2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

LiDAR-based Robot Transplanter

Masaki Asano[1] and Takanori Fukao[2]

***Abstract*— In Japan, labor shortage of agriculture is becoming increasingly severe due to the lack of farmers and aging. Therefore, the development of automation of vegetable production such as transplanting, harvesting and transporting is required. In this paper, a self-localization method by using LiDAR and a robust control method of a transplanter are proposed for accurate transplanting. In this system, the path of transplanter is generated by using 3D point cloud data, and the transplanting part follows it and plant seedlings of cabbage accurately. Path generation is performed considering vehicle tilt in the roll direction depending on the environment of grooves. An accurate calculation of lateral and angular position of the transplanting part is also proposed. For path following control, sliding-mode control and inverse optimal control are applied to transplanter. The experimental results demonstrated the effectiveness of these proposed methods and problems we have to tackle on. Basically, it was possible to perform automated transplanting accurately, but there was an occasional problem of offset error from 0. It was confirmed that inverse optimal control is superior to sliding-mode control and is more robust to environmental changes.**

I. Introduction

In Japan, labor shortage of agriculture is becoming increasingly severe due to the the lack of farmers and an aging population. The number of people working in agriculture is declining, and the number of people aged 65 and over is increasing every year, accounting for nearly 70% of the agricultural population [1]. Automation of agricultural machinery is expected to offset the labor shortage by reducing the amount of labor required for agricultural work. To realize this automation, multiple studies have been conducted on the automatic harvesting of cabbage, one of the most popular outdoor vegetables in terms of both production and consumption [2]. Research on path planning for multiple transport vehicles which carry containers loaded with harvested cabbages was also conducted.

In addition to harvesting and transportation of cabbage, we are working on automated transplanting of cabbage seedlings. A series of automated cabbage production processes is aimed. Seedlings must be planted correctly in the center of the ridges in order to prevent the fruits of cabbages from sticking out from the ridges and falling over after growing. In automated transplanting, the tolerance for error from the center of the ridge is $\pm 2\,\mathrm{cm}$. Automated transplanting methods using RTK-GNSS already exist [3] and are used for transplanting seedlings of rice, onion etc. In these methods, pre-made ridge routes and RTK-GNSS information are used for steering control. However, the horizontal positioning accuracy is $2\,\mathrm{cm}$ to $3\ \mathrm{cm}$, and automated control may be disrupted due to the temporary instability of GNSS reception while driving for a long time. When this happens, the ridges may collapse and proper transplantation may not be possible. A method of automated transplanting that is more accurate than the one using RTK-GNSS is needed to prevent this problem and achieve stable automated transplanting.

Fig. 1. Transplanter and LiDAR attached to the top part. Transplanting part is located at the rear of the vehicle.

In this paper, a self-localization method by using LiDAR and a robust control method of a transplanter are proposed for accurate transplanting. A LiDAR which is possible to obtain a 3D high-density point cloud was used. This LiDAR was attached to the top of the transplanter as shown in Fig. 1. Proposed methods enable automated driving by generating and following a path in real time using the point cloud data. This system can also be used to transplant seedlings of crops other than cabbage. In path generation, target ridges are detected, and the effect of the transplanter being tilted in the roll direction with respect to the top surface of the ridge is taken into consideration. For path following control, sliding-mode control and inverse optimal control were applied and implemented to the transplanter. Experiments using transplanter were conducted and the effectiveness of proposed methods was confirmed.

The main contributions of this paper for automated transplanting are as follows.

- Design a path considering vehicle tilt.
- Propose an accurate calculation of lateral and angular position of the transplanting part.

*This work was not supported by any organization

[1]Masaki Asano is with the Department of Mechano-Informatics at the Graduate School of Information Science and Technology, The University of Tokyo, Japan asano@ynl.t.u-tokyo.ac.jp

[2]Takanori Fukao is with the Department of Mechano-Informatics at the Graduate School of Information Science and Technology, The University of Tokyo, Japan

- Use of sliding-mode control and inverse optimal control for path following, and comparison of their performance.

