Research on Autonomous Driving Control of Electric Vehicle for Orchard Application

PhD thesis, major in agriculture

Graduate School of Agriculture, Hokkaido University

Yoshitomo Yamasaki

Table of contents

Chapter 1. Introduction

1.1. Fruits production in Japan

The Ministry of Agriculture, Forestry and Fisheries in Japan showed the top five fruit cultivation areas in Japan in 2023 were 37,300 ha for oranges, 35,900 ha for apples, 17,900 ha for persimmons, 17,600 ha for grapes, and 16,200 ha for chestnuts [1] (Fig. 1). Other citrus fruits cover an area of 23,600 ha, and if combined with mandarin oranges, they occupy approximately 60,000 ha. The orchard areas have been decreasing over the past 10 years. The decline rates for mandarin oranges, persimmons, and chestnuts from 2014 to 2023 were large at 17.8%, 18.3%, and 22.1%, respectively. For mandarin oranges, particularly, shrinking by 8,000 ha.

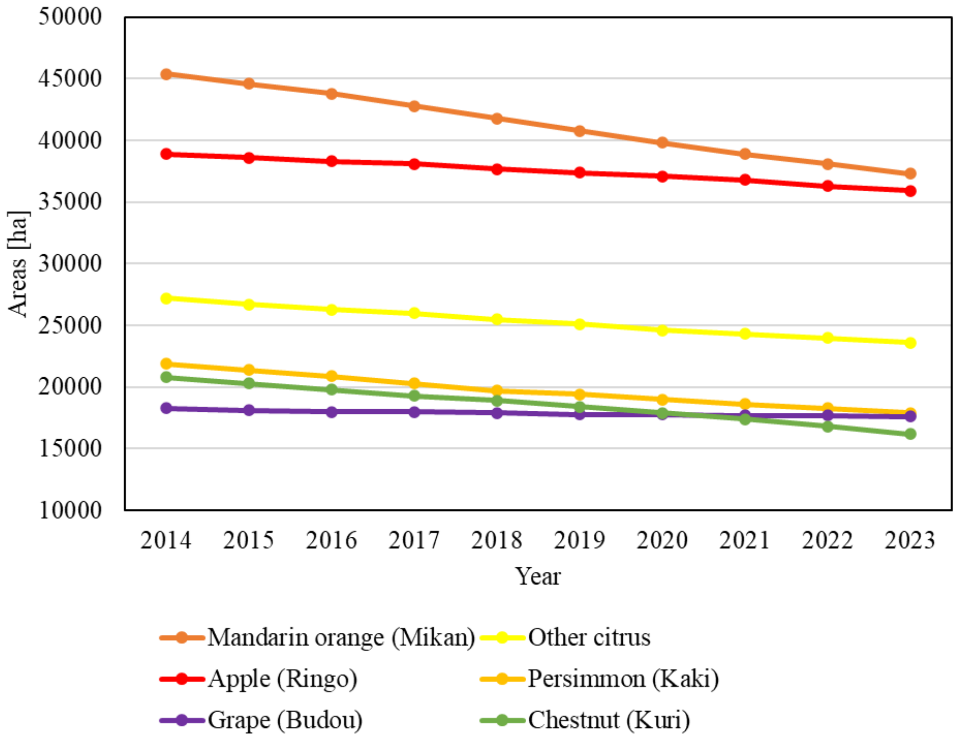


Fig. 1 Change in cropland areas for fruit cultivation (Brackets means Japanese name)

Fig. 2 shows domestic fruit consumption trends. The amount spent on fresh fruit by households with two or more people was 34,800 yen/year in 2012, while it was 36,400 yen/year in 2022 [2]. It appears to be slightly increasing. However, looking at trends in the wholesale price and quantity of fruit (Fig. 3), from 2014 to 2023, the price rose from 280 yen/kg to 401 yen/kg, while the quantity fell from 3,690 kt to 2,690 kt. Domestic wholesale volume decreased by 27% from 2014 to 2023, but domestic cultivation area decreased by an average of only 14% over 10-year period. This means that the yield per area has decreased, and the unit price increased. This may be due to the trend of growing high-priced fruits.

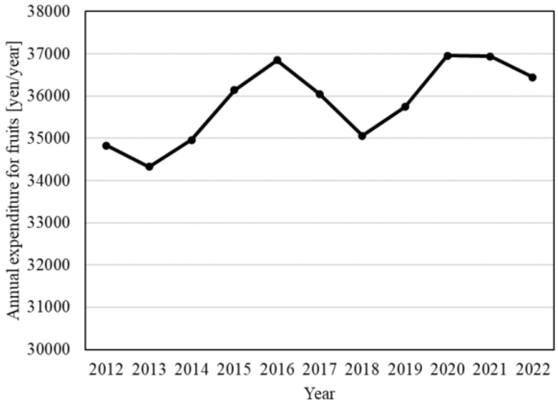


Fig. 2 Change in annual expenditure to fruits for each household

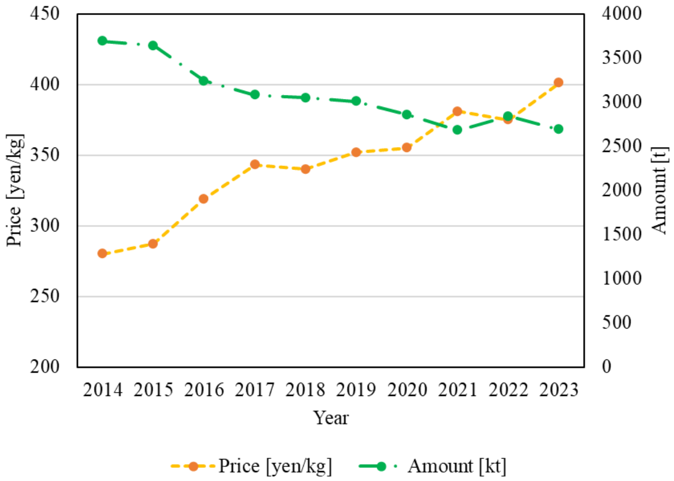


Fig. 3 Change in the amount and the wholesale price of fruits

As is shown in Fig. 4, you can see the same trend from export amounts. According to data from the Ministry of Agriculture, Forestry and Fisheries in Japan [3], the export price of fruits has been rising, from 13.3 billion yen in 2012 to 45.3 billion yen in 2020. In particular, the export price of grapes has increased more than 10 times from 0.4 billion yen to 4.1 billion yen. The unit price has also increased from 1,111 yen/kg to 2,407 yen/kg, indicating that the quality of the fruit has improved. This can be because high-quality varieties such as Shine Muscat emerged. Most of the export destinations are neighboring countries such as Taiwan, Hong Kong, or Singapore.

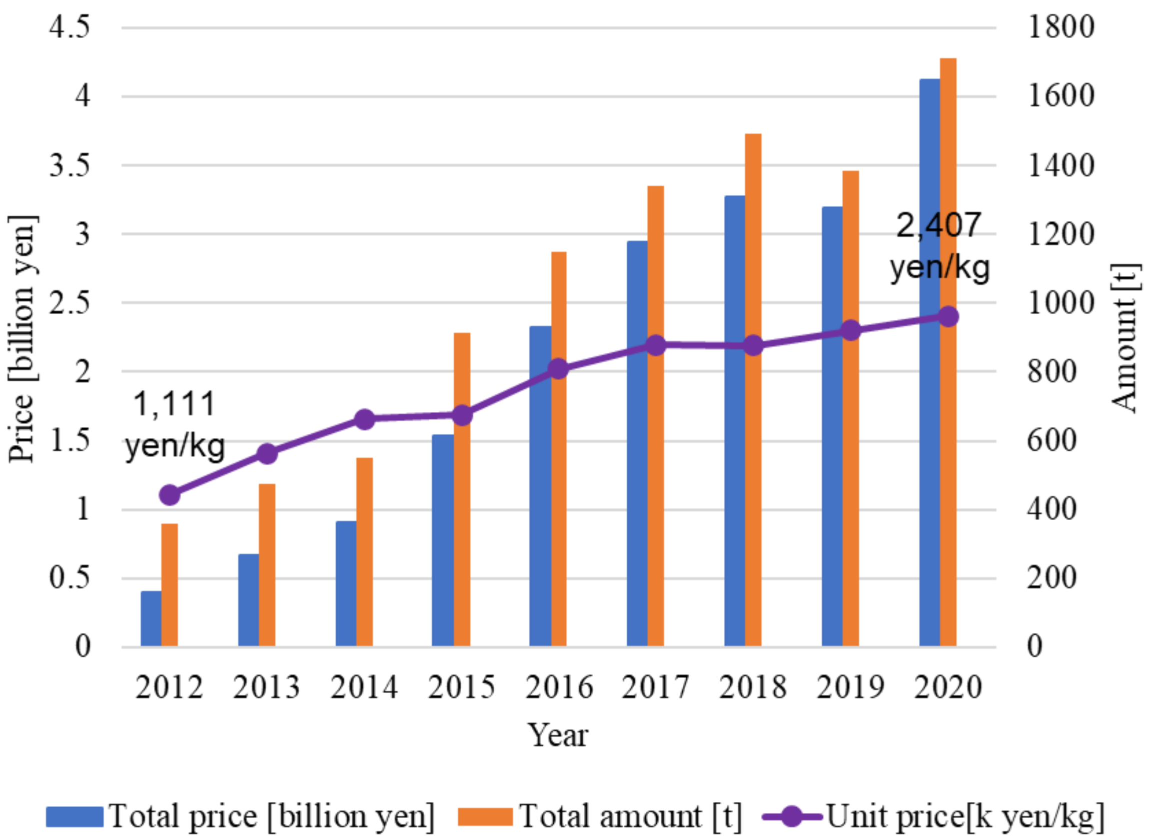


Fig. 4 Change in the prices and the amount of exported fruits

Consumption of fruit wine has been increasing in Japan. According to a 2019 National Tax Agency survey [4], the overall shipment volume of alcoholic beverages has been declined since 2000, while fruit alcoholic beverages have been rising. Domestic fruit wine has been also rising, with production volume increasing from 67,000 kiloliters in 2007 to 102,000 kiloliters in 2017, an increase of approximately 1.5 times. Also, the number of fruit wine manufacturing facilities has been increasing from 367 in 2015 to 413 in 2021 [5]. However, wines made in Japan still rely on imports for the raw materials such as fruit and fruit juice. Only 33% of wines are made from domestically produced ingredients. The main production areas for wine grapes in Japan are Yamanashi Prefecture, Nagano Prefecture, Hokkaido, and Yamagata Prefecture, and these four prefectures account for 84% of the total production. These regions are characterized by having a particularly high mountainous area compared to other regions in Japan, excluding Hokkaido. According to the Japan Meteorological Agency, "mountainous area" is not strictly defined as a place with many mountains and an antonym for flatland. Therefore, the index of mountainous area ratio is ambiguous, but for reference, the percentage of mountainous areas in the total area is 86% [6] in Yamanashi Prefecture, 84% [7] in Nagano Prefecture, 50% [8] in Hokkaido, and 72% [9] in Yamagata Prefecture. Note that these values vary greatly depending on the source of the data, so they are just reference values. However, the fruits of wine can be seen to be widely cultivated in mountainous areas. Mountainous areas generally have a small population, making it difficult to ensure a labor force. The area of fruit cultivation has been decreasing because many farmers quit cultivation due to the lack of the labor force. Therefore, ensuring labor force has been a serious issue.

1.2. Labor-saving tree shape

As mentioned above, enhancement of the workforce through mechanization is essential as the labor shortage is becoming serious, especially in rural areas. However, if we did not maintain fruit trees, they will spread out in three dimensions, making it difficult to introduce machinery. Thus, tree shaping is necessary so that machinery can fit into it. This is called a labor-saving tree shaping. In a regular orchard, the tree shaping is conducted to be easier for humans to manage trees. For example, grape trees need a trellis to support vines of them. There are two types of grape trellises: the vertical type and the horizontal type (Fig. 5). According to [5], in Japan, 63.3% of grapes for wine production are cultivated on horizontal trellises, and 36.7% are cultivated on vertical trellises. The horizontal type is popular because of its advantages such as easier management of fruits, higher yield per unit area, better ventilation for fruits, resulting in reducing diseases, and avoiding damages from typhoons. On the other hand, the vertical type of cultivation is preferable from the perspective of mechanization because it is easier for machines to move around the orchard and to operate trees because they have flat shapes. For these reasons, the vertical type of cultivation is widespread in grape cultivation around the world. In addition to grapes, some fruits and citrus fruits are also cultivated using the vertical trellis in Japan, but the area is still small.

Fig. 5 The vertical trellis and the horizontal trellis

Mandarin oranges, which are most cultivated in Japan, are generally cultivated in their natural form, but there is also a labor-saving tree shaping called the double-trunk cultivation (Fig. 6). This is a tree shape with two main trunks extending parallel to the row. The double-trunk form can create a plane of trees, like the vertical trellis. According to [10], the introduction of robot vehicles through the double-trunk cultivation has reduced work time by 16.7% for thinning, 15.2% for harvesting, and 40.9% for pruning. While pruning, it saved time significantly, because the double-trunk cultivation had a clear skeleton of trunks, so there was less time to think about which points should be cut.

Fig. 6 The double trunk cultivation of Mandarin oranges. Source: https://www.youtube.com/watch?v=GG8kHMH-iRA

The joint tree shape is a method of making the whole tree uniform by stretching the trunk parallel to the ridges and grafting it to an adjacent tree. It is widely applied to pears, apples, persimmons, etc. The joint V-shaped trellis, which is an application of the joint tree shape, has been attracting attention in recent years, and in the case of apples, it has been reported that the work time can be reduced by 50% [10].

Fig. 7 The joint V-shaped trellis of pears. Source: [10]

All labor-saving tree shapes have common characteristics, such as having a flat plane of branches to make it easy for automatic machines to work, and having easy-to-understand main trunks that require less consideration when pruning. These results suggest that adjusting the tree shape to the mechanization is necessary as well as creating machines to fit into the crops.

1.3. Autonomous navigation

Research on robot vehicles traveling in the labor-saving tree-shaped orchards as described above has been conducted, and many autonomous navigation methods also exist. The Global Navigation Satellite System (GNSS) has been widely used for robot vehicles in a field as a powerful self-positioning. However, because there are trees in orchards, many studies have developed autonomous navigation methods without the GNSS.

Ishiyama et al., 2022 developed an autonomous ground vehicle with a towed implement transporting harvested products and automating farming tasks in nine types of orchards [11]. In that study, two 2D-LiDARs were installed in the front for autonomous navigation. Between tree rows, the tree line was approximated by a straight line and the vehicle traveled according to the estimated center line. When turning, the vehicle acquiring point cloud data estimated its position by referring to a landmark map created in advance. In fact, the method of linearly approximating tree rows using the 2D-LiDAR has been developed since the late 2000s. Barawid et al., 2007 used the Hough transformation to approximate the tree rows to a straight line and estimate the center line as the traveling path [12]. When extracting a representative line from a point cloud, noise in the point cloud causes an error in the estimated straight line. That study performed noise filtering with a moving average. It was found that removing noise is required to improve accuracy when recognizing tree lines from point clouds using the 2D-LiDAR. Jones et al., 2019 compared autonomous navigation methods of the GNSS, a vision sensor, and the 2D-LiDAR for a kiwifruit orchard cultivated on horizontal trellises and suggested that the most appropriate method was to recognize trees using 3D-

LiDAR [13]. They determined the traveling path by making pairs of left and right trees in order and creating their center points.

Yanagisawa et al., 2021 developed an autonomous transport vehicle for orchards [14]. The ArUco markers were installed in the orchard, and the vehicle could travel by recognizing them with a vision sensor. The authors also developed an autonomous navigation method for orchards using vision sensors and the ArUco markers [15]. A convolutional neural network was used to learn the traveling path between tree rows in an image in a grape orchard to estimate a traveling path. When turning, the robot vehicle recognized the ArUco marker and entered the next row. In autonomous navigation using images, the color information of the surroundings changes due to the weather condition. Deep learning-based detection was found to be more robust than classic threshold classification. On the other hand, training data was required for each season because the appearance of trees changes with the seasons. Additionally, since the environment differs in each orchard, one of the challenges when using deep learning is to require a large amount of data.

Autonomous navigation methods without the GNSS basically focus on traveling in the tree rows. Although some studies attempted to turn using a landmark map and the ArUco markers, navigation on a headland or from a garage to an orchard is difficult without GNSS. Liu et al., 2024 proposed a method that combines the GNSS and the LiDAR [16]. Navigation between tree rows was performed using the LiDAR, while traveling to headlands and other areas was performed using the GNSS. Falco et al., 2019 conducted a performance evaluation of the Real-Time Kinematic GNSS (RTK-GNSS) in an orchard and verified how well the RTK-GNSS can perform in a harsh environment where trees exist [17]. In Chapter 4 of this research, we verified autonomous navigation in an orchard using the GNSS corrected by the PPP-RTK method, a type of the RTK-GNSS.

