ego vehicle information, but using the predicted path of other vehicles made it possible to exclude them from the selection. The B10 results confirm that other vehicles ahead of the ego vehicle are not incorrectly selected during cornering. The B20 results show that the preceding vehicle cutting into the lane is not selected while it is still in the other lane. Similarly, the B21 results show that the preceding vehicle moving into the other lane is not selected after it enters that lane. In addition, the B22 results confirm that the preceding vehicle is not selected when the ego vehicle finishes moving into the other lane.

Table 6: Collected Results for Not Making an Incorrect Selection (Validation Item B)

Scenario	B00	B01		B02	B10	B20	B21	B22
Vehicle speed	50 km/h	50 km/h	50 km/h	30 km/h	50 km/h	50 km/h	50 km/h	50 km/h
Distance between vehicles	-	12 m	40 m	-	40 m	-	-	-
Turning radius	-	-	-	80 m	425 m	-	-	-
Correct selection ratio	98.9%	100%	100%	84.9%	86.0%	89.1%	98.6%	96.6%

#### 4.4 Observations

This section examines the factors behind incorrect selections and their impact on the system.

##### 4.4.1 Scenario A10: Target selection during cornering

The factors leading to incorrect selection were examined first. Table 5 shows that during cornering, the correct selection ratio is lower at a vehicle distance of 40 m than at a distance of 12 m. This is expected to be due to either the ego vehicle path prediction algorithm or to the actual driving during the test. The algorithm was assessed first. The ego vehicle path prediction algorithm estimates the turn and predicts the path based on the current turning radius of the ego vehicle. Consequently, the further the distance the greater the span of variation, making it more difficult to capture the preceding vehicle in the path of the ego vehicle. The reason for the drop in the correct selection ratio at further distances is thought to be that the larger span of variation in the path of the ego vehicle relative to the variation in the turning radius prevented the capture of the preceding vehicle in the path of the ego vehicle. Next, the driving during the test was examined. Sections with an incorrect selected were observed to generally involve driving at a smaller turning radius than the path. Figure 12 shows the turning radius distribution for correct and incorrect selections for the data from a single trial on a road with a 425 m turning radius and a vehicle distance of 40 m. The graph makes it clear that that the ego vehicle was driving at a turning radius close to 425 m when correct selections were made, but at a turning radius around 250 m to 350 m when incorrect selections were made. The drop in the correct selection ratio is believed to stem from temporarily driving at a smaller turning radius to adjust the driving position, leading to predicting a path inward of that of the preceding vehicle.

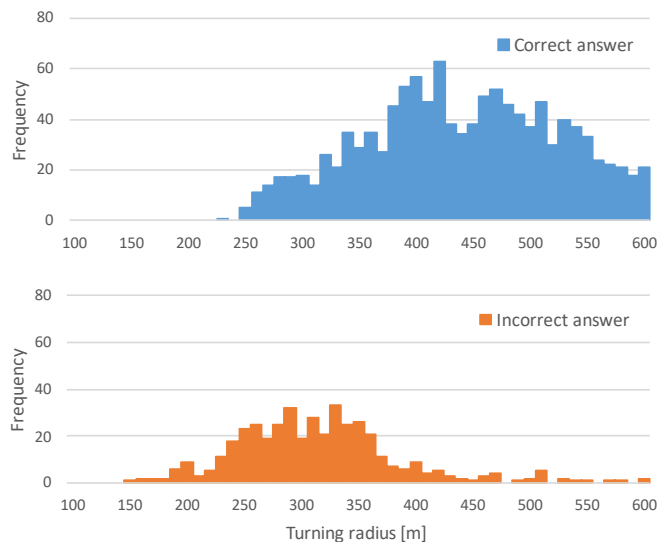


Fig. 12: Turning Radius Distribution for Correct and Incorrect Selections in the Following while Cornering Scenario.



#### 4.4.2 Scenario A20: Preceding vehicle cutting into the lane

The A20 scenario was examined next. As seen in Table 5, this scenario has a poorer correct selection ratio than the other lane change scenarios. That drop is attributed to the path prediction algorithm for other vehicles. Figure 13 shows the camera images for the A20 scenario. The light blue solid lines represent the predicted path of the ego vehicle, while the orange solid lines represent the predicted path of other. In addition, the green box around the preceding vehicle indicates it has been selected as the target vehicle, and the orange box indicates the other vehicle has been excluded from selection based on its predicted path. The number at the upper left is the frame number.

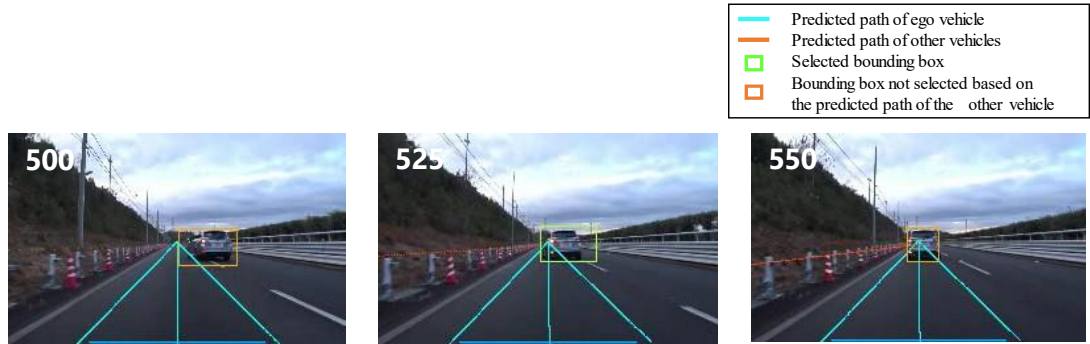


Fig. 13: Target Selection when the Preceding Vehicle Cuts into the Lane

The images demonstrate that the other vehicle was excluded from the selection based on its predicted path. The same phenomenon was observed in other data for the same scenario. In the preceding vehicle lane change scenarios, that vehicle only moves toward the side in the image. It seems that the algorithm therefore predicts a path that does not collide with the ego vehicle, and excludes the vehicle from the selection. This makes it a scenario that is not covered by the current algorithm.