The rest of the paper comprises the following sections. First, section 2 provides features of the automated transplanting system. Section 3 contains a description of the conducted experiment. Finally, a conclusion is presented in section 4.

II. Automated Transplanting System

This section describes the automated transplanting system. Overall structure is shown in Fig. 2. Firstly, ridge detection is conducted using the point cloud data from 3D-LiDAR. After that, path of the transplanter is generated using the information of detected ridges. Path following control is conducted using the path generated. Then, the transplanter is possible to run so that the seedlings can be planted in the center of the ridge.

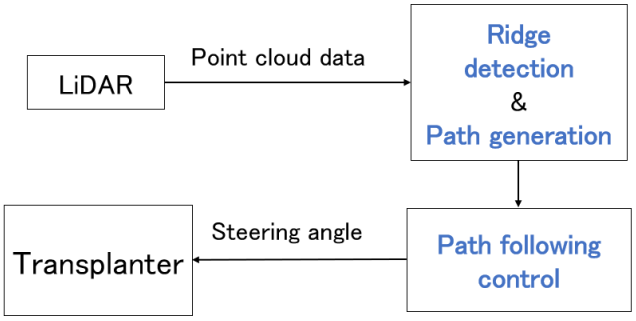

Fig. 2. System architecture

A. Ridge detection and path generation

The flow of path generation is shown in Fig. 3 In ridge detection, 2 ridges are detected for transplanting. Random Sample Consensus (RANSAC) [4] is used for the detection. This method is sufficiently robust to outliers to estimate the plane model ($ax + by + cz + d = 0$). The top surface of ridge is detected by RANSAC, and transformation is applied to the detected point cloud. Then, the inner edges of the ridges are detected and the intermediate line between them is determined as the path of the transplanter. The outside of the ridge tends to collapse due to factors such as tires colliding with each other when transplanting to the adjacent ridge, which may result in unstable edges. On the other hand, the inside of the ridges is less likely to collapse and the edges tend to be more stable. Thus, the inner edges are used for path generation. Details of point cloud transformation and path generation using ridge edges are described below.

1) Point cloud transformation considering the inclination of the transplanter: The LiDAR is mounted diagonally downward to the ground to get the point cloud data of ridges. The point cloud on the top surface of the ridge does not coincide with the XY-plane. Since the top surface of the ridge is parallel to the ground, the point cloud needs to be rotated to match the XY-plane to get the path of the transplanter. Basically, it is considered enough to rotate the point cloud by the fixed pitch angle of the LiDAR mounted on the transplanter. However, transplanter often tilts in the roll direction relative to the ridge surface depending on the groove environment as shown in Fig. 4. This causes the point cloud to rotate in the yaw direction as well. It is considered that the path ends up being at a diagonal angle compared to the correct path as shown in Fig. 5. Thus, it is necessary to rotate the point cloud by considering not only the pitch angle but also the roll angle of the top surface.

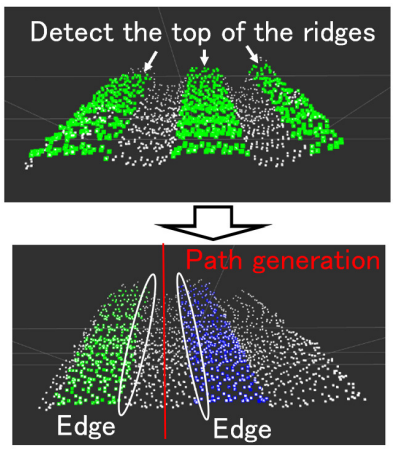

Fig. 3. Flow of path generation. Top surface detection is conducted by using RANSAC. Finally, the path is generated using Edge detection.

Fig. 4. Transplanter often tilts in the roll direction relative to the ridge surface depending on the groove environment.