1.4. Electrification of farming machinery

As well as automation, the electrification of agricultural machinery has been focused on. In 2016, Europe adopted a set of emission control regulations for off-road vehicles called the European Stage V Non-Road Emission Standards (commonly known as StageV). This agreement suggested the emission limits for particulate matter (PM) mainly emitted by diesel engines, and applies to agricultural machinery, construction machinery, railways, ships, etc., with diesel engines of 19kW to 560kW. Due to the emission regulations, the development of electric agricultural machinery that does not emit gas has been promoted. Electric tractors have been developed, mainly by John Deere, the world's largest agricultural machinery manufacturer [18]. According to John Deere, electric tractors have fewer parts and are easier to maintain than internal combustion engine tractors. Thus, they surely will enter the tractor market in the future [19]. However, battery performance is currently a bottleneck. Electric tractors have an overwhelmingly shorter running time than internal combustion engine tractors, which is the most significant drawbacks. An Electric tractor supplied power via wires has also been developed so that charging is no longer a consideration [20]. Hybrid tractors replacing parts of the driving system with electric power such as the PTO have been developed. The Robot Fleets for Highly Effective Agriculture and Forestry Management (RHEA) project conducted in the EU in 2014 reduced emissions by approximately 50% by installing fuel cells and solar cells on a unmanned internal combustion engine tractor [21]. Researchers in Poland conducted a market survey on electric tractors [22]. According to the survey, it is difficult for current electric tractors to compete with existing diesel tractors unless they are equipped with a system that does not charge during work, and they are suitable for use in relatively small greenhouses or orchards.

1.5. Structure of this thesis

This research focused on autonomous driving control of an electric vehicle for orchard application. The critical points were development of an electric vehicle and autonomous navigation. Chapter 2 describes development of driving control in off-road environments for electrification. Chapter 3 describes autonomous navigation by the Quasi-Zenith Satellite System (QZSS). Chapter 4 considered determining the optimal route in the orchard, and Chapter 5 discussed how to turn in an orchard when a towed implement is attached.

Chapter 2. Development of driving control

2.1. Introduction

2.1.1. Demand for electrification

Hybrid electric vehicles or electric vehicles, which can reduce using fossil fuels, have been developed these days to mitigate an impact on the environment. This trend in the automotive sector is the same around agricultural vehicles. Some researchers conduct research on a hybrid tractor whose implement or driving force is driven by electric power, or an electric tractor (ET) that is totally driven by electric power ([23]). [21] converted an engine PTO of a small tractor to an electric one, and developed a hybrid robot tractor that works on spraying autonomously. They tested spraying in maize, wheat, and olive fields and concluded that the robot could reduce up to 50 % of gas emissions. [24] analyzed the economic cost and the amount of emissions of a solar assist plug-in hybrid electric tractor (SAPHT) on simulation. They estimated that we can reduce 14 tons of carbon dioxide in a year if conventional tractors are replaced with the SAPHT. Additionally, [25] made a prototype of an electric tractor: parts for driving of a 10 kW small tractor were replaced with electric parts. They concluded that energy consumption could be reduced to 70 % of the original engine tractor even though the new one was heavier than the original one due to motors. [26] estimated that replacing conventional diesel engine tractors with small robot ETs leads to a 29

– 33 % cost reduction. These studies verified that replacing engines with motors can reduce energy consumption and costs. In addition, [27] suggested an estimation method of energy consumption relating to working for electric farming machines. We can design a farm with a power generation system according to energy consumption if it can be estimated accurately. However, according to a survey by [22], 75 % of farmers said that they did not want to purchase an ET. The main reason was that they had trouble if the ET needed charging frequently and interrupted farming tasks. Hence, the research concluded that ETs should be applied to orchards or greenhouses that have smaller areas than typical crop fields so far.

2.1.2. Small farming robot powered by electricity

Small robot vehicles for farming, which are designed apart from conventional tractors, have been developed for orchards and vineyards. [28] developed a spraying robot for vineyards and greenhouses. The robot had two DC motors of 1 kW output, carried up to 200 kg, and traveled up to 3 km/h. [29] also developed a bigger spraying robot for orchards and vineyards. The robot could load 3500 kg and had four 7kW motors for each wheel to adapt bumpy fields. However, the robot of [29] had a diesel engine to add power to an electric pump and fan for spraying. Control would be complicated because there were small motors on four wheels in addition to an engine. In summary, a power train system for an electric farming machine has not been established. There are few developed systems that can generate power for driving force on a offroad and for farming implement at the same time by electricity. In addition, [30] summarized commercial small farming robots, but many of them were developed for sensing or weeding on a field. Therefore, they did not have enough power to spray or mow in orchards, which need power as much as a middle between tractors and small farming robots.

2.1.3. Hybrid system for automotive

This study paid attention to a hybrid system (HS) for automotive as a power train. The HS for automotive has been already developed and verified the quality. Thus, the development cost will be suppressed if the HS will be converted into a farming machine. For example, The Toyota Hybrid System (THS) has an engine and two motors. Those three shafts are linked by a planetary gear. The efficiency of fuel is optimized by controlling both an engine and two motors, which have different characteristics in terms of energy efficiency [31]. Regularly, a motor on the THS is used for driving and the other is for charging, but both two can be used for driving. Therefore, both working and traveling are achieved by installing a farming implement into an engine shaft, and driving it and a tire shaft at the same time. [32] estimated required torque and designed motors for ET in orchards and vineyards. They reported that the maximum required torque was 110 Nm in case of tasks without plowing using a cultivator. The motors in the THS, although different between the generation, has enough power. For example, the maximum output and torque are 53 kW and 163 Nm for the 4^{th} generation Prius, and 45 kW and 169 Nm for Yaris, which can work on spraying and mowing. On the other hand, speed control for driving is a problem when converting a power train system for automotive into agricultural vehicles. In farmland, a travel speed of less than 10 km/h is required generally. However, an automotive power train system targets a higher speed range than a farming one. Thus, a simple feedback control, such as the PID control, cannot control torque well, especially at a low-speed range, and therefore, traveling will be unstable. A speed control method adapting to farming is required if a reduction gear is not changed.

2.1.4. Purpose of this study

This chapter designed a spraying robot in orchards using the THS as a power train system. We picked an issue up when applying the THS to a robot farming vehicle for orchards without any remodeling, and suggested a solution. Here, there were three achievements in this chapter.

- A challenge and a solution were suggested when the HS for automotive is applied to farming machines
- Traveling performance was improved by developing a speed controller considering a traveling resistance
- A spraying test powered by the HS for automotive was conducted and the efficiency was verified

2.2. Materials and methods

2.2.1. Vehicle body

A vehicle body is based on a golf cart and the hybrid system that is used for a commercial car, Yaris made by Toyota was installed on it. Motors in the tested Toyota Hybrid System (THS) was a three-phase AC permanent magnetic synchronized motor that has 45 kW of the maximum output and 169 Nm of the maximum torque (Table 1).

Table 1 Specification of the THS

Number of engines	1
Number of motors	2
Motor type	3-phase AC synchronous motor
Max. output of the motor	45 kW
Max. torque of the motor	169 Nm

As is shown in Fig. 8, the THS is composed of a battery, an inverter, a motor, a generator, a planetary gear, a reduction gear, and an engine. The engine part was replaced with a farming implement. From now on, the engine shaft is called an implement shaft. A lithium-ion battery of 200 V, 6.1 kWh was used for a power source. A motor and a generator are the same function each other. Here, the generator in Fig. 8 is called a motor & generator 1 (MG1), the motor is called a motor & generator 2 (MG2). The planetary gear is for connecting three shafts: MG1, MG2, and implement shafts, transferring power each other. In the planetary gear mechanism of the tested THS, the MG1 works only on the implement shaft and the MG2 works strongly on a tire shaft. Therefore, the MG1 and the MG2 are controlled for rotating the implement and traveling, respectively. An impact on the implement shaft when rotating the MG2 can be ignored because it is enough small at a low-speed range. The specification of the vehicle body is shown in Table 2.

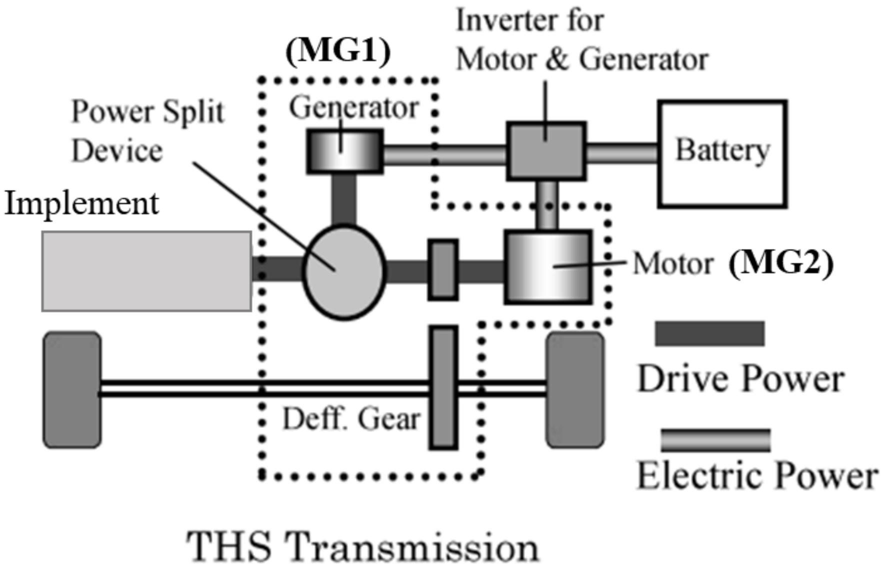


Fig. 8 The converted Toyota hybrid system for vineyards. The engine is replaced with an implement such as a pomp for a sprayer or a cutter for a mower. Based on [31].

Table 2 Specification of the developed vehicle body

Vehicle mass [kg]	650
Wheelbase [mm]	2100
Tire radius [mm]	285
Overall length [mm]	3200
Width [mm]	1400
Height (no roof) [mm]	1600

2.2.2. Sensors and network

An RTK-GNSS (SPS855, Trimble, USA) and an IMU (VN100, VectorNav, USA) were used as navigation sensors for autonomous traveling. A laser range finder (UTM-LX30, Hokuyo, Japan) for obstacle detection and a vision sensor for monitoring vegetation are also installed. Fig. 9 shows the overall system. A computer (TOUGHBOOK CF-20, Panasonic, Japan) was installed to obtain sensor information and calculate the input of a steering angle, MG1 torque, and MG2 torque. The sensor acquisition and calculation period were 10 Hz. A calculation result was sent to each node via the controller area network (CAN). The THS received an input torque from the computer and drove the MG1 and the MG2. The computer received information, such as remain capacity from the battery management system, rotation speeds of the MG1 and the MG2 from the THS, and a current steering angle from a steering control system via the CAN. A pump (6500C, Hypro, UK) was installed on the implement shaft for spraying. A liquid control unit and nozzles made by ARAG; Italy were also installed. We made a sprayer control system to open and close each nozzle automatically via CAN using Arduino Mega 2560.

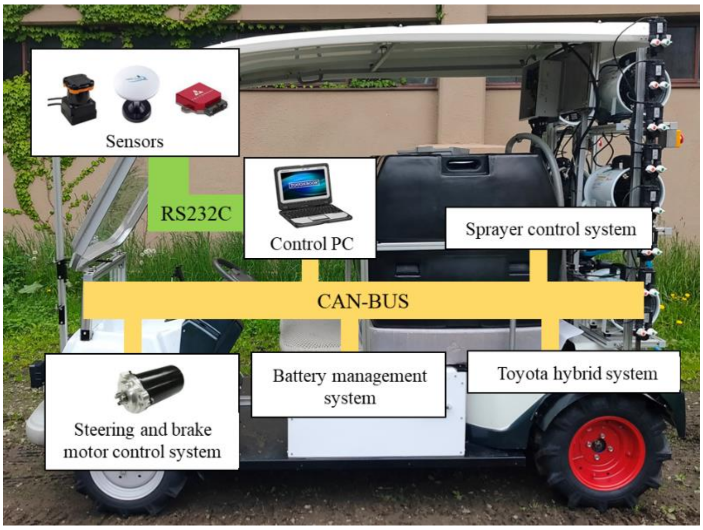


Fig. 9 The overview of the overall network system

2.2.3. Motor characteristics of the Toyota Hybrid System

The motor in the THS is, as described in Section 2.1.3, the three-phase AC permanent magnetic synchronized motor. Input current is controlled by a feedback loop so the output torque of the motor follows a desired value [33]. Two compensators are in the controller to minimize a control error due to battery voltage or a rotation speed of the motor. The error of the torque control is, to the target value of 12 Nm, 5 Nm at 2000 rpm, 3 Nm at 4000 rpm, and 2 Nm at 6000 rpm [34]. In short, the accuracy of torque control at a lower rotation speed is inferior to the one at a high.

[35] estimated an optimal reduction gear ratio of an HS for automotive and resulted in 5 to 15. Therefore, many HSs for automotive are made with a reduction gear ratio of around 10. On the other hand, [36] reported that the reduction gear ratio of 57 to 62 suits an ET of 78 kW, AWD. [37] also designed an ET that had two reduction gear ratios of 94.06 and 33.56. These ratios are 3 to 10 times larger than the ones for automotive, which is attributed to the fact that the motor for farming vehicles needs larger driving power at a low-speed range. Here, applying the HS for automotive to farming vehicles, speed control at low-speed range is challenging. For example, in the case of the tire radius r of 0.4 m, the speed V of 1 m/s, and the reduction gear ratio ξ of 50 (assumed a tractor), the motor rotation speed n will be about 1200 rpm as shown in Eq. (1). However, the one will be 240 rpm if the gear ratio ξ is 10 (assumed automotive).

$$n = \frac{60V\xi}{2\pi r} \tag{1}$$

Remember the fact that the accuracy of torque control for the motor will decrease as the rotation speed gets lower. Therefore, control for farming vehicles becomes more difficult. In addition, the vehicle speed with a smaller gear ratio is easily affected by the change of a motor rotation speed due to disturbance by traveling resistance. Particularly, the travel resistance is greater due to soil characteristics and slopes on fields in orchards. Therefore, speed control at working speeds (0.5 to 2.0 m/s) is a severe task to apply the HS for automotive to farming vehicles. There are two solutions as follows.

A) Make a gear ratio larger by adding another mechanism

B) Remove disturbance by estimating travel resistance

A) is solving by a hardware, while B) is by a software. However, A) is unpractical with respect to durability and developing cost, so we adopted B) and developed a speed controller considering travel resistance.

2.2.4. Vehicle model

A power flow from a motor to a vehicle is shown in Fig. 10. The motor outputs torque controlled by an internal computer according to an input torque T. The torque generated by the motor is transferred to a tire with a radius r via a reduction gear with a reduction ratio ξ. Then driving force F_D is generated between the ground and the tire (Eq. (2)).

$$F_D = \frac{\xi}{r} T \tag{2}$$

On the other hand, the vehicle has travel resistance F_R against the driving force, and the sum of them is the total force F_{all} (Eq. (3)).

$$F_{all} = F_D - F_R \tag{3}$$

Here, given the vehicle mass m, velocity v changes as Eq. (4) according to the Newton's equation of motion.

$$dv = \frac{F_{all}}{m} dt \tag{4}$$

Therefore, a block diagram of the power flow between the motor and the vehicle is shown in Fig. 11. Note that the character s is the Laplace operator, and $F_R(s)$ is a Laplace transformed value from the travel resistance F_R.

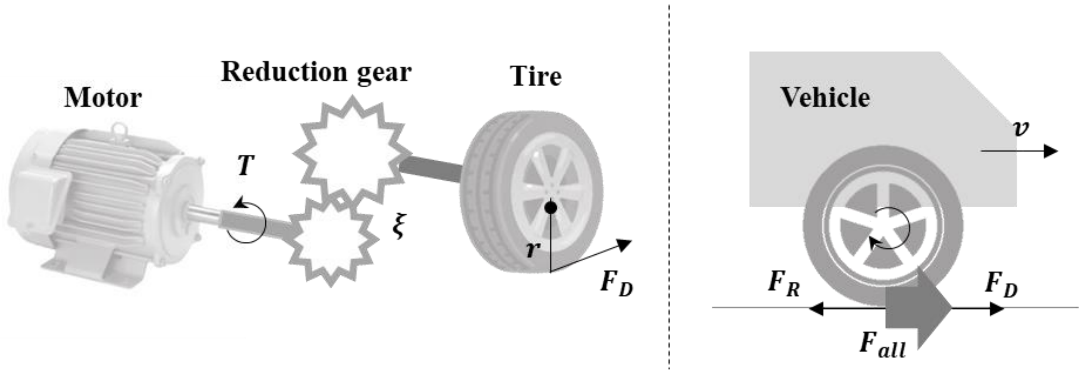


Fig. 10 The power flow from a motor to a vehicle via tires

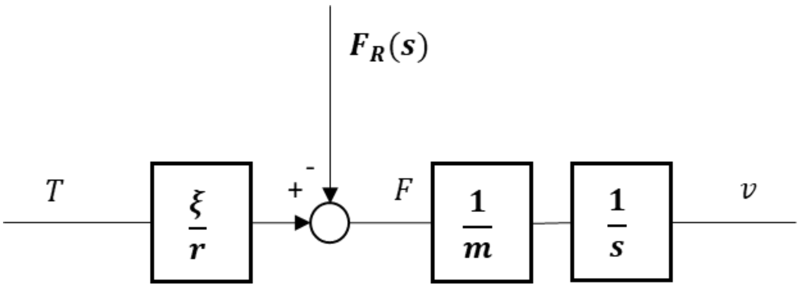


Fig. 11 A block diagram of the system between the motor and the vehicle

Next, considering the travel resistance F_R, there are six types: rolling resistance, slope resistance, air resistance, acceleration resistance, cornering resistance, and resistance when the vehicle climbs over an obstacle, according to [38]. Regularly, the rolling resistance and the air resistance are considered for automotive ([39,40]). However, the robot vehicle traveling in orchards can get larger rolling resistance and cornering resistance because of rugged tires and soil surfaces. In addition, orchards have a lot of slopes: the slope resistance must be considered. On the other hand, the vehicle speed when working is usually slower than 7 km/h, so the air resistance, which is affected by the vehicle speed, can be ignored. Hence, the rolling resistance F_{RR}, the slope resistance F_{SR}, and the cornering resistance F_{CR} are considered in this paper. The rolling and slope resistance is represented as Eqs. (5), (6). Note that *m* is the vehicle mass, g is the gravitational acceleration, C_r is a rolling resistance coefficient, and θ is a slope angle.

$$F_{RR} = C_r mgcos\theta \tag{5}$$

$$F_{SR} = mgsin\theta \tag{6}$$

The cornering resistance, in the automotive sector, was modeled using the lateral force derived from a linear bicycle model such as [39]. However, that model assumed rubber tires with fewer grooves and hard ground. There are different conditions between rugged tires and soft soil surfaces because the resistance according to tire sinking occurs [41]. This paper estimated the cornering resistance F_{CR} by calculating a cornering resistance coefficient C_c

in Eq. (7) by a single regression analysis between a steering angle δ and input torque acquired from actual traveled data

$$F_{CR} = C_c\delta \tag{7}$$

2.2.5. Proposed speed control methodology

A speed controller was developed according to the system proposed in Section 2.2.4. A Proportional Integral (PI) controller with feedforward control to compensate for disturbance modeled in Section 2.2.4 was designed and named an “FF-PI” controller here. A block diagram of the FF-PI control is shown in Fig. 12. Note that K_p and T_i in Fig. 12 are a proportional gain and an integral time of the PI controller, respectively. An observed vehicle speed v was acquired from an RTK-GNSS and was passed by a moving average filter of n = 3 to remove high-frequency noise.

