Proceedings of the 2025 SICE Festival with Annual Conference
September 9-12, 2025, Chiang Mai, Thailand

Development of a Pacemaking Runner Robot for Long-Distance Running*

Kazunori MINAKAWA[1], Jumpei SUGITA[1], Yasushi ENOMOTO[2], and Hisashi DATE[3]

***Abstract*— It is essential for long-distance runners to take and maintain long strides in order to become faster. Long stride movements require a training environment in which the runner's pace is stable at 5.5 m/s or more. We have developed a pacemaking runner robot designed to provide a constant pace running environment. This robot utilizes a camera for white line detection and the running lane recognition, controlling steering via the Pure Pursuit method. In addition, a 3D LiDAR system is incorporated as an important backup feature to ensure continued operation in the event that camera detection is compromised by environmental factors. During camera-based running at 8 m/s, we observed the robot exhibiting oscillation and deviating from the running lane. We have successfully reproduced this oscillation in a model that takes into account the system's dead time. We attributed the system dead time to the combined effects of the camera frame rate and motor communication delay. This analysis reveals that dead time is an important factor in high-speed stability.**

I. INTRODUCTION

In long-distance running, a runner's pace is determined by the product of their cadence and stride length. This relationship is crucial for a runner's performance.

Research by Kurosaki et al. has shown that elite athletes who run 10,000 m in under 27 minutes can maintain a longer stride length compared to student athletes who complete the same distance in over 30 minutes [1]. This suggests that for student-athletes to improve their performance, they must be able to secure and maintain a longer stride.

To maintain long strides at the same pace, the cadence should be lowered. To maintain long strides at the same pace, the cadence should be lowered. However, Oda reported that runners cannot control all of their muscles during a run [2], making it difficult to control and consciously lower one's cadence. A situation that allows the runner to maintain a constant pace and unconsciously lower the cadence will help the runner master the movement of taking long strides. The pace required is 5.5 m/s or faster, which is the typical pace for student athletes.

One way to keep a runner's pace is to follow a pacer who runs at a constant pace. Pacemakers in long-distance running help maintain the pace of the runners behind them.

This work was partly supported by JSPS KAKENHI Grant Number 23K03896.

[1]Kazunori MINAKAWA and Jumpei SUGITA are with Graduate School of Science and Technology, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Japan minagawa-k@roboken.iit.tsukuba.acjp

[2]Yasushi ENOMOTO is with Institute of Health and Sport Sciences, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Japan

[3]Hisashi DATE is with Institute of Systems and Information Engineering, University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Japan hdate@iit.tsukuba.ac.jp

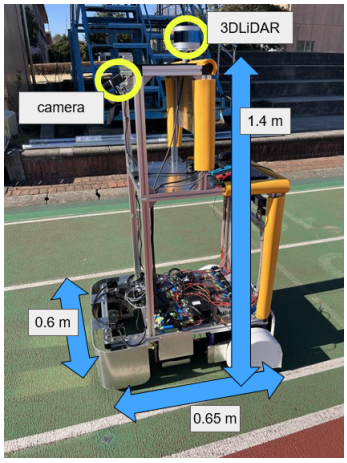

Fig. 1: A Pacemaking Runner Robot for Long-Distance Running developed by Ban et al.

But weather conditions and fatigue can make it difficult to maintain the constant pace. There is an electronic pacer with LED lights installed under the curb of the stadium [3] and a shoebox-shaped BeatBot robot developed by Puma [4]. It may be possible to maintain pace by following these, but too much attention is needed to follow, which may hinder their ability to learn long-stride movements.

A Pacemaking Runner Robot for long-distance running, shown in Fig. 1, was developed by Ban et al. [5] as a robot that can run at speeds of 5.5 m/s or more and prevent the loss of line of sight. The dimensions of the robot are 1.4 m in height, 0.6 m in tread width, and 0.65 m in wheelbase length, which are about the same size as the space an athlete occupies while running. The robot detects the white lines on a track with a camera mounted in front of the robot and runs at a specified speed. Sugita et al. have confirmed that 3DLiDAR attached to the top of the robot enables the robot position to be estimated [6]. However, the accuracy of self-position estimated with LiDAR is about 30 cm, while that of the camera is as high as less than 1 cm. This does not meet the location accuracy for lane room recommended by Reid et al. [7]. For this reason, 3D LiDAR driving is used as a backup in situations where camera detection shows deficiencies.