We used the Euler angles to derive the pitch and roll angle for the transformation. When the Euler angles corresponding to the (x, y, z) axes are (α, β, γ), calculating the rotation matrix that adds rotation in the order of $x \rightarrow y \rightarrow z$ is as follows:

$$\begin{pmatrix} \vec{e}_X \\ \vec{e}_Y \\ \vec{e}_Z \end{pmatrix} = R_{xyz} \begin{pmatrix} \vec{e}_x \\ \vec{e}_y \\ \vec{e}_z \end{pmatrix}, \tag{1}$$

$$R_{xyz} = \begin{pmatrix} C_\beta C_\gamma & -C_\beta S_\gamma & S_\beta \\ C_\beta S_\gamma + C_\gamma S_\alpha S_\beta & C_\alpha C_\gamma - S_\alpha S_\beta S_\gamma & -C_\beta S_\alpha \\ S_\alpha S_\gamma - S_\alpha C_\gamma S_\beta & S_\gamma S_\alpha + S_\alpha S_\beta S_\gamma & C_\alpha C_\beta \end{pmatrix}. \tag{2}$$

$C_\theta = \cos\theta$ and $S_\theta = \sin\theta$. When $[e_X, e_Y, e_Z]^T = [0, 0, 1]^T$ and $[e_x, e_y, e_z]^T = [a, b, c]^T$, the following formu-

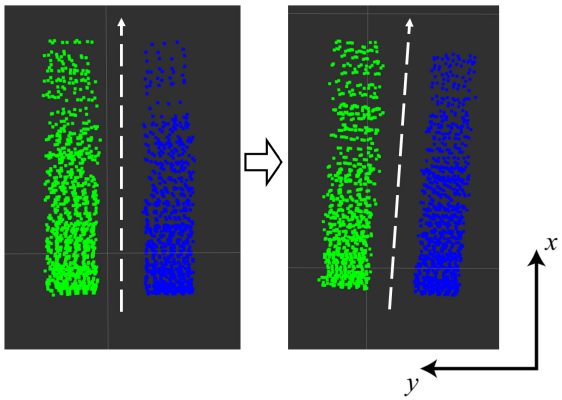

Fig. 5. Point cloud of upper surface also tilts affected by the roll direction tilt.

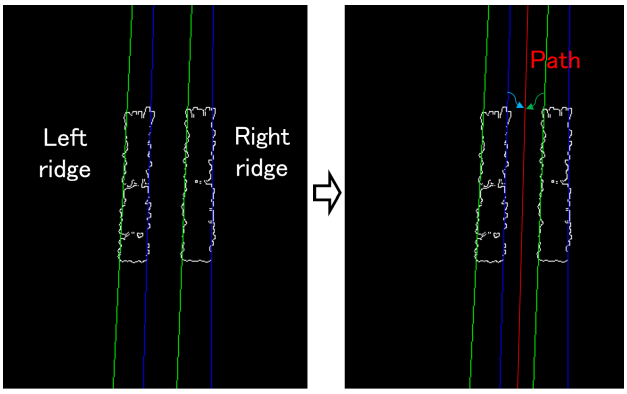

Fig. 6. Path generation using Canny edge detector. Green and blue lines are detected edges, and red line is the path generated for transplanter.

las are derived.

$$a = \sin\beta, \tag{3}$$

$$b = -\cos\beta\sin\alpha, \tag{4}$$

$$c = \cos\alpha\cos\beta. \tag{5}$$

α and β are derived as Eq. (6) and Eq. (7). α corresponds to roll angle and β corresponds to pitch angle.

$$\alpha = \tan^{-1}(-\frac{b}{c}), \tag{6}$$

$$\beta = \sin^{-1} a. \tag{7}$$

Then, by transforming the point cloud of the top surface of the ridges detected in the order of pitch→roll angle, it is possible to obtain the desired point cloud parallel to XY-plane. We can get a, b and c from the result of RANSAC as normal vector.