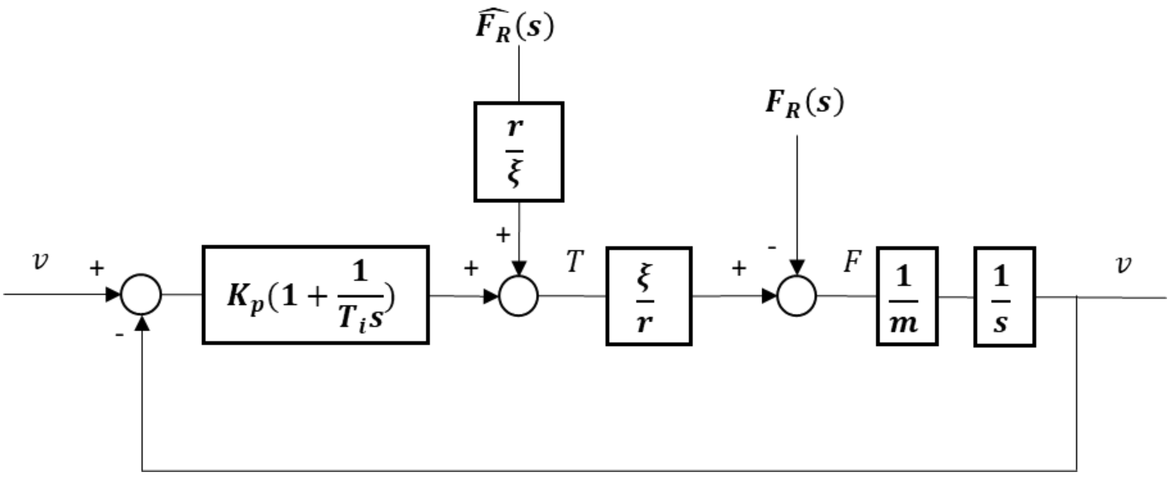
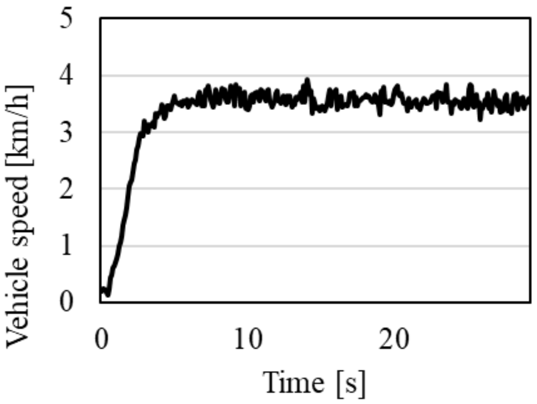
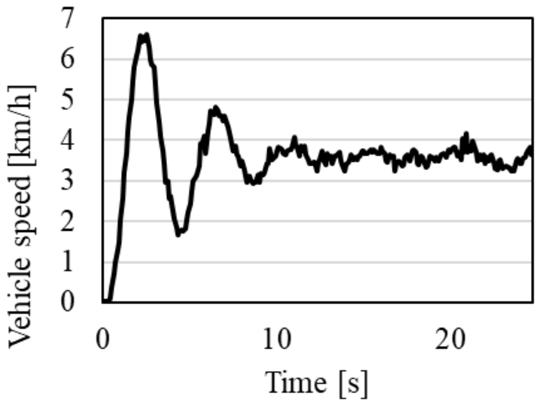


Fig. 12 A block diagram of a whole system including the PI controller and the feedforward controller based on travel resistance

Here, we can tune the proportional gain K_p, the integral time T_i of the PI part, the rolling resistance coefficient C_r, and the cornering resistance coefficient C_c for the feedforward part. First, we tuned the parameters for the PI part by making the robot travel in a flat and straight path. At that time, we tuned K_p and T_i by the try-and-error method to avoid overshoot without feedforward control. By the way, we did not use the Ziegler-Nichols method, a famous autotuning method, because too much overshoot occurred for the system handled in this study by that method. Fig. 13 shows the tuning results of both the try-and-error and the Ziegler-Nichols.



(A) By the try and error method



(B) By the Ziegler-Nichols method

Fig. 13 Traveling results for tuning the parameters of the PI controller. The PI controller tuned by the try and error method (A) was more appropriate than the one by the Ziegler-Nichols method (B) in this system.

Second, the rolling resistance coefficient C_r was estimated using the traveled data by the PI control. As you can see in Fig. 14, we regarded an average of the input torque when the vehicle speed became the steady state as rolling resistance and estimated C_r by substituting it for F_{RR} in Eq. (5). The tested field was a farm road where the soil surface was compacted and hard.

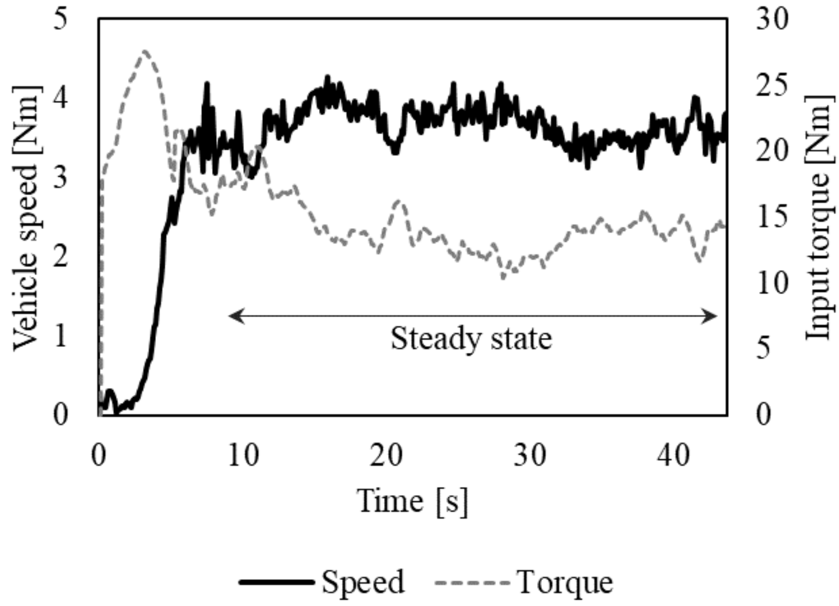


Fig. 14 The result of the vehicle speed and the input torque traveled on an unpaved road using the PI controller

Finally, the cornering resistance coefficient C_C was estimated using time-series traveled data, including large steering angles. As shown in Eq. (8), time-series cornering resistance $F_{CR,k}$ was estimated by calculating driving force $F_{D,k}$ slope resistance $F_{SR,k}$, and acceleration a_k at each time, according to Eqs. (2), (6), (9) using the vehicle speed, the input torque, and the vehicle pitch angle. The tested field was a vineyard with a grass surface.

$$ma_k = F_{D,k} - F_{SR,k} - F_{CR,k} \tag{8}$$

$$a_k = \frac{v_{k+1} - v_k}{\Delta t} \tag{9}$$

Regression analysis was conducted following procedures i, ii, iii below. The used dataset was a couple of the cornering resistance $F_{CR,k}$ and an absolute steering angle $|\delta|_k$. The result is shown in Fig. 15. Table 3 shows the specification of the tested vehicle and the parameters estimated in following procedures.

i. Acquiring an average of the cornering resistance $\overline{F_{CR,|\delta|}}$ at each absolute steering angle $|\delta|$ ($|\delta| = 1,\ 2, \dots,\ 40$)

ii. Estimating cornering resistance $\widehat{F_{CR,\,|\delta|}}$ by deducting rolling resistance from the average cornering resistance (Eq. (10)). Note that the average cornering resistance at $|\delta| = 0$ was regarded as rolling resistance.

$$\widehat{F_{CR,\,|\delta|}} = \overline{F_{CR,\,|\delta|}} - \overline{F_{CR,\,0}} \tag{10}$$

iii. Calculating a regression equation after removing outlier such as $\widehat{F_{CR,\,|\delta|}} < 0$

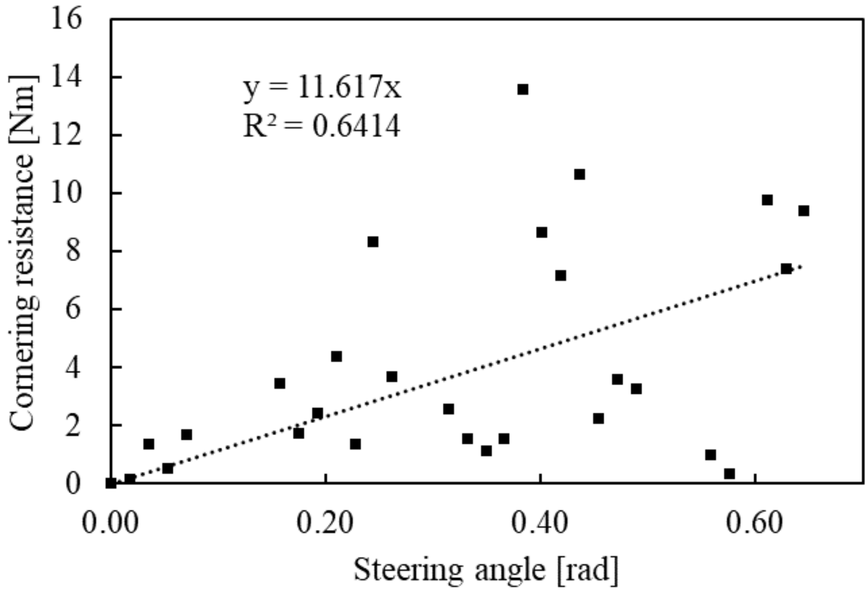


Fig. 15 Regression analysis between steering angles and cornering resistance

Table 3 Specification of the tested vehicle and the parameters

Vehicle mass m	650	[kg]
Wheelbase l	2.1	[m]
Tire radius r	0.285	[m]
Reduction gear ratio from MG2 to tire ξ	8.4	[-]

Proportional gain K_p	6.25	[-]
Integral time T_i	1.25	[-]
Rolling resistance coefficient C_r	0.060	[-]
Cornering resistance coefficient C_c	342	[N/rad]

2.2.6. Stability analysis

Characteristic roots were calculated in various conditions to verify the stability of the FF-PI controller. In Fig. 12, let $G_c = K_p\left(1+\frac{1}{T_i s}\right)$, $G_m = \frac{\xi}{r}$, $G_v = \frac{1}{ms}$, input *U*, output *Y*, the transfer function G of the whole system including the controller, the plant, and the disturbance is shown as Eq.(11) (Fig. 16).

$$G = \left\{(G_c + \frac{\widehat{F_R}}{G_m})G_m - F_R\right\} G_v \tag{11}$$

From Eq.(11) the characteristic equation is shown as Eq.(12).

$$1 + G = 1 + \left\{(G_c + \frac{\widehat{F_R}}{G_m})G_m - F_R\right\} G_v = \frac{ms^2 + \left(G_m K_P + \widehat{F_R} - F_R\right)s + G_m K_P}{ms^2} = 0 \tag{12}$$

Therefore, the characteristic roots are the solution of Eq.((13).

$$ms^2 + \left\{G_m K_P + \left(\widehat{F_R} - F_R\right)\right\}s + \frac{G_m K_P}{T_i} = 0 \tag{13}$$

According to the Quadratic Formula, the solutions are shown as Eq. (14).

$$s_1, s_2 = \frac{-\left(G_m K_P + \widehat{F_R} - F_R\right) \pm \sqrt{\left(G_m K_P + \widehat{F_R} - F_R\right)^2 - 4m\frac{G_m K_P}{T_i}}}{2m} \tag{14}$$

Here, assuming the disturbance of travel resistance was accurately estimated, Eq. (15) is established.

$$\widehat{F_R} - F_R = 0 \tag{15}$$

Thus, Eq. (16) is derived.

$$s_1, s_2 = \frac{-(G_m K_P) \pm \sqrt{(G_m K_P)^2 - 4m\frac{G_m K_P}{T_i}}}{2m} \tag{16}$$

In this case, the system is considered stable if both s_1 and s_2 are negative values. As you can see in Eq.(16), the sign of the characteristic root is determined by four parameters of G_m, K_P, T_i, m. In this study, the tire radius and the reduction gear ratio were constant: $G_m = \frac{\xi}{r} = \frac{8.4}{0.285} = 29.47$ referred to Table 3, and the sign of the characteristic root was examined in case that three parameters of K_P, T_i, m are variable. K_P varied 0 − 20, T_i did 1 − 10, m did 600 − 1600.

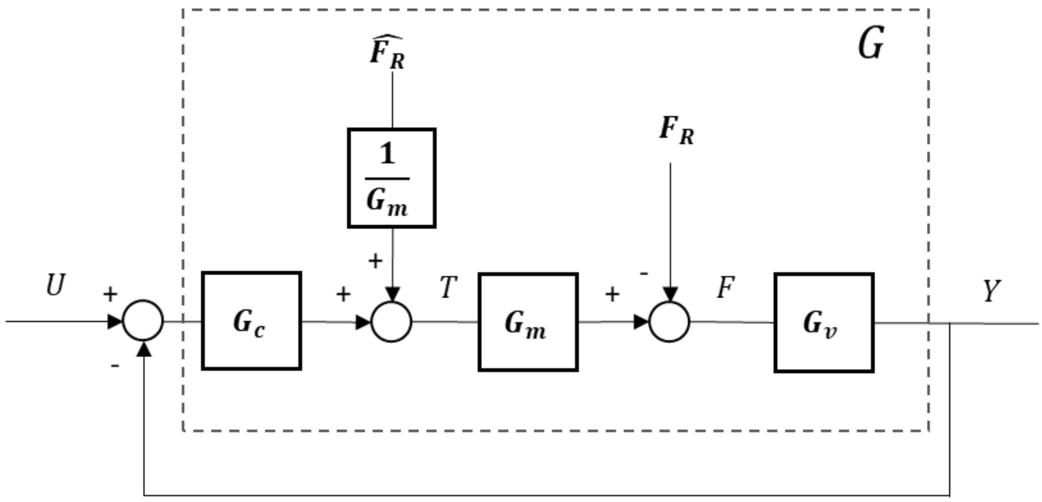


Fig. 16 An abstract block diagram including a controller, a plant, and a disturbance

On the other hand, the characteristic root is shown in Eq.(17) if $\widehat{F_R} = 0$, $F_R = \frac{k}{s}$ in Eq. ((13).

$$s_1, s_2 = \frac{-(G_m K_P) \pm \sqrt{(G_m K_P)^2 - 4m\left(\frac{G_m K_P}{T_i} - k\right)}}{2m} \tag{17}$$

This equation shows the system, including the amount of step disturbance k for the PI control without feedforward items. Values of the three parameters of K_P, T_i, m were the same in Table 3. The behavior of the characteristic root was examined as the travel resistance F_R (the amount of the disturbance k) increased.

2.2.7. Field trial with spraying

A traveling test with spraying in an orchard was conducted to verify that the robot vehicle worked a farming task using both two motors in the HS. The MG1 was controlled so the implement shaft was rotated at 700 rpm, which suit the tested pump. The resistance to the pump was constant, so the input torque was a constant value of 3 Nm. We verified the HS could be used for both traveling and farming tasks and discussed desired battery capacity from traveled data of battery consumption.