This study summarizes the development of the Pacemaking Runner Robot for long-distance running. We also report that we have identified the cause of the problems encountered during development.

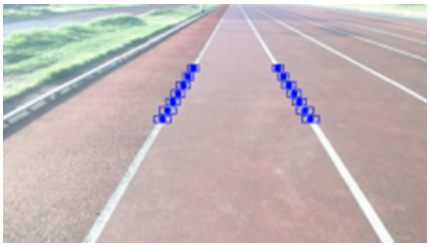

Fig. 2: Example of Tracking Window (TW) Placement detection point, based on the authors' work [8].

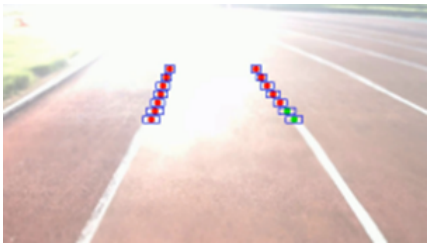

Fig. 3: Example of TW placement where detection points cannot be identified due to sunlight, based on the authors' work [8].

II. ROBOT SYSTEM

In this chapter, we summarize the pacemaking robot developed by Ban et al. [5] [8] and Sugita et al. [6].

A. Camera-Based Lane Detection

The robot detects the running lane by identifying the white lines on a track using a forward-mounted camera. This detection method is based on the study by Rui et al. [9]. Specifically, as shown in Fig. 2, seven tracking windows (TWs) are placed at regular intervals on the left and right white lines of the running lane. By using a weighted least squares method on the center of gravity of the white line within each window, the system derives an approximate straight line to detect the running lane.

However, as shown in Fig. 3, this method can fail to detect the running lane due to overexposure caused by direct sunlight or when the other runners obstruct the white line.

To solve these problems, we track four white lines—the two lines defining the running lane and the two lines on either side—as shown in Fig. 4. This configuration allows the system to reliably detect lane lines even when a portion of the line is blocked by a runner or is washed out. By utilizing four lane lines, the robot is able to detect the running lane more stably and reliably.

B. Driving Control Method

The control method used to ensure that the robot follows the running lane without deviation is the Pure Pursuit method [10]. In the Pure Pursuit method, a point located ahead on the running lane is set as the target point, and the steering angle is calculated based on the relative angle between the robot and the running lane, as well as the lateral deviation of the robot from the running lane.

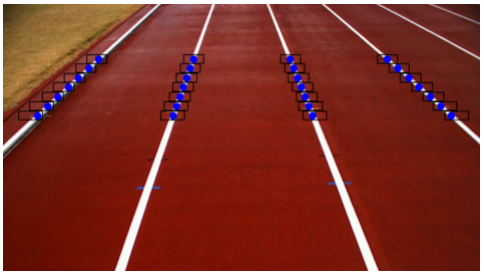

Fig. 4: Example of TW Placement on Four White Lane, based on the authors' work [5].

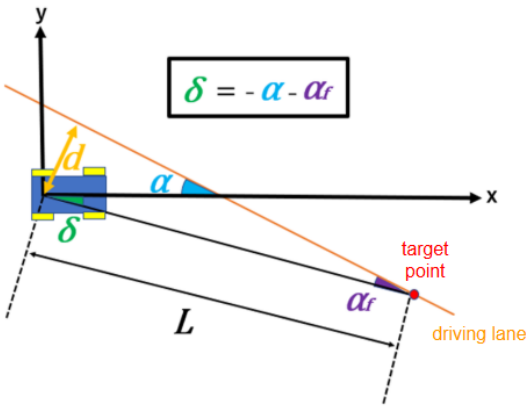

Fig. 5: Steering Angle of the Robot by Pure Pursuit, based on the authors' work [5]

As shown in Fig. 5, the steering angle δ is calculated using the relative angle α and the lateral deviation d of the robot from the running lane, as expressed in (1). Here, L represents the look-ahead distance, which is the distance between the robot and the target point.

$$\delta(t) = -\alpha(t) - \arcsin\left(\frac{d(t)}{L}\right) \tag{1}$$

By applying this control method, the robot was able to stably follow the track at a speed of 4.0 m/s, as shown in Fig. 6. When the speed was increased, stable running was maintained up to 6.0 m/s. However, at 8.0 m/s, a lateral oscillation was observed, and eventually it deviated from the running lane.