2) Path generation using Canny edge detector: After obtaining the point cloud of upper surface of the ridge horizontal to the XY-plane by Euler angle transformation, the path of the transplanter is generated using Canny edge detector [5]. This method is one of the strong edge detection methods. Path generation is shown in Fig. 6. Each points of the cloud are drawn in white on the black image and process it by using OpenCV [6]. Firstly, find the edges of the ridges using Canny edge detector. White parts are detected edges of the ridges. Then line segments are detected by RANSAC using these edges, and two edges inside the ridges are extracted. In this case, the blue line is for left ridge, and the green line is for the right ridge. Then, the equation of the straight line of left edge (Eq. (8)) and right edge (Eq. (9)) are obtained. Finally, the equation of the path of the transplanter is derived as Eq. (10).

$$a_l x + b_l y + c_l = 0, \tag{8}$$

$$a_r x + b_r y + c_r = 0, \tag{9}$$

$$(a_l + a_r)x + (b_l + b_r)y + (c_l + c_r) = 0. \tag{10}$$

B. Path following control of transplanter

In order to achieve a transplantation accuracy of $\pm 2\,\mathrm{cm}$, sliding-mode control and inverse optimal control are applied for path following control. Robustness against softness and slope of the ground and slippage can be expected from these control methods.

1) Vehicle Kinematics: We define the velocity v, the angular velocity ω and yaw angle θ of the transplanter. Then, we obtain a kinematic model of vehicle as Eq. (11)

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{11}$$

The same applies to the reference vehicle. e_2 and e_3 in Fig. 7 represent the lateral and angular errors between the axle center of the reference vehicle and that of the real vehicle. These states are expressed as follows:

$$\frac{d}{dt}\begin{bmatrix} e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} v\sin e_3 \\ \omega - \omega_r \end{bmatrix}. \tag{12}$$

We set $e_{2_{LiDAR}}$ as lateral error calculated from the generated path. In the path generation, the point cloud of the top surface was transformed in roll direction, and this affects the actual e_2. e_2 can be expressed as following,

$$e_2 = e_{2_{LiDAR}} - H\sin(\theta_{roll}). \tag{13}$$

H is height from the top of the ridge to the center of LiDAR as shown in Fig. 8. The central part of tha transplanting part should be controlled to follow the path generated, and the lateral error $e_2'$is shown as Eq. (14).

$$e_2' = e_2 - L\sin(e_3). \tag{14}$$

L is the distance from the center of the rear wheel axis to the center of the transplanting part. the angular error e_3' is equal to e_3. In the control performed by automated driving, when e_2 and e_3 can be converged to 0, e_2' and e_3' can also be converged to 0.

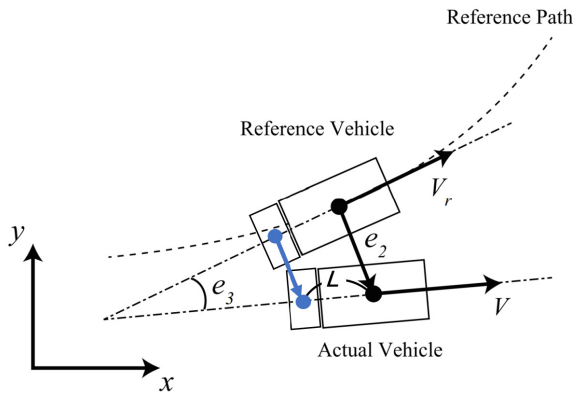

Fig. 7. Path following system. The blue arrow shows lateral error of the transplanting part.

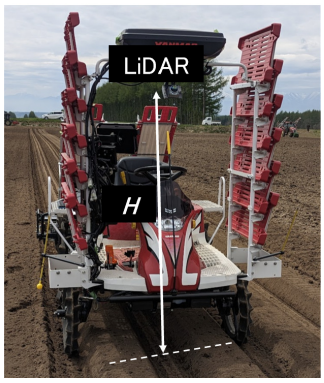

Fig. 8. Positional relationship between LiDAR and ridges. Correcting the effect of rotating the point cloud in the roll direction is necessary.