2.3. Results and discussion

2.3.1. Evaluation of the rolling resistance model

Fig. 17 shows the results of actual traveling and a simulation of the rolling resistance model. (A) and (B) in Fig. 17 show the simulation result at a set speed of 3.0 km/h. The simulation only included the rolling resistance as a disturbance. (C) and (D) show the travel result at a set speed of 3.6 km/h on a flat farm road. The PI controller was used for (A) and (B), while the FF-PI controller was used for (C) and (D). The travel result had a similar trend to the simulation result. Overshoot was observed for the FF-PI control. In addition, as shown in Fig. 18, a similar result was found when the vehicle weight increased by adding 500 kg of water to the tank, though the time to convergence was longer. Table 4 shows the travel results of the average speed and standard deviation without water (vehicle mass: 650 kg) and with water (vehicle mass: 1150 kg). The average speed was almost the same between the two controllers even if the vehicle mass changed. The standard deviation for the FF-PI control was smaller, but that was because the steady state did not include a part of the overshoot. Thus, if data includes that part, the standard deviation for the FF-PI control was definitely more significant than that for the PI control.

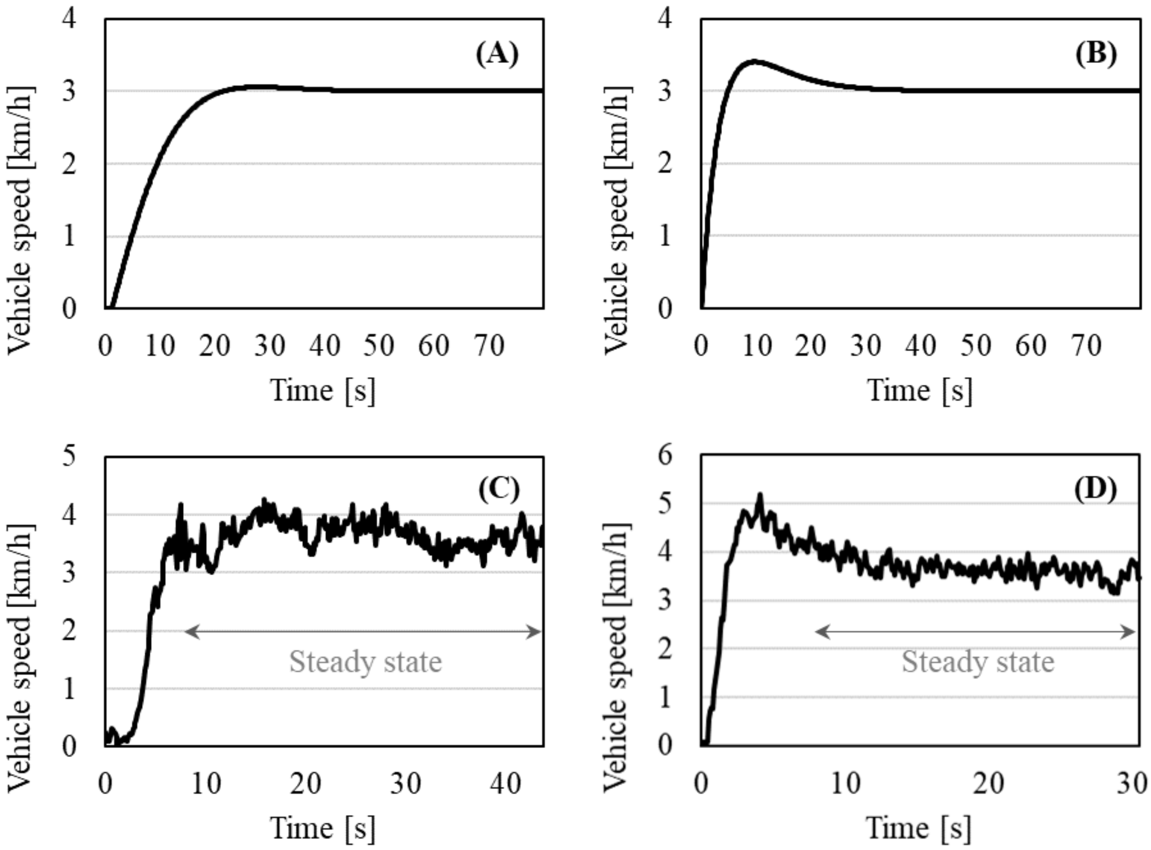


Fig. 17 The results of simulation (A and B) at the set speed of 3 km/h, and actually traveled (C and D) at the set speed of 3.6 km/h. (A) and (C) are the results using the PI controller, whereas (B) and (D) are the results using the FF-PI controller.

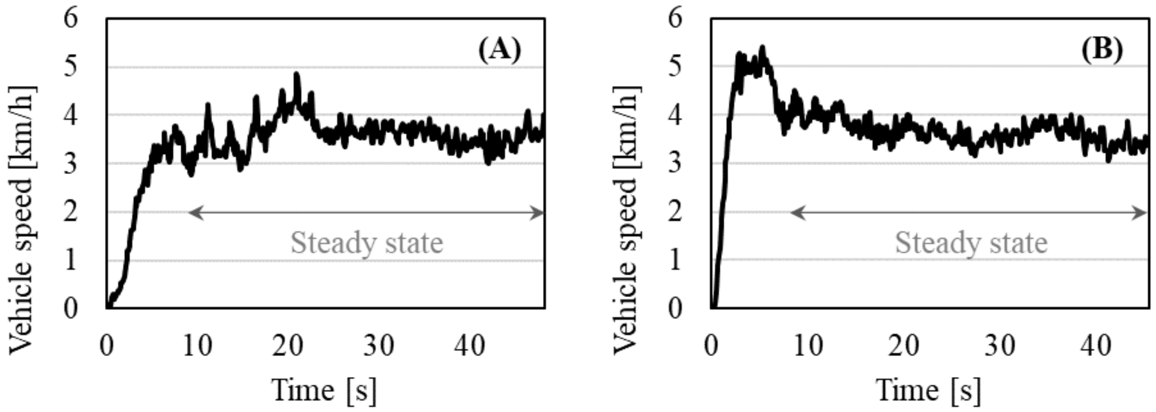


Fig. 18 The actual travel results at the set speed of 3.6 km/h using the PI controller (A), and using the FF-PI controller (B) with 500 kg added by water

Table 4 The results of an average and standard deviation of the vehicle speed during the steady state on each controller and weight

	Ave. [km/h]	Std. dev. [km/h]
PI (650 kg)	3.7	0.25
FF-PI (650 kg)	3.6	0.17
PI (1150 kg)	3.6	0.32
FF-PI (1150 kg)	3.6	0.20

A reason why the FF-PI control generated an overshoot was that the PI controller had already considered the rolling resistance because it was tuned by traveling on the road. In the tested environment, the feedforward item of the rolling resistance caused the overshoot. However, we have already observed that control performance improves at the start of moving by adding a smaller value of the rolling resistance than the actual value to the feedforward item. In case there is no feedforward control, it takes a long time to start moving if the control gain is a small value, leading to overcurrent for the motor because the current flows without any rotation for the motor. Then, if the control gain increases to avoid the delay time, the output speed oscillates. This phenomenon particularly occurred on a soft soil surface where the rolling resistance was remarkably larger than on a paved road. Fig. 19 shows a simulation result of the PI and FF-PI control given the static rolling resistance of 0.12 and the dynamic rolling resistance of 0.20 of the plant. Note that the rolling resistance was divided into static and dynamic ones in the simulation: static one was until the start of moving, dynamic one was during moving. K_p was the same for the tested three controllers: an FF-PI controller and two PI controllers, while T_i was changed. $T_i = 8.0$ and $C_r = 0.06$ for the FF-PI control, and $T_i = 8.0$ (PI-8.0) and $T_i = 1.0$ (PI-1.0) for the PI control. As a result, PI-8.0 had a delay time, and PI-1.0 had an overshoot, while the FF-PI had no delay time and less overshoot. These results indicated the conclusion that adding a smaller value of rolling resistance than the actual value in the FF-PI control improved control performance because that compensated the static rolling resistance.

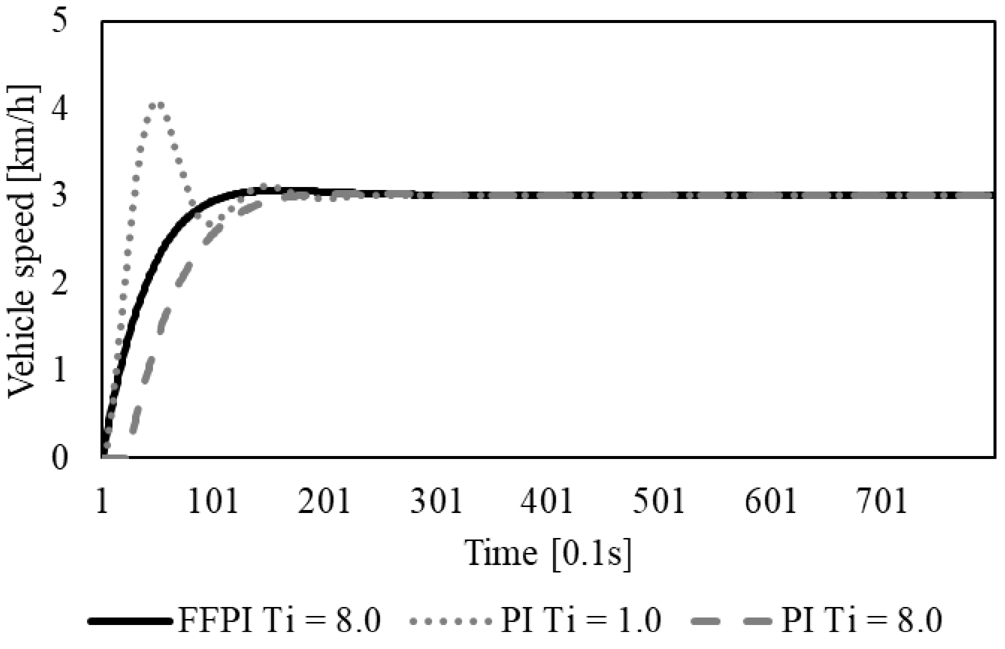


Fig. 19 A simulation result of speed controls by the FF-PI and the PI controller

2.3.2. Evaluation of the slope resistance model

Fig. 20 shows a simulation result of the situation that the step disturbance of slope angles of 5 degrees was added 40 seconds after the control began. The vehicle speed decreased according to the disturbance for the PI control, while it was constant for the FF-PI control.

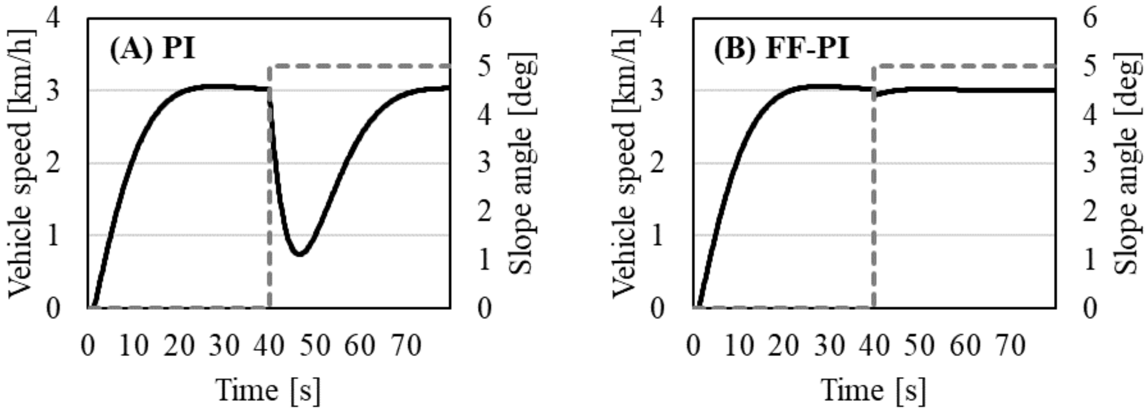


Fig. 20 Simulation results of vehicle speeds by the PI (A) and the FF-PI (B) controller. Solid and dashed lines indicate vehicle speeds and slope angles, respectively.

Fig. 21 shows a result in the vineyard where the robot vehicle traveled a slope surface of 8 degrees. The vehicle speed got slow against the set speed as the slope angle increased for the PI control, while the phenomenon was mitigated for the FF-PI control. Table 5 shows the observed vehicle speed at the set speed of 3.6 km/h. The PI control had deviation between the average speed and the set speed, while the FF-PI had no deviation. In addition, the standard deviation of the vehicle speed for the FF-PI control was half less than the PI control. These results indicated that the FF-PI control was effective to the slope resistance.

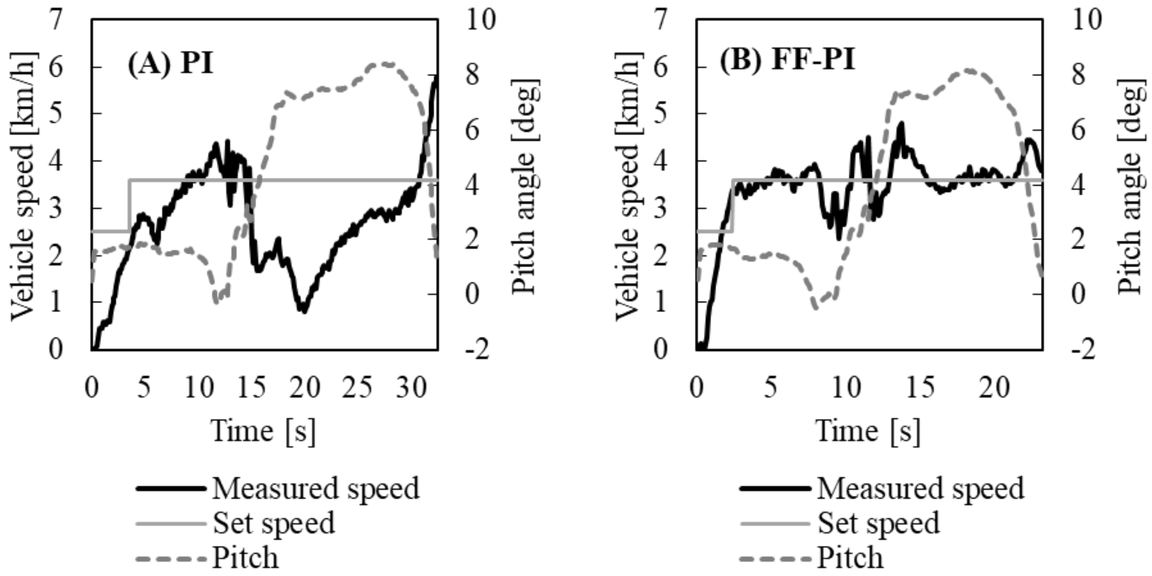


Fig. 21 Measured vehicle speeds by the PI (A) and the FF-PI (B) controller on a slope field

Table 5 Averages and standard deviation (SD) of measured vehicle speeds at a set speed of 3.6 km/h

	Average [km/h]	SD [km/h]
PI	2.8	1.0
FF-PI	3.6	0.4

2.3.3. Evaluation of the cornering resistance model

Fig. 22 shows a result of the vehicle speed according to changes in the steering angles. The vehicle once stopped at the steering angle of 20 degrees for the PI control. Then, the input torque was increased by the integral item, and the vehicle moved again at the elapsed time of 10 seconds. Unfortunately, the vehicle speed kept increasing and reached 6 km/h, which was obviously an overshoot to the set speed of 2.0 km/h. On the other hand, though the vehicle speed increased a little as the steering angle increased at the elapsed time of 10 seconds, the vehicle speed did not reach 0 km/h and converged to the set speed. The standard deviation and the root mean square (RMS) error for the FF-PI control were half less than the ones for the PI control (Table 6). This result indicated that the FF-PI control was effective in the cornering resistance. The site of https://youtu.be/Iyoqoz0Scjo includes a video showing the behavior of cornering. Note that the location we acquired data and took the video differed.

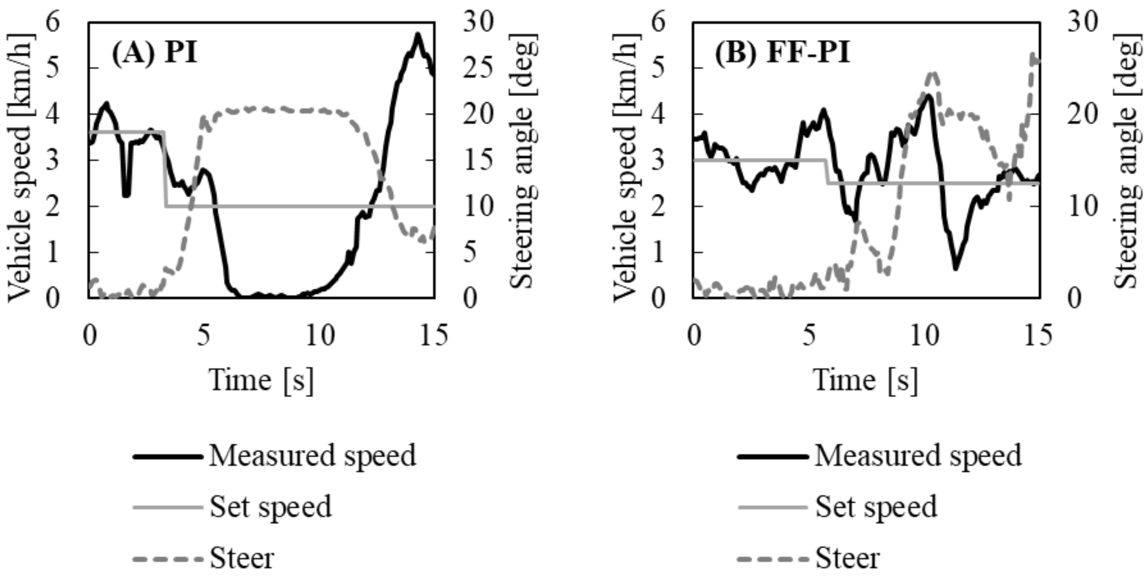


Fig. 22 Measured vehicle speeds by the PI (A) and the FF-PI (B) controller at a field trial including turning maneuver.