Furthermore, the impact of this phenomenon on the system was examined. The durations of the delay in selecting the target were collated to assess the impact of that delay due to the exclusion of the vehicle from the selection. The selection delay was obtained using the A20 test conditions by collating the time from a stable continuous full second until the first frame of the section in which the vehicle was selected after the preceding vehicle entered the ego vehicle's lane. The results are shown in Fig. 14, and indicate that the target vehicle is chosen after a maximum delay of 2.2 seconds. There are concerns that during that selection delay, the system would not react to approaching the preceding vehicle and fail to activate.

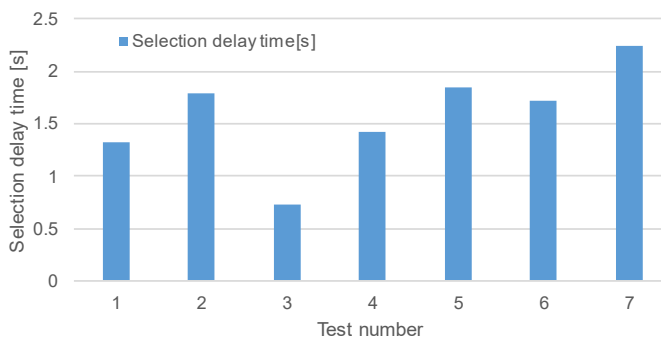


Fig. 14: Duration of Target Selection Delay When the Preceding Vehicle Cuts into the Lane

#### 4.4.3 Scenario A22: Ego vehicle cutting into the lane

The impact on the system of the delay in selecting the target vehicle after changing lanes was examined for the A22 scenario. The selection delay was obtained using the A22 test conditions by collating the time taken for the system to stabilize and select the target vehicle after the ego vehicle finishes its lane change. The results are shown in Fig. 15, and indicate that the target vehicle is chosen after a maximum delay of 1.6 seconds. There are concerns that during that selection delay, the system would not react to approaching the preceding vehicle and fail to activate.

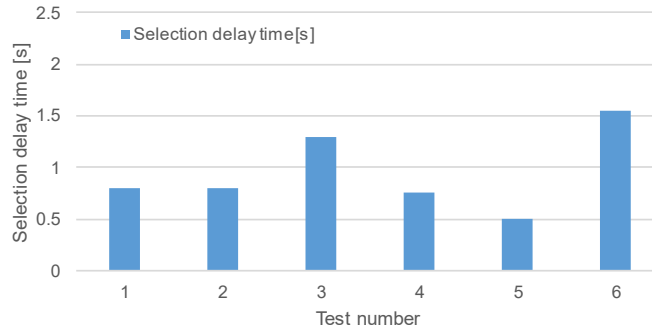


Fig. 15: Duration of Target Selection Delay When the Ego Vehicle Cuts into the Lane

## 5 Conclusion

This study analyzed and examined the effectiveness of ego vehicle and other vehicle path prediction algorithm that accounts for ego vehicle behavior for a rider assistance system intended for two-wheeled vehicles. The following results were obtained.

- The proposed algorithm was confirmed to select the correct target object in six of a total of eight scenarios involving straight line driving, cornering, and lane changes.
- The proposed algorithm was confirmed to avoid selecting an incorrect target object in all of the eight scenarios involving straight line driving, cornering, and lane changes.
- In the scenario involving the preceding vehicle cutting into the lane, it was found that correct selection of the preceding vehicle was limited to 56% of the frames. At the same time, incorrect selection was identified as a factor of the other vehicle path prediction algorithm, and the current algorithm was determined to be unable to address that scenario.

To achieve the eventual adoption of the algorithms in rider assistance systems for two-wheeled vehicles, the next step will be to address the issues brought to light in this study and perform evaluations in situations similar to the actual use environment.

## 6 References

- 1) 2023 TRACER 9 GT+ European Model Exhibited at EICMA - Equipped with the World's First UBS Linked to the Millimeter Wave Radar, Adaptive Cruise Control, and More <https://global.yamaha-motor.com/jp/news/2022/1108/tracer9.html>
- 2) DMP: DMP, Launch External Sales of Camera System, the Result of Collaboration with Yamaha Motor, DMP News, <https://www.dmpof.com/en/news/dmp-launch-external-sales-of-camera-system-the-result-of-collaboration-with-yamaha-motor/>
- 3) S. Hasegawa, T. Takeda, T. Onoue, A. Shinagawa : "Path Prediction Algorithm for Motorcycle Rider Assistance System with Camera", *Transactions of Society of Automotive Engineers of Japan*, Vol. 55, No. 5, pp. 967-971, 2024