2) Sliding-mode control: Sliding-mode control [7] is a method of attaining control by sliding the state towards the equilibrium point while constraining it to the switching plane [8]. It is more robust to additive disturbances and model uncertainty than nonlinear control based on the Lyapunov stability theorem [9].

We perform a state transformation as $\xi = \sin e_3$. The model after the state transformation is as follows:

$$\frac{d}{dt}\begin{bmatrix} e_2 \\ \xi \end{bmatrix} = \begin{bmatrix} v\xi \\ (\omega - \omega_r)\sqrt{1-\xi^2} \end{bmatrix}. \tag{15}$$

The control law for the convergence of e_2 and e_3 to 0 is expressed as follows:

$$s = \xi + K_s e_2. \tag{16}$$

K_s represents a positive constant. We define a Lyapunov function V_s as following:

$$V_s = \frac{1}{2}s^2. \tag{17}$$

By computing a derivative of Eq. (13) with respect to time t,

$$\begin{aligned} \dot{V} &= s\dot{s} && (18)\\ &= s(K_s V\xi - \omega_r\sqrt{1-\xi^2} + \omega\sqrt{1-\xi^2}). && (19) \end{aligned}$$

Then, we define an input ω as shown in Eq. (14)

$$\omega = \omega_r - \frac{K_s v\xi + \eta f(s)}{\sqrt{1-\xi^2}}. \tag{20}$$

η is a positive constant, and $f(s) = \mathrm{sgn}(s)$, but to avoid chattering, we set $f(s)$ as follows:

$$f(s) = \frac{s}{|s| + \delta}. \tag{21}$$

δ is an infinitesimal positive constant. This system converges to zero with finite time when reaching to the sliding surface $s = 0$.

3) Inverse optimal control: Inverse optimal control [10] is also robust to additive disturbances and model uncertainty. Design an inverse optimal control system. First, perform the following coordinate transformation.

$$h(z) = A\tan^{-1}(Bz). \tag{22}$$

A and B are positive constants and are determined by considering the relationship between the size of $z = e_2 v$ and $h(z)$.

$$h'(z) = \frac{dh}{dz} = \frac{AB}{1 + (Bz)^2}, \tag{23}$$

$$\tilde{e}_3 = e_3 + h(e_2 v). \tag{24}$$

Then, the equation of state is as follows:

$$\frac{d}{dt}\begin{bmatrix} e_2 \\ \tilde{e}_3 \end{bmatrix} = \begin{bmatrix} v\sin(\tilde{e}_3 - h) \\ -\omega_r + h'(v^2\sin(\tilde{e}_3 - h) + e_2\dot{v}) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\omega \tag{25}$$

$$\equiv f + g\omega. \tag{26}$$

Here, define the following function:

$$V(e_2, \tilde{e}_3) = \frac{1}{2}K_2 e_2^2 + \frac{1}{2}K_3\tilde{e}_3^2. \tag{27}$$

This function $V(x)$ is CLF (Control Lyapunov Function). The control input follows the Sontag format [11] and can be shown as follows. c_0 is a constant.

$$\begin{aligned} a &= L_f V && (28)\\ &= K_2 e_2 v\sin(\tilde{e}_3 - h) + && (29)\\ &\quad K_3\tilde{e}_3(-\omega_r + h'(v^2\sin(\tilde{e}_3 - h) + e_2\dot{v})), \\ b^T &= L_g V = K_3\tilde{e}_3, && (30) \end{aligned}$$

$$\omega = \begin{cases} \omega_r - \left(c_0 + \frac{a + \sqrt{a^2 + (b^T b)^2}}{b^T b}\right)b & b \neq 0, \\ \omega_r & b = 0. \end{cases} \tag{31}$$