Table 6 Standard deviation (SD) and root mean square error (RMSE)

	SD [km/h]	RMSE [km/h]
PI	1.8	1.6
FF-PI	0.7	0.7

For the FF-PI control, the vehicle speed was over the set speed as the steering angle increased, and was slower than the set speed by the feedback controller. This result was because the modeled cornering resistance was more significant than the actual one. The model was designed so the resistance increased linearly according to the steering angle, but actually, the cornering resistance was smaller when the steering angle is small because the relationship between the steering angle and the resistance was non-linear. However, the FF-PI control could prevent the vehicle from stopping, while the PI control led to a stop and large overshoot. Generally, the static friction resistance was larger than the dynamic friction resistance. Particularly when the steering angle is large, the phenomenon was often observed. In short, the system remarkably changes by the vehicle states: moving and stopping. Therefore, avoiding stopping is essential for the robot vehicle to travel stably. The feedforward approach proposed in this study gave a quick response to disturbance, and was effective in preventing the vehicle from stopping from those experiments. In terms of the slope resistance and the cornering resistance, especially, the PI control sometimes gave the vehicle stop due to the slow response, while the FF-PI control prevented the vehicle from stopping because of a quick response by estimating the travel resistance.

2.3.4. Stability analysis

The real solutions of the characteristic roots s_1 and s_2 in Eq.(16) were negative in the condition that K_P, T_i, and m changed from 0 to 20, 10 to 1, and 600 to 1600, respectively. The real part σ was also negative when the characteristic root had an imagination value (Fig. 23). This result showed the system was stable.

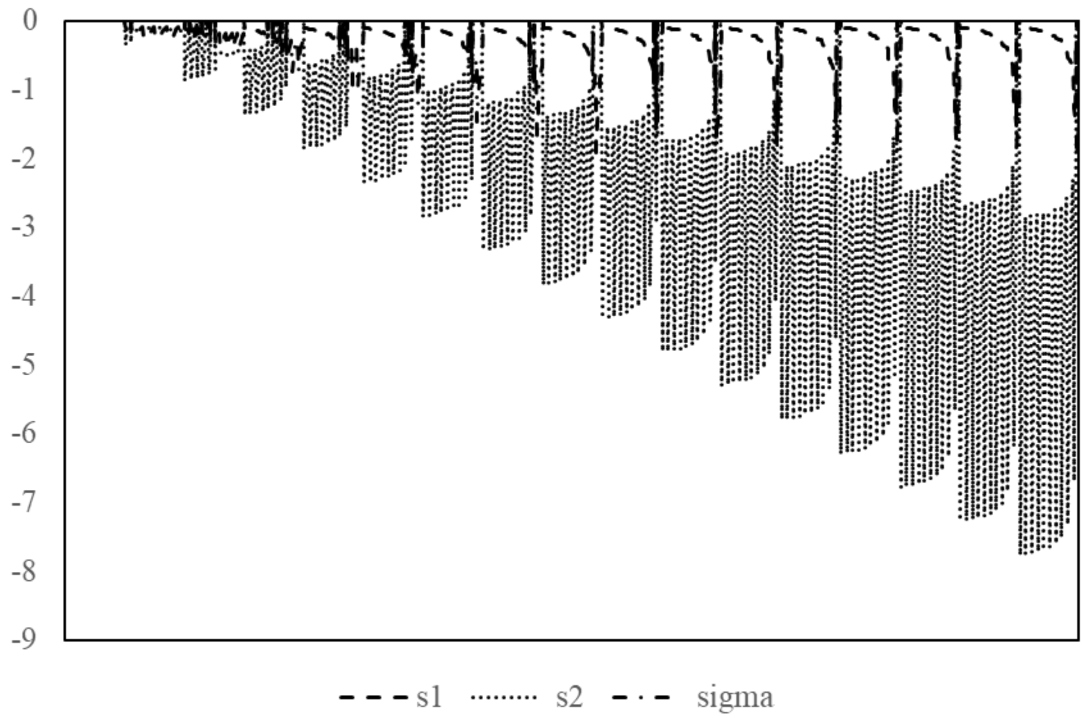


Fig. 23 Plot of the characteristic roots of s_1, s_2, and the real part of imaginary solution σ in every K_P, T_i, and m

As described above, we can verify the stability of the PI control to observe the behavior when k increases in Eq. (17). Fig. 24 shows a result of the characteristic root as k increased gradually given $K_P = 50,\ T_i = 10,\ m = 650$. The result shows that one of the characteristic roots was positive at the traveling resistance $F_R = 150$ [N], which indicated the system became unstable. $F_R = 150$ [N] means a constant resistance value equivalent to the rolling resistance coefficient of around 0.03 or around 2 degrees for the slope resistance. In the case of traveling on a flat asphalt road with radial tires, the rolling resistance coefficient was about 0.01, so we can understand that the PI control was enough stable. The vehicle traveled stably by the PI control in such an environment. However, the rolling resistance coefficient is relatively large: 0.03 – 0.1 in case of traveling on a soil surface by rugged tires. In addition, orchards have many slopes with 5 – 15 degrees. The stability analysis showed, in such an environment, that the PI control is unstable easily, whereas the FF-PI control considering traveling resistance gives stable traveling.

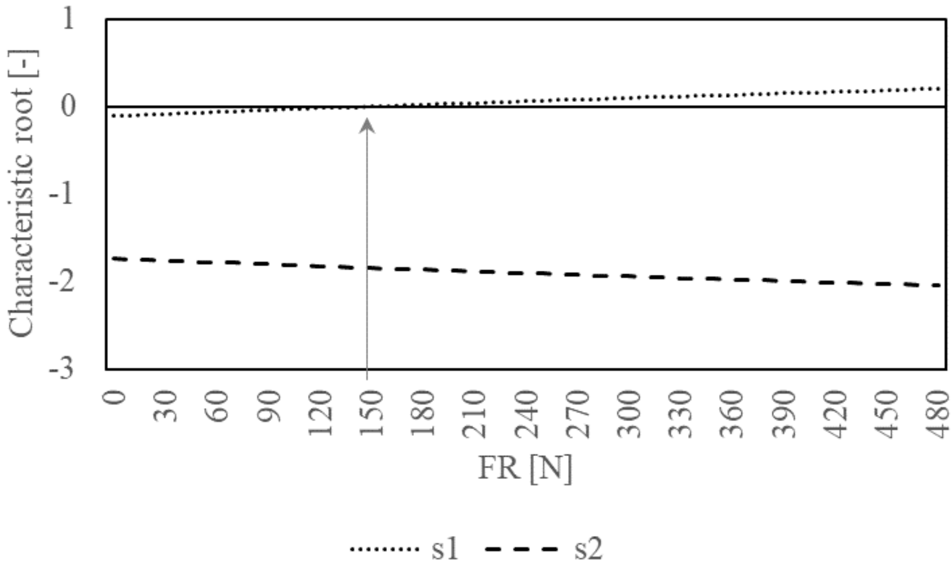


Fig. 24 Plot of s_1 and s_2, according to F_R

2.3.5. Evaluation of spraying robot

The robot sprayed autonomously on four paths, the total of 520 m, in the orchard. Fig. 25 shows the results of a traveled trajectory and a change of the MG1 rotation speed. When the robot started to turn, the MG1 was automatically stopped. The state of charge (SOC) changed by 13 % at this trial. Having a battery capacity of 6.1 kWh, the robot vehicle consumed 0.8 kWh in that work. The tested orchard has an area of about 1 ha and 43 paths. Given the width of 2.5 m, the average distance of a path was about 100 m. Thus, energy consumption was about 6.6 kWh if the robot vehicle works in a whole area of 1 ha in the orchard based on traveled data. The tested battery was used within the SOC of 80% upper and 30% bottom to save the battery life, so the active power was 50%, equivalent to 3.05 kWh. Therefore, two times charging is necessary for work in 1 ha. The charging time can be reduced if we use a bigger battery, such as 30 – 100 kWh, usually installed on an electric car.

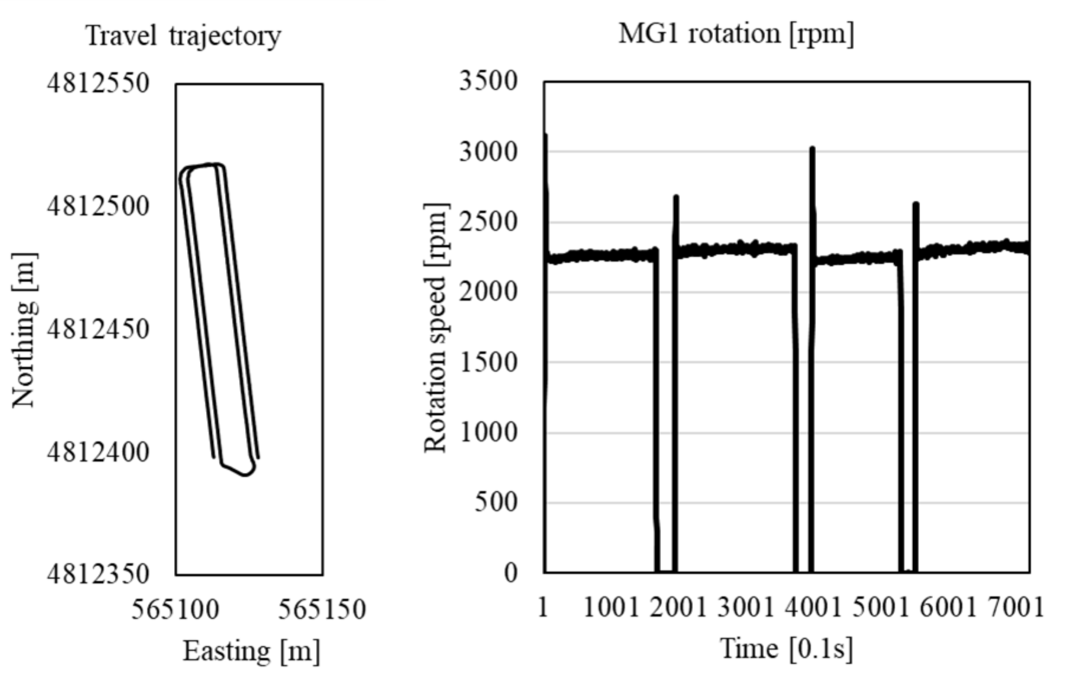


Fig. 25 Travel trajectory (left) and MG1 rotation (right) while spraying in an orchard

2.4. Conclusion of this chapter

We clarified the issues when applying the hybrid system to an agricultural machine and solved the problem by developing a speed controller considering travel resistance. The proposed controller quickly responded to the disturbance by modeling the travel resistance, and therefore, the travel performance was improved, especially on a sloped field and at turning. The stability analysis showed that the proposed controller was stable under field conditions, whereas the conventional PI controller sometimes became unstable. The result showed that the HS could be used for work on spraying by controlling two motors on it. This study concluded that the robot vehicle in orchards using the HS for automotive has the potential to be automated for a spraying task.

Chapter 3. Autonomous navigation using the Quasi-Zenith Satellite System (QZSS)

3.1. Introduction

3.1.1. A challenging of the RTK-GNSS

The real-time kinematic global navigation satellite system (RTK-GNSS) is one of the most common sensors for autonomous driving in crop fields because it can locate itself by using the global coordinate system [42]. The RTK-GNSS requires a correction signal and has two general methods of obtaining it: using a reference base station or using the internet. The base-station method uses two RTK-GNSS receivers (Radio-RTK). One is for a rover and the other is for a base station that sends a correction signal to the rover via the radio. Typically, it is difficult to use the Radio-RTK in vast farmland because the rover and the base station must remain within hundreds of meters of each other. In addition, the Radio-RTK initially requires an accurate position for the base station, which means that an additional measurement procedure is needed to obtain such a position. Therefore, the virtual reference station (VRS) method, which uses the internet, is common for farmland. The RTK-GNSS using VRS (VRS-RTK) obtains a correction signal for the rover via the internet from a service provider. The provider receives an approximate position from the rover and calculates correction parameters using four or more GPS-based control stations near the rover. These control stations were established by the Geospatial Information Authority of Japan. Thus, only a rover is necessary for positioning. However, it is difficult to use the VRS-RTK in mountains, forests, or rural areas that the internet does not reach. It can also be difficult to use VRS-RTK near buildings, windbreaks, or forests in farmland because covering the top of an antenna blocks GNSS signals obtained from the same satellites as the GPS-based control station.

3.1.2. What is the Centimeter-Level Augmentation Service (CLAS)?

The CLAS is a service that provides an augmentation signal via the QZSS. The CLAS adopts the precise point positioning real-time kinematic (PPP-RTK) method defined by RTCM STANDARD 10403.2 as augmentation information [43]. The target satellites for augmentation are QZSS (L1C/A, L1C, L2C, L5), GPS (L1C/A, L1C, L2P, L2C, L5), and Galileo (E1B, E5a). The number of target satellites has increased from 11 to 17 since November 2020. The Quasi-Zenith Satellite System Performance Standard [43] shows that GLONASS (L1, L2) is also going to be targeted in the near future. The official horizontal accuracy of the CLAS is less than 0.06 m (95%) at static positioning and less than 0.12 m (95%) at dynamic positioning, while that of the RTK is 0.008 m + 1 ppm root mean square (RMS) at static positioning [44]. These results show that the accuracy of the CLAS is inferior to that of RTK, but the CLAS has enough potential to be used as a navigation sensor for a robot vehicle in farmland. Moreover, the CLAS does not require the internet because it creates augmentation information by estimating position errors using about 250 observation data sources from GPS-based control stations around Japan and sends them to the rover via the QZSS [45] (Fig. 26). This may enable the rover to obtain an accurate position even near buildings or windbreaks because the CLAS does not have to rely on the same satellites that the GPS-based control stations do, unlike the VRS-RTK.

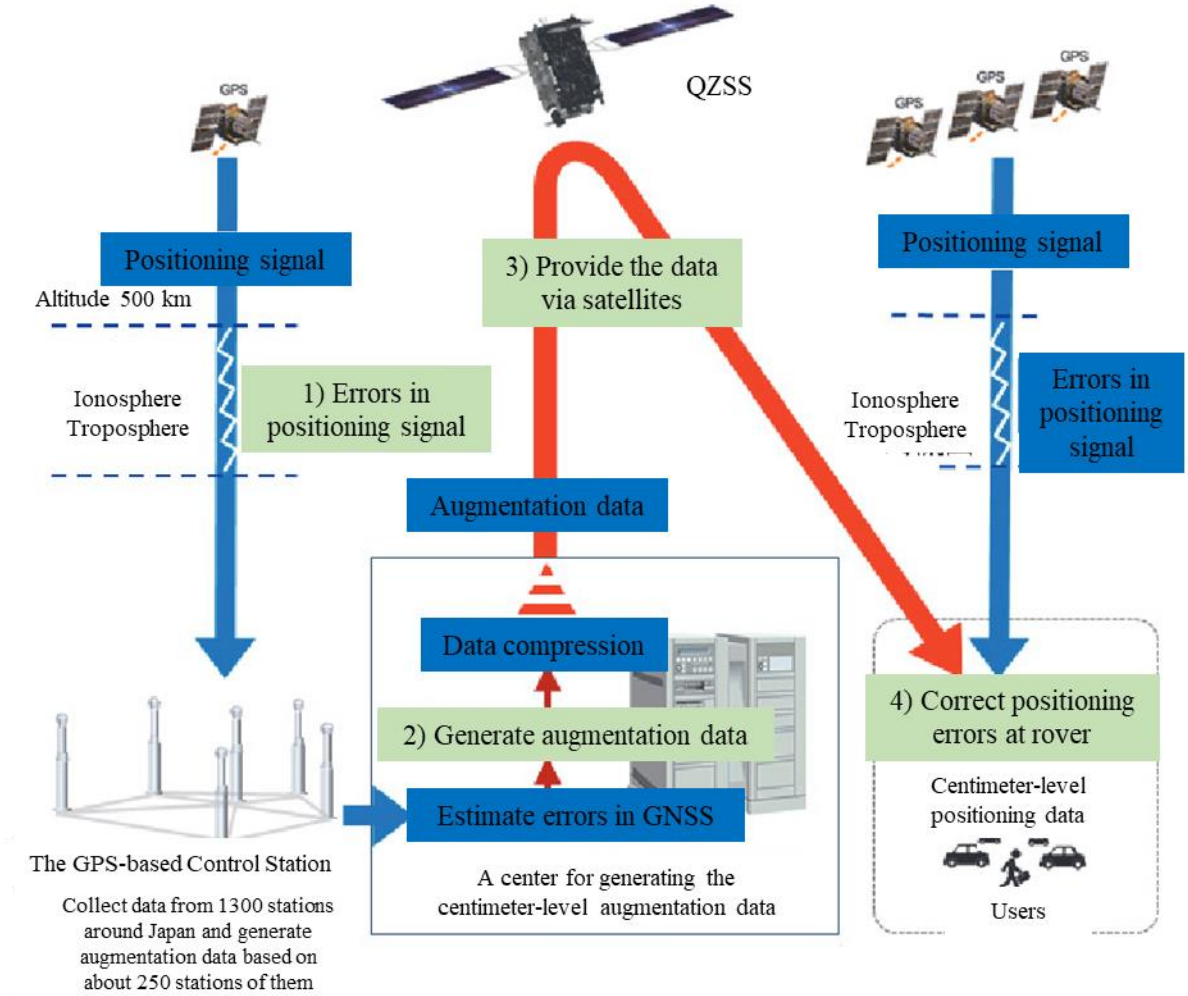


Fig. 26 An overview of the CLAS (based on [45])

As far as I know, one of the internet signals can be available at orchards in Japan. However, although there are many internet providers, it is possible that only one provider can be available. In this case, if you do not make a contract with the provider, you cannot use the

VRS-RTK. If there are buildings or windbreaks near an orchard, the internet signal becomes weak because of those obstacles even if you can get the internet signal at different places in the same orchard. Therefore, the CLAS without the internet has great merit. From a different perspective, the VRS-RTK sometimes drops down from a FIX solution to a FLOAT solution near trees. It possibly happens not only in orchards but also in normal cropland that has large windbreaks. I have faced the situation many times in demonstrations for autonomous navigation. The CLAS also has the same problem because it uses satellites. However, the CLAS signal is provided by the QZSS from just above the robot, and augmentation information is generated based on 250 stations in Japan. Due to those reasons, the CLAS positioning can be much more robust than the VRS-RTK.