C. Lidar-Based Driving

As a countermeasure for situations where white line detection fails, we examined whether the robot could maintain stable running through self-localization using the 3D LiDAR mounted on top of the robot.

By combining the 3D LiDAR with an IMU and employing the LiDAR Inertial Odometry and Mapping(LIO-SAM) method [11], a 3D point cloud map of a track was constructed. During autonomous driving, the running path was set based on the official dimensions of a track, as shown in Fig. 7. Similar to camera-based driving, the Pure Pursuit

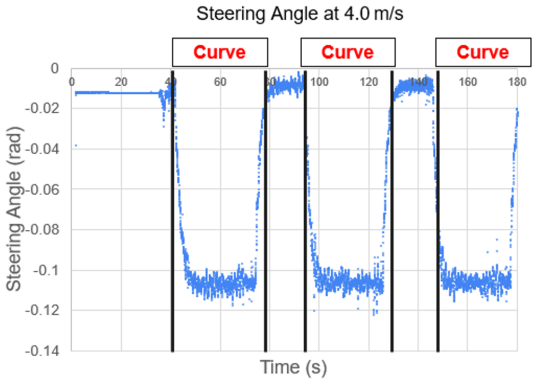

Fig. 6: Steering Angle Variation While Circling a track at 4 m/s. The segments between curves correspond to straight sections of the track.

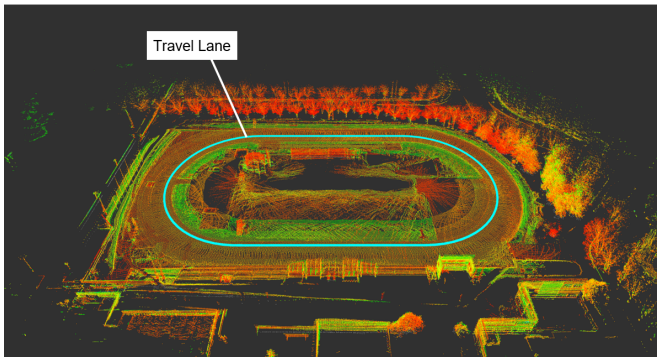

Fig. 7: 3D Point Cloud Map of a track with driving lane.

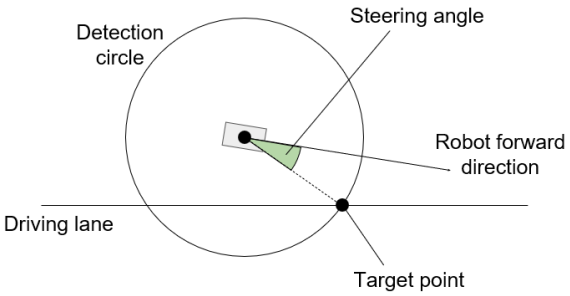

Fig. 8: Pure Pursuit Path Following Model.

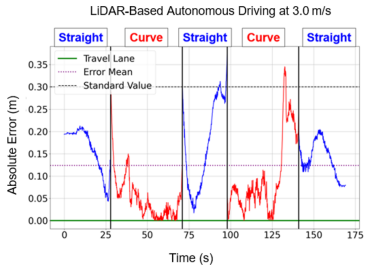

Fig. 9: Lateral Position Error Over While Circling an Athletic Track at 3.0 m/s, using 3D LiDAR.This figure is reproduced from [6], which is the authors' work, in accordance with Article 7 of the Copyright Regulations of the Japan Society of Mechanical Engineers.

method was utilized, and the target point was determined using a virtual search circle centered on the robot, as depicted in Fig. 8. Among the two intersection points between the search circle and the running lane, the point on the robot ' s forward side was selected as the target point for steering control.

For localization, we used `hdl_localization` [12]. As shown in Fig. 9, it was confirmed that the robot could travel at a low speed of 3.0 m/s while performing self-localization. This provides a backup for the robot to continue traveling in situations where detection of the running lane by the camera is difficult.

III. Driving Using Camera and 3D LiDAR

This chapter summarizes the integrated operation of camera and 3D LiDAR systems for navigation. Since camera-based driving is more accurate than 3D LiDAR, we developed a system that normally uses camera-based driving, and 3D LiDAR driving can be used as a backup in case of failure in detecting the lanes.