For steering, sliding-mode control is applied. We define a sliding surface as follows:

$$\sigma = \omega - u(x). \tag{32}$$

ω is angular velocity of ego vehicle and $u(x)$ is input. Then, it is needed to satisfy

$$\begin{cases} \dot{\sigma} > 0, \sigma < 0, \\ \dot{\sigma} < 0, \sigma > 0. \end{cases} \tag{33}$$

The input steering angle is given by following equation:

$$\delta = -K_s \frac{\sigma}{|\sigma|}. \tag{34}$$

Then, practical steering angle is as follows:

$$\delta = \delta_{ff} - K_s \frac{\sigma}{|\sigma| + \epsilon}. \tag{35}$$

where δ_{ff} is feedforward term and ϵ is a small positive constant to prevent chattering.

III. Experiments

We implemented the proposed method on a transplanter and conducted experiments to confirm its effectiveness. The transplanter was made by Yanmar Co., Ltd., and the steering unit and transmission unit were modified by Yanmar Agribusiness Co., Ltd.. For implementing a control program written in C++, we used a computer embedded with an Intel Core i7-1260P CPU processor with a maximum clock frequency of 4.70 GHz. We used a LiDAR as external sensor for path generation and self-localization in automated driving. Livox Mid-360 was used as LiDAR.

A. Effectiveness of tilt correction in the vehicle roll direction

We confirmed the effectiveness of the path generation method that takes into account the inclination of the transplanter in the roll direction relative to the top surface of the ridge. On the transplanter, RTK-GNSS receiver is mounted for test. Since the ridges are created automatically by a tractor and the path is saved, this information can be used for comparison. We compared the difference of e_3 between one with roll and pitch correction and one with only fixed value of pitch (60°) correction. e_3 is the same as the slope of the path and this data was collected when automated driving is being conducted. In Fig. 9, there are sections where there is a discrepancy with the e_3 from GNSS such as around 175s, 250s, 260s etc. in the graph. As Fig. 9 and Fig. 10 show, the difference between the e_3 of the proposed method and GNSS is smaller than that of the method only using pitch correction. From this, we could confirm that the path generation method that performs roll and pitch correction is working.

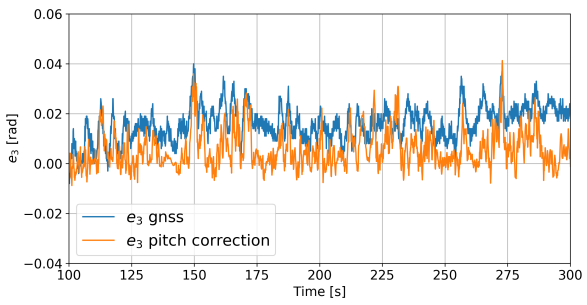

Fig. 9. Comparison between e_3 given by the method only using pitch correction and e_3 from GNSS.

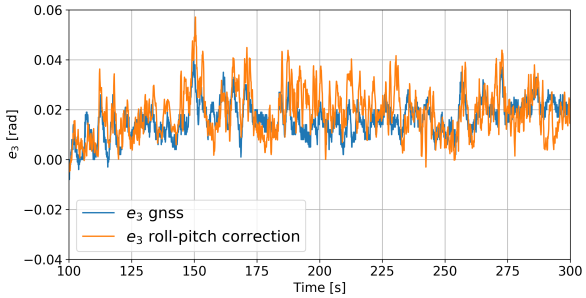

Fig. 10. Comparison of e_3 between proposed method and GNSS.