3.1.3. Purpose of this study

Open flat fields were often targeted for research into CLAS as a navigation sensor for an agricultural robot vehicle. Wang and Noguchi [46] verified the traveling accuracy of a robot tractor navigated by the CLAS on a farm road near windbreaks and in a crop field. The results showed that the traveling accuracy of the CLAS was as good as that of the VRS-RTK. They also showed that the CLAS could obtain a FIX solution near the windbreaks more frequently than the VRS-RTK. Udompant et al. [47] applied the CLAS to a robot combine harvester and showed that the CLAS could obtain a FIX solution constantly over time. They concluded that the robot combine harvester using the CLAS could travel autonomously at any time. These studies indicate that the CLAS has enough potential to be applied to robot farm vehicles in flat fields. On the other hand, little to no research has been done to show that the CLAS can be utilized in mountainous farmland such as orchards or vineyards that are surrounded by windbreaks or forests and have steep slopes. These key differences in terrain and environment constitute the originality of this research.

This study focuses on the application of the CLAS to mountainous farming. The robustness of the system to an antenna inclination or windbreaks and the accuracy of the CLAS were verified by static experiments. We then determined the use case where the CLAS was superior to the VRS-RTK in vineyards, defined as fields where grapevines are trained over vertical trellises, in mountainous areas. Then we tested autonomous driving navigated by the CLAS and discussed the potential applicability of the CLAS to a robot vehicle in mountainous farmland.

3.2. Materials and methods

3.2.1. Basic experiment for validating robustness to buildings and antenna inclination

To verify the robustness of the CLAS system to buildings and windbreaks, the time to first fix (TTFF), defined as the elapsed time from catching satellites to acquiring a FIX solution, of both the CLAS and the VRS-RTK were measured near buildings of the Faculty of Agriculture at Hokkaido University (Fig. 27). Two receivers acquired satellite signals from an antenna simultaneously using a signal splitter. The tested receivers, the antenna, and the splitter are shown in Table 7. The receivers could utilize the GPS (L1・L2・L5), QZSS (L1・L2・L5・L6), GLONASS (G1・G2), Galileo (E1・E5a・E5), and Beidou (B1・B3) systems. RTCM 3.2 was used to provide correction information of the VRS-RTK. The elevation angles of both the CLAS and the VRS-RTK were 15 degrees. Considering that the positions of satellites vary over time, the positioning tests were conducted 20 times at different times of day, and the time in each test was within 25 minutes. The average of the TTFF and the number of times needed to get a FIX solution earlier than the other receiver were recorded.

Fig. 27 The condition of the experiment near buildings

Another positioning test was conducted to verify the robustness of the antenna inclination. An inertial measurement unit (IMU) (Table 7) was installed on the antenna. As mentioned above, the TTFF of both the CLAS and the VRS-RTK was measured 10 times with the antenna tilted 15 degrees (Fig. 28), based on our assumption that the robot vehicle travels a field that has 15-degree slopes. At the same time, after acquiring the FIX solution for both the CLAS and the VRS-RTK, the FIX status was observed as the antenna tilted gradually.

Fig. 28 The condition of the experiment with antenna titled

Table 7 List of tested equipment for positioning

	CLAS receiver	VRS-RTK receiver	Antenna	Splitter	IMU
Model	MJ-3021-GM4-QZS-EVK	MJ-3021-GM4-QZS-EVK	MJ-3009-GM4-ANT	LDCBS1X2	VN100
Manufacturer	Magellan Systems Japan	Magellan Systems Japan	Magellan Systems Japan	GPS Networking, Inc.	VectorNav
Country	Japan	Japan	Japan	USA	USA

3.2.2. Robustness to windbreaks and internet environment in mountainous farmland

A static positioning test was performed at Noto Vineyard, Anamizu-Cho, Ishikawa, Japan, in February 2021 to verify the positioning accuracy of the CLAS (Fig. 19). The positions of the CLAS were measured at eight points randomly chosen in the vineyard, and the VRS-RTK was used as reference. Table 8 shows the two tested receivers. The accuracy, defined as the horizontal positioning error (HPE) from the reference point, was measured. The HPE is an important index to verify the GNSS receiver in precision agriculture [48]. Here, the reference point was the average position of the VRS-RTK in this experiment. Assuming that the reference point was $(\hat{X}, \hat{Y})$, the time-series positioning data were (X_i, Y_i) and the HPE ε_i was calculated according to Eq. (18). The average μ, standard deviation σ, and μ+1.96σ (95% confidence interval) of ε_i were calculated, and the accuracy of the CLAS was discussed.

$$\varepsilon_i = \sqrt{(\hat{X} - X_i)^2 + (\hat{Y} - Y_i)^2} \tag{18}$$

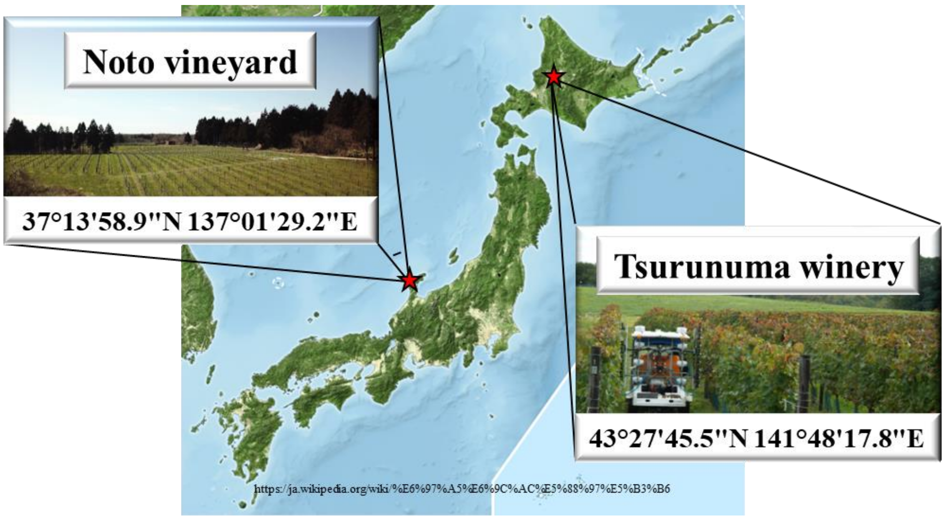


Fig. 29 Experiment field (Based on Google Earth)

Table 8 List of tested GNSS receiver

	CLAS receiver	VRS-RTK receiver	Antenna
Model	MJ-3021-GM4-QZS-EVK	SPS855	MJ-3009-GM4-ANT
Manufacturer	Magellan Systems Japan	Trimble	Magellan Systems Japan
Country	Japan	USA	Japan

Next, the TTFF of the CLAS and the VRS-RTK were measured at three points near trees to verify the effect of windbreaks existing in an actual vineyard for positioning (Fig. 30).

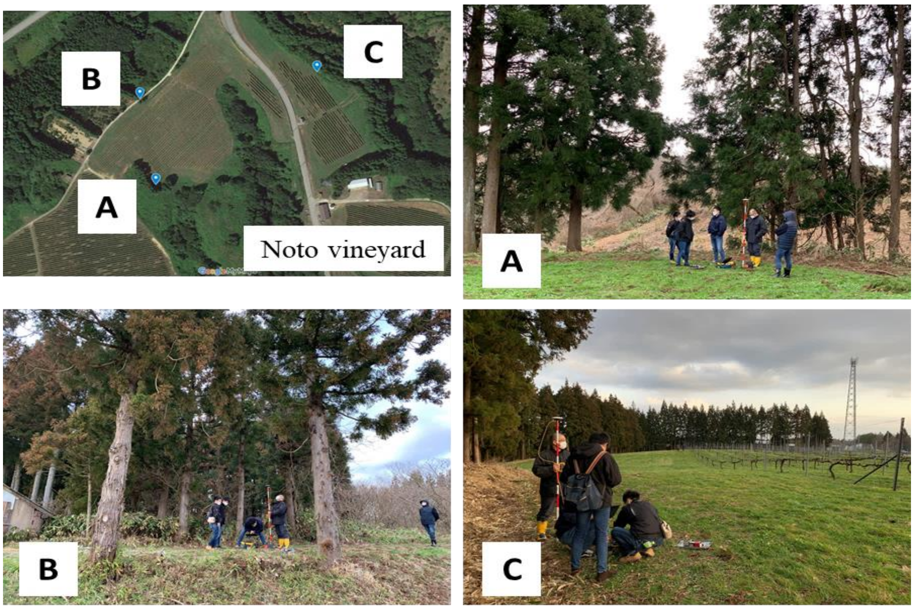


Fig. 30 Positioning points near windbreaks at Noto Vineyard. Point A was in a headland of the vine rows near approximately 10 trees. Point B was on a farm road with a few trees. Point C was next to vine rows near a large forest.

In addition, a static positioning test was performed at Tsurunuma Winery, Urausu-Cho, Hokkaido, Japan, to verify the FIX ratio between the CLAS and the VRS-RTK in mountainous farmland where the internet signal was weak. The tested receivers of the CLAS and the VRS-RTK are shown in Table 7, and the Radio-RTK is shown in Table 9.

Table 9 List of tested Radio-RTK receiver

	Radio-RTK base	Radio-RTK receiver
Model	NetR9	SPS855
Manufacturer	Trimble	Trimble
Country	USA	USA

3.2.3. Tested vehicle configuration

Fig. 31 shows an overview of the tested robot vehicle used for the experiments. The antenna, the splitter, and the two receivers of the CLAS and the VRS-RTK were installed in the vehicle to acquire position data on traveling. The IMU (Table 7) for acquiring the vehicle attitude angle, a computer (Toughbook CF-20, Panasonic, Japan) for vehicle control, and a laser scanner (UTM-LX30, Hokuyo, Japan) for obstacle detection were also installed. The tested robot vehicle had a tank and nozzles because it was designed to be used as a sprayer machine.

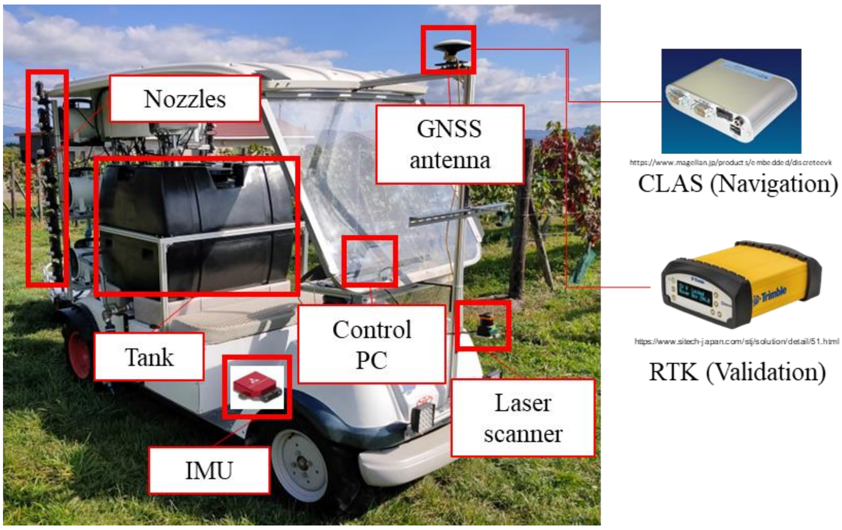


Fig. 31 Overview of the robot electric vehicle

3.2.4. Dynamic experiment in a mountainous area

A dynamic positioning test of the CLAS was conducted at Noto Vineyard in February 2022 to verify its dynamic accuracy as a navigation sensor for autonomous driving. The slope angle of the tested vineyard was between -4.0 and +3.3 degrees. The tested vehicle is shown in Section 3.2.3. The tested vehicle traveled two paths that were each about 230 meters long at speeds of 1.0 m/s and 1.3 m/s. The measurement frequency was 10 Hz. The trajectory error e_i was estimated as shown in Fig. 32 by comparing the CLAS with the RTK as reference. Note that the line segment L_i in Fig. 32 indicates a line segment from point R_i to point R_{i+1}, and S_i represents the area of the triangle consisting of points R_i and R_{i+1} from the RTK and point C_i from the CLAS.

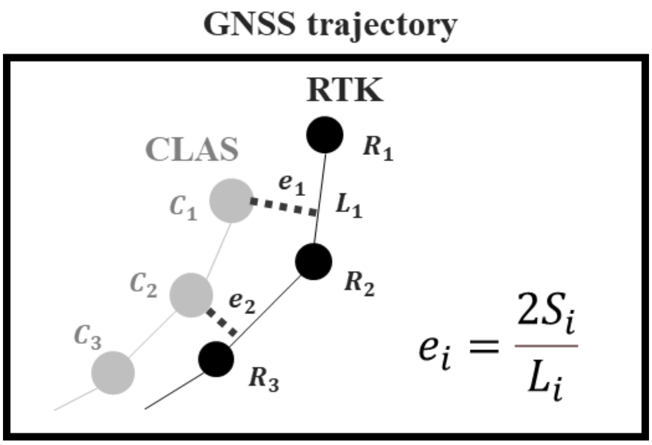


Fig. 32 Method of estimating trajectory errors

3.2.5. Autonomous navigation trial

The robot vehicle traveled while being navigated autonomously by the CLAS. The method of Kise et al. [49] was adopted for deciding the input steering angle, which is estimated by the feedback control from the state of the lateral deviation d and the heading error $\Delta\phi$ based on a navigation map as shown in Fig. 33. After the CLAS created the navigation map, steering was controlled using the vehicle position obtained by the CLAS. The RTK position was also recorded for reference.

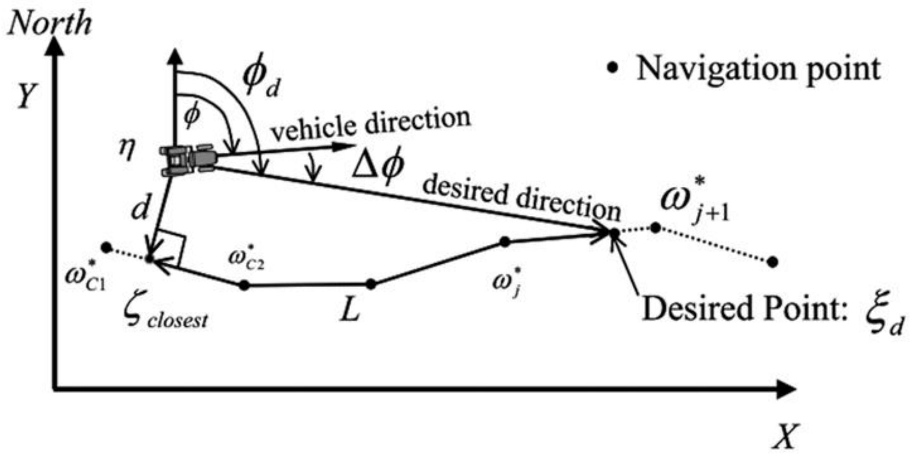


Fig. 33 Definition of a navigation map (Source: Noguchi et al. [50])

3.3. Results and discussion

3.3.1. Robustness to multiple paths and to antenna inclination

Table 10 shows the static positioning result of the CLAS and the VRS-RTK near buildings in Section 3.2.1. The TTFF of the CLAS was 1 minute earlier on average and 2 minutes smaller with standard deviation than that of the VRS-RTK. The CLAS obtained a FIX solution earlier than VRS-RTK 12 times in 20 tests. In 18 of the 20 tests, the CLAS obtained a FIX solution within 25 minutes, whereas the VRS-RTK did so 17 times. This result indicated that the TTFF of the CLAS was shorter than that of the VRS-RTK near buildings. Note that, however, in at least 2 of the 20 tests, neither the CLAS nor the VRS-RTK could get a FIX solution within 25 minutes.