The basic system configuration is shown in Fig. 10. During normal operation, both the camera and 3D LiDAR constantly calculate steering angles. The Steering switcher normally uses the camera's steering angle for driving control. When the lane estimation by the camera system fails, the Steering Switcher switches to use the steering angle from the 3D LiDAR system.

During normal operation, the camera system continuously records the positions of the TWs in the TW storage list for recovery purposes. When camera lane recognition fails, after a certain period of time has elapsed, the camera system retrieves the TW position one second before failure from the TW storage list and attempts to resume lane estimation. The highest number of white line detections is used for lane estimation, since the robot is moving during this time, and if all detections are below the threshold, lane estimation is tried again after a certain period of time.

As shown in Fig. 11, the development of a 3D LiDAR camera backup system ensured the continuity of stable running. However, at the same speed, the angular amplitude of the oscillation is larger for 3D LiDAR than for the camera, indicating that the 3D LiDAR driving is less stable than the camera. One possible reason for this is that the camera period is 0.02 s (50 fps) and the 3D LiDAR period is 0.1 s (10 Hz), and this difference in period may affect the stability of the system.

IV. Oscillation in a Pacemaking Runner Robot

Ban et al. [5] reported that an oscillation occurs at a speed of 8.0 m/s in camera runs, causing the vehicle to deviate from the running lane. The running control method of this robot is the Pure Pursuit method, which can be linearly approximated

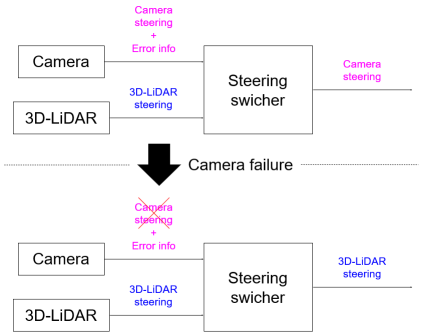

Fig. 10: Backup system architecture.

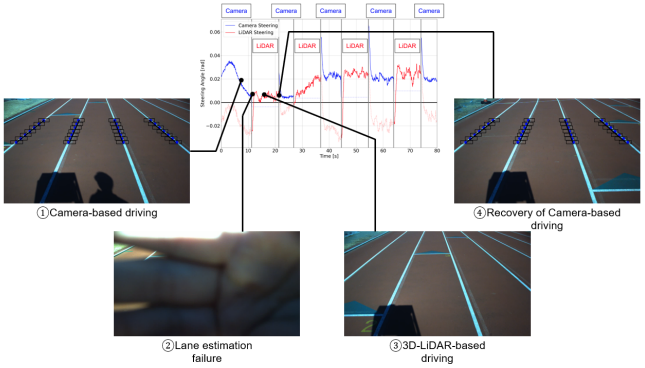

Fig. 11: Steering angle during camera and 3D LiDAR switching.

because the angles and deviations are sufficiently small in the actual running conditions. The Pure Pursuit method with linear approximation does not predict the vibration, but the actual running of the robot causes an oscillation. It should be noted that 3D LiDAR has a larger amplitude and longer period at the same speed. The sensor period in the calculation of the steering angle is the dead time in the system. An oscillation is reproduced in a model that takes into account dead time, and the factors that cause an oscillation to occur in this robot are clarified.

A. Kinematic Model of Steering Robot

The robot used in this study is a four-wheeled vehicle with front-wheel steering and rear-wheel drive. When the robot moves at forward speed v and steering angle δ, the angular velocity ω of the robot is expressed as (2) from robot kinematics.

W is the wheelbase length of the robot.

$$\omega(t) = \frac{v(t)\tan\delta(t)}{W} \tag{2}$$

The steering angle δ is obtained by (3) using the relative angle of the robot to the travel lane α and the lateral deviation of the robot relative to the travel lane d and look-ahead distance L based on the Pure Pursuit method.

$$\delta(t) = -\alpha(t) - \arcsin\left(\frac{d(t)}{L}\right) \tag{3}$$

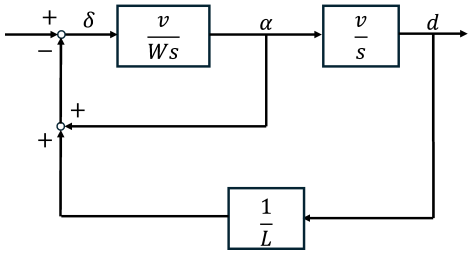

Fig. 12: Block Diagram of Robot Kinematics and Pure Pursuit Control.