B. Automated driving by path following control

We implemented sliding-mode control and inverse optimal control on the transplanter, and conducted an autonomous driving experiments. The effectiveness of these control methods was evaluated running 100 m along the ridge. We also actually transplanted seedlings using sliding-mode control. The speed of the transplanter was 0.3 m/s. Fig. 11 shows e'_2 and e'_3 when automated driving was conducted. It can be seen that e_2 basically converges to about ±4 cm around 0 although it increased momentarily. Fig. 12 shows how to

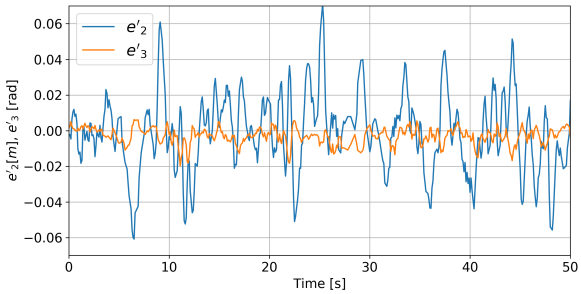

Fig. 11. Lateral error e'_2 and angular error e'_3 of sliding-mode control.

measure the error in the transplanting location of seedlings. Measure from the inside edge of the ridge to the planting point on each side(D_{left}, D_{right}), and half of the difference can be determined as the planting error. This error E can be expressed as

$$E = \frac{|D_{left} - D_{right}|}{2}. \tag{36}$$

Over a total distance of 100 m, we measured 3 measurements every 10 m to account for a total of 30 errors. The average error was 1.5 cm, achieving the target accuracy 2 cm running in different locations 10 times, total distance 1000 m. Fig. 13 shows the errors of autonomous driving by inverse optimal control. It can also be seen that control is achieved with an error that converges around 0, that is similar to sliding mode control.

However, there were cases where it was difficult for the error to converge, and offset from 0 occurred. Fig. 14 shows the error of using sliding-mode control. In this graph, e_2 is vibrating around 0.03 m instead of 0. It is possible that the

Fig. 12. Measurement of the error of transplanting on the ridge.Red two-way Arrow shows an example of left ridge.

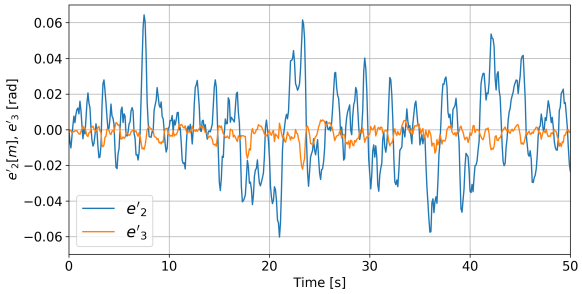

Fig. 13. Lateral error e'_2 and angular error e'_3 of inverse optimal control.

environments such as depth of the grooves that the left and right tires run on are different, making the vehicle tilt and slide. Fig. 15 shows the error of using inverse optimal control traveling on the same ridge. In this graph, the offset error from 0 can also be seen, but it is better than the sliding-mode control. The result of comparing the distribution of e_2 is shown in Fig. 16. This represents the mean squared error every 10 seconds. It is confirmed that inverse optimal control is more robust against environmental changes such as slippage than sliding mode. Thus inverse optimal control is superior to sliding-mode control for automated driving in transplanting. There was a problem with the front tires being thin and causing them to slip, but this is mainly due to mechanical factors and needs to be resolved when greater accuracy is needed.

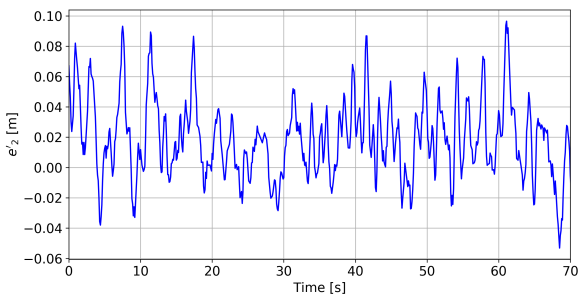

Fig. 14. Lateral error e'_2 of sliding-mode control. e'_2 doesn't converge to 0 and offset error from 0 can be seen.