Table 10 Average and standard deviation of the TTFF and successfully fixed time within 25 minutes between the CLAS and the VRS-RTK near buildings (n = 20)

	Ave. TTFF [min]	Std. TTFF [min]	FIX within 25 min.
CLAS	3.2	1.9	18
VRS-RTK	4.3	3.9	17

Table 11 shows the static positioning results for the CLAS and the VRS-RTK with the antenna tilted about 15 degrees. The TTFF of the CLAS was 90 seconds earlier on average and 90 seconds smaller with standard deviation than that of the VRS-RTK, which was the same trend as that near buildings. The CLAS got a FIX in all 20 of 20 tests within 25 minutes, whereas the VRS-RTK got 19.

Table 11 Average and standard deviation of the TTFF and successfully fixed time within 25 minutes between the CLAS and the VRS-RTK with antenna tilted (n = 20)

	Ave. TTFF [min]	Std. TTFF [min]	FIX within 25 min.
CLAS	3.1	4.0	20
VRS-RTK	4.7	5.7	19

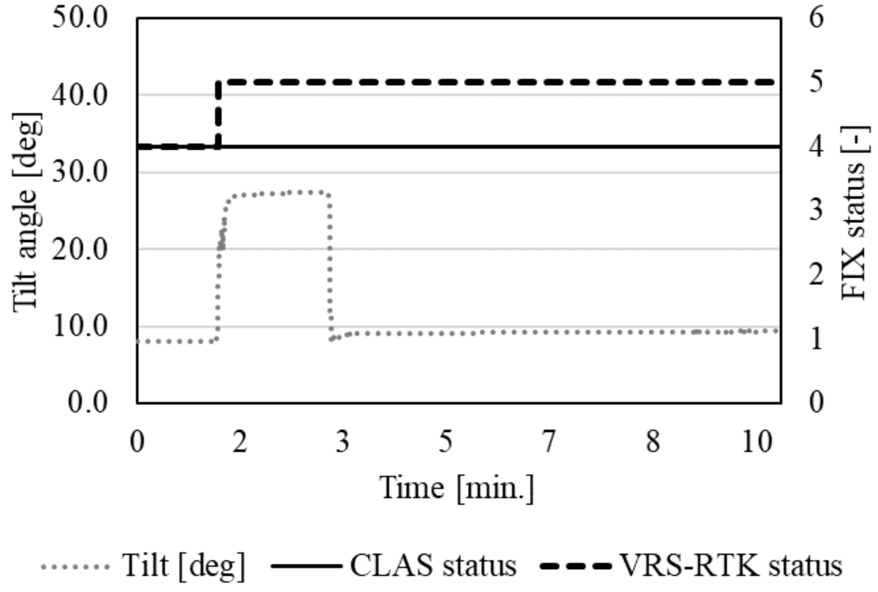


Fig. 34 Change of the FIX status of the CLAS and the VRS-RTK according to antenna tilt angles. Status 4 means obtaining a FIX solution while 5 is a FLOAT solution.

Fig. 34 shows the change in the FIX status when the antenna was tilted after acquisition of a FIX solution. The FIX status of the VRS-RTK changed from 4 (FIX) to 5 (FLOAT) when the antenna was tilted 17.9 degrees and then did not change back to 4 within 8 minutes. For the CLAS, on the other hand, the FIX status remained constant even when the antenna was tilted 27 degrees.

These results indicate the effect of the difference between the correction methods. The receivers of the CLAS and the VRS-RTK were the same model, so the performance of the calculation process was considered uniform. They could also receive the same types of satellite signals. Here, the VRS-RTK creates correction information based on the positioning data at four or more GPS-based control stations near the rover [51]. For the VRS-RTK theoretically, the rover needs to receive the satellite signals acquired by the GPS-based control stations at the same time. Therefore, the TTFF was longer or the FIX solution could not be calculated because the VRS-RTK rover could only catch different satellites from the GPS-based control stations due to the buildings or the antenna inclination. On the other hand, the CLAS utilizes augmentation information based on the positioning data from approximately 250 GPS-based control stations around Japan. Thus, neither buildings nor antenna inclination affected the creation of augmentation information for the CLAS even if a few satellite signals were shut out. This characteristic of the CLAS can contribute to smart farming utilizing robot vehicles in mountainous areas that have steep slopes and many windbreaks.

3.3.2. Accuracy and applicability of CLAS in a mountainous area

Table 12 shows the positioning accuracy at the eight data points in Section 3.2.2. For those points, the mean of the average error was 0.040 m and the error between the confidence intervals was less than 0.064 m, which corresponds to the officially verified value of 0.06 m.

Table 12 Accuracy of the CLAS

Measurement point	Average [m]	S.D. [m]	Ave. + 1.96S.D. [m]
1	0.048	0.021	0.089
2	0.042	0.015	0.070
3	0.015	0.004	0.022
4	0.049	0.012	0.072
5	0.057	0.017	0.091
6	0.037	0.009	0.055
7	0.037	0.008	0.053
8	0.036	0.013	0.061
Mean	0.040	0.012	0.064

Table 13 shows the TTFF of the CLAS and the VRS-RTK near the windbreaks shown in Fig. 30. The FIX solution could not be acquired by either CLAS or VRS-RTK within 4 minutes on point A, located on a headland of vine rows near a small windbreak, or on point C, located on a headland near a large windbreak. Meanwhile, the CLAS got a FIX solution earlier than the VRS-RTK on point B, which was on a farm road surrounded by a few trees. This result corresponded to that near buildings in Section 3.1. Even in actual farmland, trees affected the creation of correction information for the VRS-RTK, whereas trees did not affect the creation of augmentation information for the CLAS.

Table 13 Comparison of the TTFF [second] between the CLAS and the VRS-RTK

Measurement point	CLAS	VRS-RTK

A	-	-
B	105	318
C	-	-

Fig. 35 shows the results of static positioning of the CLAS, the VRS-RTK, and the Radio-RTK; their respective TTFF were 87 seconds, 97 seconds, and 10 seconds. The notable result was around 190 seconds in Fig. 35. Although the CLAS and the Radio-RTK each got a FIX solution, the VRS-RTK did not. This was definitely because the VRS-RTK could not get a correction signal due to a weak internet signal. The tested surroundings were located in an area where the intensity of the internet signal switched frequently between out of service and very weak according to the position of the mobile broadband receiver. This phenomenon, in which the correction signal for the VRS-RTK was hardly acquired, must occur in farmland that has no base station nearby for mobile broadband. In contrast, the CLAS needs no internet and can get a FIX solution stably even in mountainous farmland.

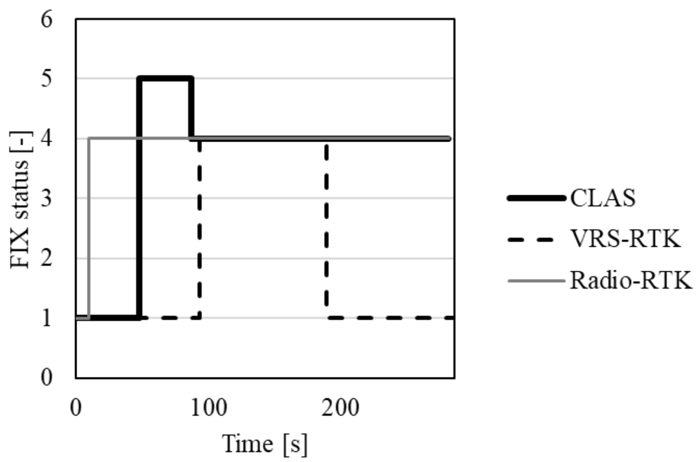


Fig. 35 FIX status between the CLAS, the VRS-RTK, and the Radio-RTK in time series. Status 1, 4, and 5 represent single, FIX, and FLOAT solutions, respectively.

3.3.3. Dynamic test

Table 14 shows the results of the trajectory error of the CLAS in Section 3.2.4. The error was less than 0.03 m RMS and did not differ between speeds of 1.0 m/s and 1.3 m/s. the maximum error was 0.076 m in the acquired data. This validation method ignored the error for the vehicle travel direction because trajectories, not points, were compared. However, lateral error is important for steering. If the positioning accuracy for the vehicle lateral direction is the same as the travel direction, the positioning accuracy would be 0.04 m RMS ($0.03 \times \sqrt{2}$). This result was almost the same as the accuracy of the static experiment in Section 3.3.2. Hence, under low speeds for a farm vehicle, the positioning accuracy was almost the same statically and dynamically. These results indicate that the CLAS is accurate enough as a navigation sensor for a robot farm vehicle.

Table 14 Trajectory errors between the CLAS and the VRS-RTK corresponding to two travel speeds

Average vehicle speed [m/s]	RMSE on path1 [m]	RMSE on path2 [m]	Max error on path1 [m]	Max error on path2 [m]
1.0	0.027	0.022	0.068	0.076
1.3	0.025	0.022	0.058	0.057

3.3.4. Accuracy of autonomous navigation

Fig. 36 and Table 15 show the results of autonomous driving of the robot vehicle navigated by the CLAS. The vehicle could travel and turn autonomously on two paths, which were 460 m long in total, with 0.03 m RMS error on path 1 and 0.09 m RMS error on path 2. There was no significant difference in error between the set speeds of 1.0 m/s and 1.3 m/s. The maximum lateral error was 0.20 m at 1.3 m/s on path 2. However, as the vehicle width was 1.7 m and the row width was 2.5 m, the vehicle did not collide with grape trellises. The reason why the lateral error was greater on path 2 than on path 1 was attributed to instability after turning and the observation error caused by the slope.

The RMS error and the maximum error of observed data between the CLAS and the RTK were less than 0.02 m. This result was almost the same as that in Section 3.3. The maximum slope of the tested field was less than 5 degrees in this experiment, and thus, the VRS-RTK could also be utilized. The robot vehicle navigated by the CLAS, however, can be effective in orchards that have around 15 degrees slopes and tall high trees from the results in Section 3.3.1 and 3.3.2. These results suggest that the CLAS can be utilized instead of the RTK as a navigation sensor for a robot vehicle in mountainous farmland.

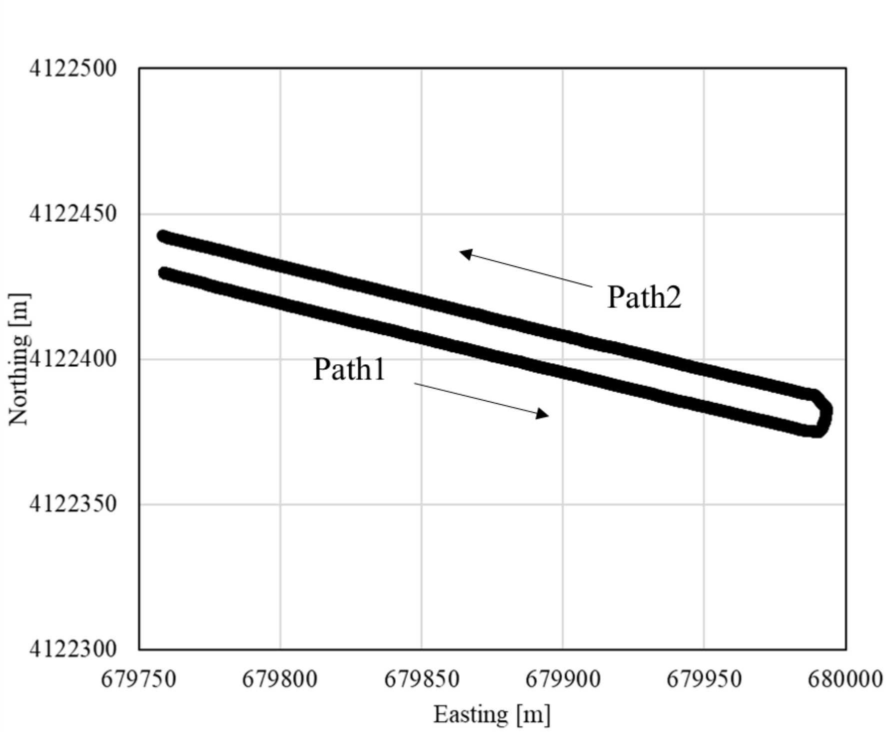


Fig. 36 Traveled trajectory of the robot vehicle controlled by the CLAS (https://youtu.be/wAIia-c7j-g)

Table 15 Lateral errors when the robot vehicle traveled autonomously in a vineyard navigated by the CLAS

Average vehicle speed [m/s]	Positioning system	Lateral RMSE [m]		Max error [m]	
		Path1	Path2	Path1	Path2
1.0	CLAS	0.05	0.07	0.15	0.17
	RTK	0.03	0.09	0.13	0.19
1.3	CLAS	0.04	0.08	0.13	0.19
	RTK	0.03	0.09	0.11	0.20

3.4. Conclusion of this chapter

In the presence of buildings, windbreaks, and antenna inclination, the CLAS had a more robust ability to get FIX solutions compared to the VRS-RTK, which is popular as a conventional navigation sensor. Moreover, the VRS-RTK was affected by the rover's internet environment, while the CLAS could be used anywhere because it did not depend on the internet. The robot vehicle could travel and turn autonomously using the CLAS in an actual vineyard in a mountainous area. The positioning accuracy of the CLAS was sufficient for a navigation sensor instead of the RTK. In conclusion, the CLAS was an appropriate navigation sensor for a robot vehicle in mountainous farmland.

Chapter 4. Optimal route planning

4.1. Introduction

Regarding a turning method in a vineyard, a path skip turn that skips paths and enters the next row can be adopted instead of the switchback turn that is popular in an upland crop field (Fig. 37). The path skip turn does not include backing operation, which results in an easy operation and applicability even if a vehicle has a large minimum turning radius. When performing the path skip turn, there are two methods: turning to a predetermined angle without a navigation map, or generating the navigation map for turning and following it. However, vineyards are often not accurate rectangles, so it is difficult to predetermine the turning angle. Therefore, it is necessary to dynamically generate a navigation map on turning from the relationship between the current and next path.

In this chapter, an optimal route planning method was devised. At the same time, a turning map connecting paths was generated in the real time. This turning map creating method can be applied even in irregularly shaped fields. Using both the optimal route planning and the turning map creating methods, the robot drove autonomously in a real vineyard.

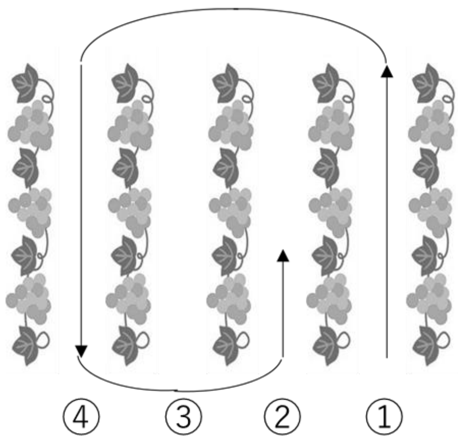


Fig. 37 The overview of the path skip turning method

4.2. Materials and methods

4.2.1. Optimal route planning

Let the rows of vineyards, which have *n* rows in total, be 1, 2, …, *n*, in order from the start to the end, and the sequence of travel routes be a_k ($a_k \in \mathbb{N}$ | k = 1, 2, …, n). When performing the path skip turn as shown in Fig. 37, the number of skips *p* from a_k to a_{k+1} is expressed by Eq. (19) when *k* is an odd number, while the number of skips *q* from a_k to a_{k+1} is expressed by Eq. (20) when k is an even number. Then, Eq. (21) always works.

$$p = |a_k - a_{k+1}| \; (k = 2u - 1 \mid u \in \mathbb{N}) \tag{19}$$

$$q = |a_k - a_{k+1}| \; (k = 2u \mid u \in \mathbb{N}) \tag{20}$$

$$p = q + 1 \tag{21}$$

Therefore, from the vine row width *w* and the minimum turning radius *r* of the robot, the minimum number of skips n_{min}, that is the minimum number of *q*, is expressed by Eq. (22) with the Gaussian symbol, that expresses the integer part of the quotient. Here, the minimum turning radius r is geometrically represented by Eq. (23) using the wheelbase L and the maximum steering angle δ_{max} from the bicycle model assuming the rear wheel is the turning center.

$$q \geq n_{min} = \left\lfloor \frac{2r}{w} \right\rfloor + 1 \tag{22}$$

$$r = \frac{L}{\tan \delta_{max}} \tag{23}$$

For example, if the total number of paths is n = 8, the first route is a_1 = 1, and the number of skips is q = 3, the number of skips is p = 4, so the travel route is $a_k = \{1, 5, 2, 6, 3, 7, 4, 8\}$. Here, if TRUE is the state in which the vehicle can travel on the route without excess or deficiency, while FALSE is the case where it is not TRUE, it is TRUE when n = 8. Even if n = 7, it is also TRUE if a_8 is deleted from a_k above. The condition to be TRUE depends on the relationship between p and n. Thus, if Eq. (24) is satisfied, it becomes TRUE inductively.