The lateral deviation d is sufficiently small compared to the distance L to the target point and can be approximated as $\arcsin d/L \approx d/L$. Since the steering angle δ is also small enough, it can be approximated as $\tan\delta(t) \approx \delta(t)$. Applying these approximations to (2) and (3), the angular velocity ω is given by equation (4).

$$\omega(t) = \frac{v(t)\left(-\alpha(t) - \frac{d(t)}{L}\right)}{W} \tag{4}$$

The angular velocity ω can be expressed as the time derivative of α as in (5).

$$\dot{\alpha}(t) = \frac{v(t)\left(-\alpha(t) - \frac{d(t)}{L}\right)}{W} \tag{5}$$

The time derivative of the lateral deviation $\dot{d}$, is expressed as the product of the robot's velocity v and the sine component of its relative angle to the lane α, as shown in (6).

$$\dot{d(t)} = v(t)\sin\alpha(t) \tag{6}$$

Since the relative angle to lane α is small, (6) is rewritten as $\dot{d(t)} \approx v(t)\alpha(t)$. Expressing (5) in terms of d gives (7), and a transformation gives (8).

$$\frac{\ddot{d}(t)}{v(t)} = \frac{v(t)\left(-\frac{\dot{d}(t)}{v(t)} - \frac{d(t)}{L}\right)}{W} \tag{7}$$

$$W\ddot{d}(t) + v(t)\dot{d}(t) + \frac{v^2 d}{L} = 0 \tag{8}$$

The block diagram representation of this model is shown in Fig. 12. The transfer function derived from this model is given in equation (9).

$$G(s) = \frac{v^2 L}{WLs^2 + vLs + v^2} \tag{9}$$

The poles of the transfer function in equation (9) are real and negative when $L \geq 4W$ and $v > 0, W > 0$. In this robot, $W = 0.65$, $L = 5$ and $v > 0$ are satisfied and an oscillaton is not observed.

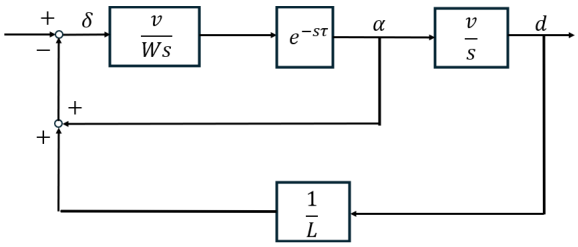

Fig. 13: Block Diagram of Robot Kinematics and Pure Pursuit Control Considering Time Delay.

B. Model with Time Delay

In order to elucidate an oscillation that could not be explained by the linear approximation model in the previous section, a model incorporating dead time is introduced in this section and its stability is analyzed. The block diagram of the system considering dead time is shown in Fig. 13. The transfer function in this case is expressed as shown in (10).

$$G(s) = \frac{v^2 L e^{-s\tau}}{WLs^2 + (vLs + v^2)e^{-s\tau}} \tag{10}$$

To examine the stability of the system, it is necessary to find the solution of (11), which makes the denominator of (10) equal to zero. Equation (11) is not algebraically solvable because it contains $e^{-s\tau}$, an infinite order polynomial of s. Therefore, in this study, the poles of the characteristic equation were obtained numerically using Python's `scipy.optimize.root` function.

$$WLs^2 + (vLs + v^2)e^{-s\tau} = 0 \tag{11}$$

Equation (11) was solved at 8 m/s, the speed at which an oscillation occurred and the vehicle left the running lane. $W = 0.65$ m, $L = 5$ m, according to the experimental conditions. The result is shown in Fig. 14. It is observed that an oscillation occurs when the dead time reaches 0.032 s, as a result of the imaginary part appearing. Furthermore, we confirmed the system becomes unstable, with the real part turning positive, when the dead time reaches 0.12 s. Specifically, at a dead time of 0.12 s, the pole of (11) is $s = 0.203 \pm 12.140j$. From this, the oscillation frequency can be calculated by dividing the imaginary part by 2π, yielding approximately 1.932 Hz.