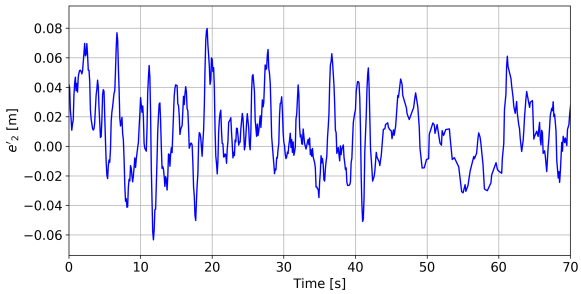

Fig. 15. Lateral error e'_2 of Inverse optimal control. e'_2 has difficulty converging to 0 and the value is slightly biased towards the positive side.

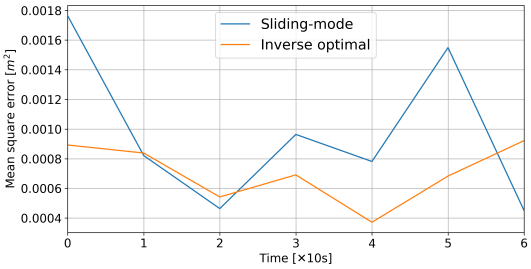

Fig. 16. Mean square error of e'_2 every 10s. Error of inverse optimal control is overall smaller than that of sliding-mode control.

IV. Conclusions

We described an automated transplanting system using 3D-LiDAR. In this system, a path of transplanter is generated considering the transplanter being tilted in the roll direction with respect to the top surface of the ridge. We proposed an accurate calculation of lateral and angular position of the transplanting part and a control method of a transplanter for accurate transplanting. We also applied sliding-mode control and inverse optimal control to the transplanter. The effectiveness of these proposed methods was confirmed through actual machine experiments. Automated transplanting could be performed within the target accuracy of $\pm 2\,\mathrm{cm}$. However, there was an occasional problem of offset error from 0 due to the slipping of the front tires of the transplanter. From the comparison results, it was confirmed that inverse optimal control is superior to sliding-mode control and is more robust to environmental changes.

Acknowledgment

This research was supported by development and improvement program of strategic smart agricultural technology grants (JPJ011397) from the Project of the Bio-oriented Technology Research Advancement Institution (BRAIN).

References

[1] Japan Statistical Yearbook 2021. Japan: Statistics Bureau, Ministry of Internal Affairs and Communications; 2021.

[2] M. Asano, K. Onishi and T. Fukao, Robust cabbage recognition and automatic harvesting under environmental changes. Advanced Robotics, vol. 37, pp.960-969, 2023.

[3] X. Yin, J. Du, D. Geng and C. Jin, Development of an automatically guided rice transplanter using RTK-GNSS and IMU. IFAC(International Federation of Automatic Control), pp.374-378, 2018.
[4] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM, vol. 24, no. 6, p. 381395, 1981.
[5] J. Canny, A Computational Approach to Edge Detection. IEEE Trans. Pattern Anal. Machine Intell, vol. 8, no. 6, pp. 679-698, 1986.
[6] G. Bradski, The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.
[7] T. Fukao, T. Aoki, T. Sugimachi, Y. Yamada, and H. Kawashima, Preceding Vehicle Following Based on Path Following Control for Platooning. 7th IFAC Symposium on Advances in Automotive Control, vol. 46, pp. 47-51, 2013.
[8] J. E. Slotineand W. Li, Applied nonlinear control. Prentice Hall, pp. 276-310, 1991.
[9] J. Yoshida, T. Sugimachi, T. Fukao, Y. Suzuki, and K. Aoki, Autonomous Driving of a Truck Based on Path Following Control. 10th Int. Symposium on Advanced Vehicle Control, 2010.
[10] T. Fukao, Inverse optimal tracking control of a nonholonmic mobile robot. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 1475-1480, 2004.
[11] E.D. Sontag, Mathematical control theory. Springer, 1998.