C. Estimation of Dead Time from Experimental Data

The actual steering angle δ data was used to detect local peaks and valleys using the Python function `scipy.signal.find_peaks`, and the period and total amplitude were calculated. The result for a velocity of 8 m/s is shown in Fig. 14. The frequency was approximately 1.155 Hz. This measured frequency of 1.155 Hz differs by approximately 0.8 Hz from the 1.932 Hz frequency at which (11) predicted divergence. We calculated the dead time that minimized the difference between the frequency obtained from the model for various dead times and the experimentally observed frequency. The frequency of 1.174 Hz obtained for the dead time of 0.23 s is closest to the experimental value.

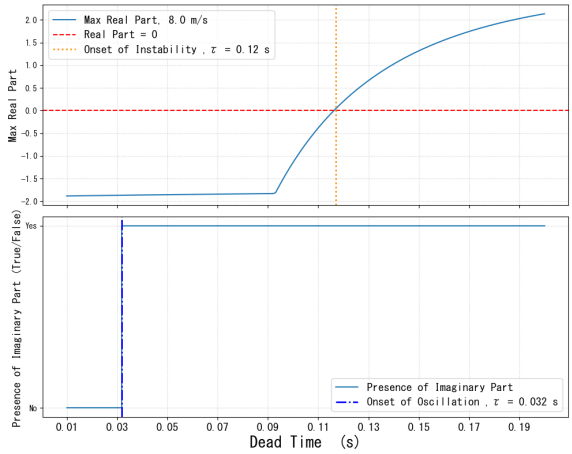

Fig. 14: Dead time and system response change at 8 m/s.

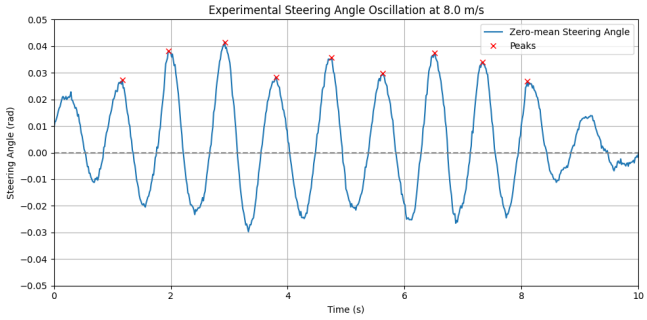

Fig. 15: Actual steering angle data at a speed of 8 m/s.

Furthermore, we investigated the system's response at a velocity of 6 m/s, assuming the dead time of 0.23 s. The poles of (11) are $s = 1.340 \pm 6.962j$ and the real part is positive, so the system is expected to diverge. However, it has been confirmed that in actual driving, the vehicle is able to drive without deviating from the running lane. This may be due to the influence of nonlinear factors such as friction between the robot and the ground and wind, which could not be considered in this model.

D. Discussion

By comparing the estimated and experimental frequencies, the dead time of approximately 0.23 s was inferred. In the model in Fig. 13, we assume that the dead time $e^{-s\tau}$ is placed between the blocks v/Ws and v/s. Possible sources of dead time include camera cycles and the time required to recognize running lanes. There are also other possible causes, such as the time required for arithmetic processing and communication time when giving commands to the steering motors, it is necessary to consider a model that takes delays into account in addition to Fig. 13.

To suppress the oscillation, it is necessary to shorten or cancel the dead time. When traveling at 8 m/sec, dead time should be reduced to about 0.03 s. Since it already takes more than 0.03 s in communication delay to command the motor, it is difficult to reduce dead time to the point where meandering is prevented.

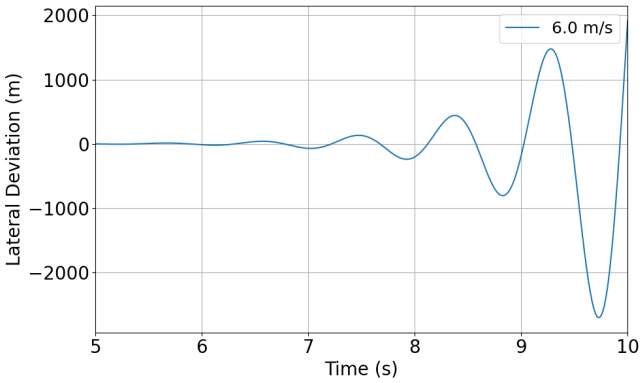

Fig. 16: Response analysis of a pole at a velocity of 6 m/s.

Therefore, a predictive control strategy in which control is performed using the future state of the system as input for dead time is considered to be effective. By introducing state prediction control that takes into account dead time and model predictive control (MPC) with optimization, stability at high speeds can be improved. In the future, these predictive control methods will be implemented and verified.

V. CONCLUSIONS

This study presented the development of a pacemaking runner robot for long-distance running. The driving control method uses a camera to detect the white lines on a track and recognize the driving lane, and calculates the steering angle using the Pure Pursuit method. To address the problem of lane detection failures caused by camera overexposure or occlusion of lane markings by athletes, we developed a method to detect not only the running lane markings but also the adjacent ones, thereby solving these issues.

As a backup to camera-based driving, we confirmed that 3D LiDAR can be used for self-position estimation while driving. 3D LiDAR could be used for obstacle avoidance or to change the pace of the robot to match the pace of the athlete. The robot was also equipped with a backup system that enables switching between camera and 3D LiDAR systems depending on the situation.

An oscillation at 8 m/s could be reproduced by a system that takes into account dead time. When reproduced, the system was found to have a dead time of approximately 0.23 s, which is a factor that causes meandering and deviation from the running lane. The model, which incorporates dead time, predicted that oscillation begins when dead time reaches 0.032 s at 8 m/s Considering that the communication delay to the motor already exceeds 0.03 s, the difficulty of simply reducing this time is highlighted. Therefore, future research will implement predictive control, such as state prediction control and model prediction control, to compensate for the delay and enable more stable high-speed driving. This robot is expected to be further developed and will be beneficial to long distance runners.

REFERENCES

[1] A. Kurosaki, M. Ae, et al.: ”Changes in running motion types of male long-distance runners in 10,000-m races”, Bulletin of Nippon Sport Science University, vol. 52, pp. 1089-1101, 2023 (in Japanese)

[2] S. Oda: Shintai Undo ni Okeru Migi to Hidari (Right and Left in Bodily Motion), p. 240, Kyoto University Press, (1998) (in Japanese)

[3] Taylor, J., Atkinson, G., and Best, R. (2021). Paced to perfection: Exploring the potential impact of WaveLight Technology in athletics. The Sport and Exercise Scientist, 68(Summer), 8-9.

[4] Puma Beatbot, `https://www.areaofeffect.io/projects/beatbot`

[5] T. Ban, J. Sugita, H. Date, T. Ohya, and Y. Enomoto: ”A lane-following robot for pacemaking of middle and long-distance runners”, Proceedings of the Annual Conference of the Robotics Society of Japan, 2023 (in Japanese)

[6] J. Sugita, Y. Enomoto, and H. Date: ”Ensuring the Safety of the Escort Runner Robot on an Athletic Track -Accuracy Verification of Localization Using a 3D Point Cloud Map-”, Proceedings of the Robotics and Mechatronics Conference 2024, The Japan Society of Mechanical Engineers, 2024 (in Japanese)

[7] Reid, T. G., Houts, S. E., Cammarata, R., Mills, G., Agarwal, S., Vora, A., and Pandey, G: ”Localization requirements for autonomous vehicles. arXiv preprint arXiv:1906.01061, (2019).

[8] T. Ban, H. Date, Y. Enomoto, and T. Ohya: ”Accompaniment Robot Assists Mid-Distance Runners in Training-Detection of running lanes under direct sunlight-”, Proceedings of the 22nd SICE System Integration Division Annual Conference, 2020, pp. 1081-1084 (in Japanese)

[9] R. Song, H. Chen, Y. Xu, and R. Klette: ”Lane detection algorithm based on geometric moment sampling”, Scientia Sinica Informationis, Vol. 47, No. 4, 2017 (in Chinese).

[10] Snider, Jarrod M. ”Automatic steering methods for autonomous automobile path tracking.” Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08 (2009).

[11] Englot, B. Meyers, D. Ratti, C. Rus, D. Shan, T. and Wang, W. “LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping ” ,IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),pp.5135–5142,2020.

[12] Koide, K. Menegatti, E. and Miura, J. “A portable three-dimensional LIDARbased system for long-term and widearea people behavior measurement ”, International Journal of Ad- vanced Robotic Systems, 2019.