$$2p - 1 \leq n < 2p + 1 \tag{24}$$

Next, consider the case where n is greater than 8. From the above, if p increases according to the value of n, all paths can be traveled. However, the larger p is, the longer the distance traveled on the headland is, and thus, the loss of energy and time is more significant. Therefore, the loss is minimized by dividing the total route into units. If n = 15, it is possible to continue turning with the number of skips p = 4 by dividing n into n_1 = 8 and n_2 = 7. At this time, the turning direction is opposite after a_9. Let b_k be the set of this divided unit, and let m be the number of units ($k \in \mathbb{N}$, $b_k \in \mathbb{N}$ | k = 1, 2, …, m). In this case, $b_k = \{8, 7\}$ and m = 2, and Eq. (25) is always satisfied. Here, Eq. (24) does not work when $n(b_k) \geq 2$, but if b_1 satisfies Eq. (26), b_m satisfies Eq. (27), and other units satisfy Eq. (28), it is TRUE. For example, when p = 4, n = 22, $b_k = \{8, 7, 7\}$, and a_k at this time is represented by Eq. (29) and becomes TRUE. The reason why the relationship between b_k and p in Eq. (27) and Eq. (28) is different from Eq. (26) after b_2 is that the last path of each unit also serves as the first path of the next unit from the second unit onwards (Fig. 38).

$$n = \sum b_k \tag{25}$$

$$b_1 = 2p \tag{26}$$

$$2p - 2 \leq b_m < 2p \tag{27}$$

$$b_k = 2p - 1 \ (1 < k < m) \tag{28}$$

$$a_k = \begin{Bmatrix} 1, 5, 2, 6, 3, 7, 4, 8, \\ 12, 9, 13, 10, 14, 11, 15, \\ 19, 16, 20, 17, 21, 18, 22 \end{Bmatrix} \tag{29}$$

If the total paths divided into units as described above, the condition for TRUE can be changed by adjusting the number of skips p for each unit. For example, assuming n = 22, if the number of skips is p = 4 when b_1 and p = 8 when b_2, then $b_k = \{8, 14\}$, that is a different solution. In other words, this can be ascribed to the subset sum problem: whether the sum of the set b_k of natural numbers is divisible by the natural number n. However, b_k must satisfy Eq. (30) in this case.

$$b_k = \begin{cases} 2u & (k=1) \\ 2u-1 & (1<k<m) \end{cases}$$
$$b_k \geq 2q \quad (k=m)$$
$$(u \in \mathbb{N} \mid u > q) \tag{30}$$

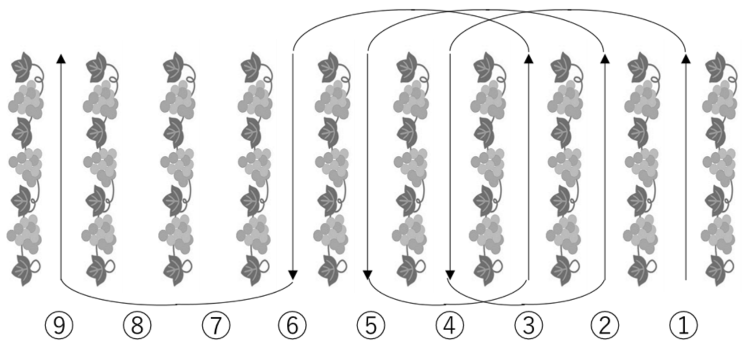


Fig. 38 A schematic diagram of transit from the first unit to the second unit

It is a well-known fact that the dynamic programming method is more effective than the brute force method in terms of the calculation cost of the subset sum problem. After finding candidates for b_k by solving the conditional subset sum problem by the dynamic programming method, b_k is selected so that the loss of the energy and time is minimized. Here, since the variable is only the travel distance in headlands that depends on the number of skips, the travel route based on b_k is determined to minimize the extra travel distance. Therefore, the loss function J_k is defined as shown in Eq. (31) in the case of b_1, Eq. (32) in b_k, Eq. (33) in b_m, and b_k is selected so that J Eq. (34) is minimized. Note that the number of skips in b_k is indicated by p_k and q_k, respectively. At this time, the total travel distance *d* in headlands is expressed by Eq. (35) using the vine row width *w* that is assumed to be constant.

$$J_1 = p_1 \times \frac{b_1}{2} + q_1 \times \left(\frac{b_1}{2} - 1\right) \tag{31}$$

$$J_k = p_k \times \left\lfloor \frac{b_k}{2} \right\rfloor + q_k \times \left\lfloor \frac{b_k}{2} \right\rfloor \quad (1 < k < m) \tag{32}$$

$$J_m = \begin{cases} p_k \times \left(\frac{b_k}{2} - 1\right) + q_k \times \frac{b_k}{2} & (b_k = 2u) \\ p_k \times \left\lfloor\frac{b_k}{2}\right\rfloor + q_k \times \left\lceil\frac{b_k}{2}\right\rceil & (b_k = 2u + 1) \end{cases} \tag{33}$$

$$J = \sum_{k=1}^{m} J_k \tag{34}$$

$$d = J \times w \tag{35}$$

4.2.2. Turning path creation

Considering the irregularly shaped field, a turning path is created. Now, we consider a turn from path 1 to path 4 in Fig. 39. Each path consists of navigation points ω_{ij} ($\omega \in \mathbb{R}^3$) (i is the path sequence, and j is the point number in the path) with three dimensions: latitude, longitude, and work flag (working or idling). The navigation points of black circles indicate the state of working, while the white circles represent the state of idling. The end navigation point of the path is defined on the center of the straight line connecting the end points between the vine rows. Then, the extended navigation point P_0 is added, which is far from the end of the path by the implement offset d_I. The implement offset is the distance between the GNSS antenna and the work implement, which is a constant value depending on the vehicle. Next, the turning path starts at the point P_1 that is extended from P_0 by the idling offset d_o. Let the path before turning be i_o, the path after turning be i_n, and all paths between them be i_k ($o < k < n$), then the point P_k is placed at the extension of the headland distance d_h from the end of i_k. The headland distance is a constant value depending on the vineyard, assuming that the distance between all paths and the field boundary does not vary. Finally, a point P_n is placed at the end of path i_n, which is extended by a distance d_r considering the turning radius, field slope, etc. The above procedure was performed in real time while driving. The user can adjust d_o, d_h, and d_r to lead the robot to travel accurately.

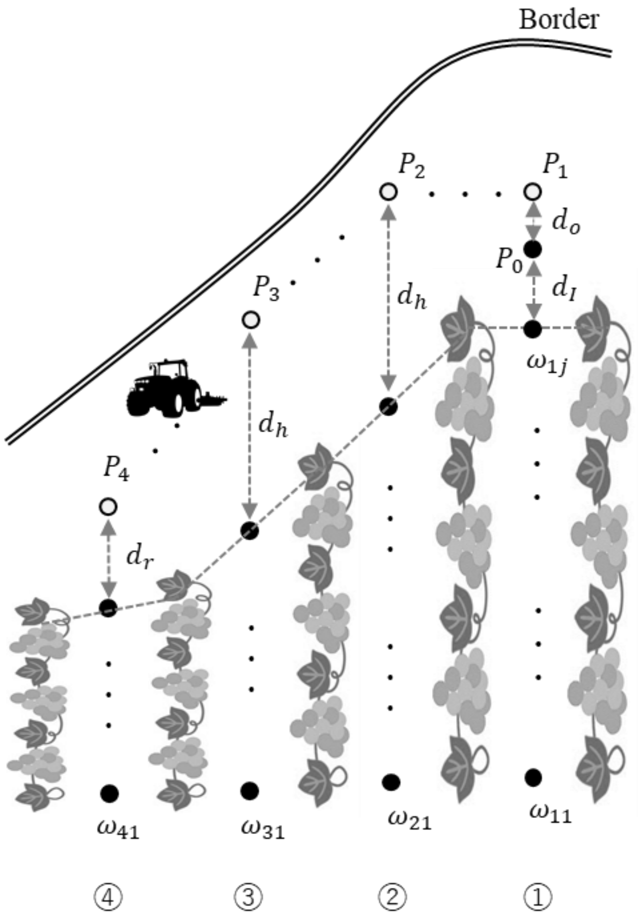


Fig. 39 The overview of the turning path creation

4.2.3. Field trial

Route planning and autonomous traveling tests were conducted on a plot at the Tsurunuma Winery in Urausu, Hokkaido, Japan. There were 43 paths in total, with a width of 2.5 m between vine rows, and each path was about 90 m to 130 m long, covering an area of approximately 12000 m^2. The steering control method was the same as described in Section 3.2.5.

4.3. Results and discussion

4.3.1. Optimal route planning

From the conditions of the tested vineyard and vehicle, distance between vine rows w = 2.5 m, maximum steering angle δ = 35 degrees and wheelbase L = 2.0 m, then the minimum number of skips q = 3 from Eq. (22). When the range of the number of skips is $3 \leq q \leq 6$, Eq. (24) shows that the question can be solved in one unit when the total number of paths n is between 7 and 14. As a result of calculation for the total number of paths n from 15 to 50, it was computable for all paths n. As an example, the set of units b_k for $30 < n \leq 50$ is shown in Table 16.

Table 16 Results of optimal path planning

Total path	31	32	33	34	35
b_k	8 7 9 7	8 7 9 8	8 9 9 7	8 9 9 8	8 9 9 9
Total path	36	37	38	39	40
b_k	8 7 7 7 7	8 7 7 7 8	8 7 7 9 7	8 7 7 9 8	8 7 9 9 7
Total path	41	42	43	44	45
b_k	8 7 9 9 8	8 9 9 9 7	8 7 7 7 7 7	8 7 7 7 7 8	8 7 7 7 9 7
Total path	46	47	48	49	50
b_k	8 7 7 7 9 8	8 7 7 9 9 7	8 7 7 9 9 8	8 7 9 9 9 7	8 7 7 7 7 7 7

Since the total number of paths n in the tested vineyard was 43, specific results were obtained as shown in Table 17.

Table 17 The result of route planning assuming n = 43

b_k line up	Headland distance [m]	Optimal b_k	Path plan
8 7 7 7 7 7	325	**8 7 7 7 7 7**	1 5 2 6 3 7 4 8
8 7 7 13 8	440		12 9 13 10 14 11 15
8 7 9 11 8	420		19 16 20 17 21 18 22
• • •	• • •		26 23 27 24 28 25 29
14 9 9 11	530		33 30 34 31 35 32 36
14 7 9 13	550		40 37 41 38 42 39 43

Due to the minimum number of skips q = 3, if the total number of paths n is $n < 2q + 1$ (n < 7), this method cannot be used to solve the problem, but if $7 \leq n \leq 50$, the optimal route can be calculated. The proposed optimal route planning method is effective when the turning radius is large and the path skip turn is thus required.

4.3.2. Traveling result

Fig. 40 shows the results of the autonomous operation. In this experiment, the minimum number of skips q = 4, and the driving was stopped in the middle to take the charging capacity into account, so the driving route $a_k = \{1, 6, 2, 7, 3, 8, 4, 9, 5, 10, 16, 11\}$. As shown in Fig. 40, the robot could travel 12 paths consecutively without collision with vines in an irregularly shaped vineyard. The total traveled distance was about 1.6 km, and the traveled time was about 28 minutes. Table 18 shows the root mean square error (RMSE) on each straight path and each turning path. The average value of the RMSE were 0.07 m and 0.62 m, in straight and turning paths of all 12 paths, respectively.

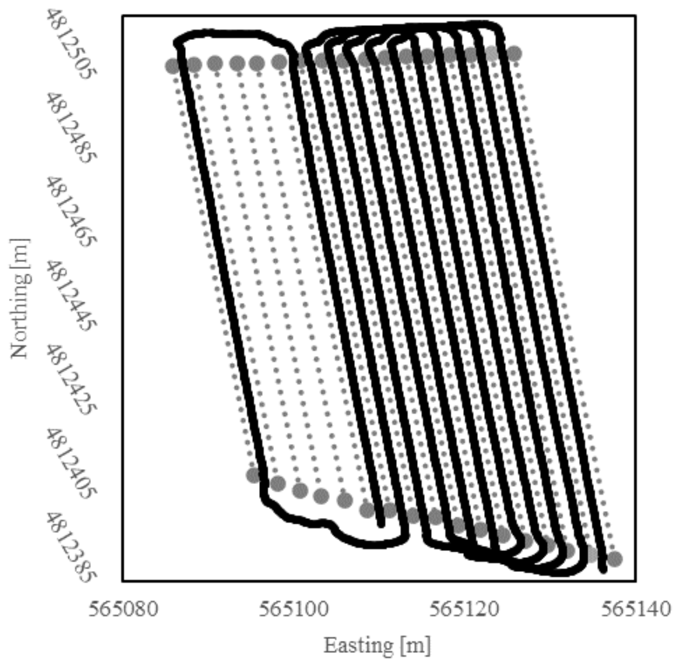


Fig. 40 The traveled trajectory of the robot and vine rows

Table 18 The traveling results on each path

Straight path	RMSE [m]	Turn path	RMSE [m]
1	0.07	1--->6	0.55
6	0.06	6--->2	0.53
2	0.08	2--->7	0.50
7	0.07	7--->3	0.44
3	0.09	3--->8	0.46
8	0.08	8--->4	0.74
4	0.04	4--->9	0.53
9	0.06	9--->5	0.60
5	0.04	5--->10	0.49
10	0.06	10--->16	1.32
16	0.08	16--->11	0.63

11	0.05		
Ave.	0.07	Ave.	0.62

In this turning method, we could observe that the lateral deviation was large at the beginning of the turning, and then it converged gradually. Thus, in this experiment, the robot could crash into the next vine row if the lateral deviation did not converge. This is why the minimum number of skips in this experiment was set to $q = 4$ instead of the theoretical value of $q = 3$ to recover the lateral deviation. The result indicates that the RMSE from the path 10 to 16 on the turning path was the largest, and was over 1 m. Fig. 41 shows the behavior at that time between the end of the path 10 and the beginning of the path 16. The situation where the lateral deviation was large at the beginning of turning is observed. The same trend was discovered on other turning paths. However, from the path 10 to 16, the turning radius of the robot was relatively large because the turning path at the beginning was the acute angle according to the irregularly shaped vineyard, resulting in a significant lateral deviation. In addition, the lateral slip due to a lateral slope around the turning path was considered another important cause. We have to consider some measures such as setting the turning path like a circular arc according to the vehicle dynamic model or modifying the steering control method taking into account the slope of the field. However, it is difficult to estimate turning radius determinately due to soil conditions or lateral slopes, so a method to estimate it statistically using travel data should be considered as well as the vehicle dynamic model.

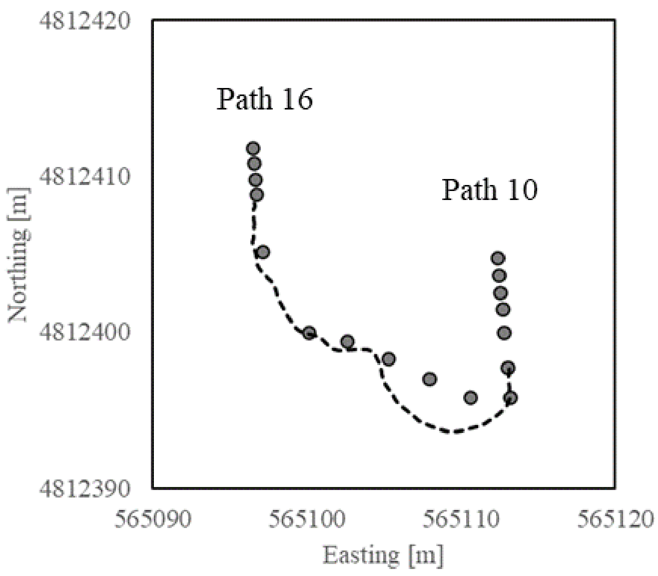


Fig. 41 The traveled trajectory (dashed line) and the navigation map points between path 10 and 16

4.4. Conclusion of this chapter

An effective path skip turn in an irregularly shaped vineyard was devised, the optimal route planning according to the path skip turn was suggested, and autonomous driving was tested in the actual vineyard. The path skip turn method worked well for the tested field, and the robot was able to travel 12 paths, about 1.6 km, without collision with vines. As the lateral deviation was large at the beginning of turning, the turning path should be created based on a turning radius of the vehicle that is estimated by the vehicle model and travel data.

Chapter 5. Turning path planning considering a towed implement

5.1. Introduction

5.1.1. What is a turning path?

Many studies on robot vehicles traveling autonomously in farmland have been conducted. Especially, those robot vehicles for paddy or upland crop fields are commercialized. The robot vehicles commonly turn in such fields with a switchback or a keyhole turning method and enter the next rows in order [42]. On the other hand, it is different in orchards. Orchards have trees fixed on the ground, resulting in the headland distance being fixed. Orchard tasks such as mowing or spraying are sometimes conducted by a towed implement. In this case, going backward is not an easy procedure. Therefore, even for manual operation, a path skip turning is used [52]. In this study, we discuss the turning strategies of a robot vehicle with a towed implement in orchards. There are two ways to turn: with or without a target path. If you do not use the target path, the robot vehicle is controlled based on the GNSS or the IMU, so the heading and traveling distance reach the ones decided in advance. This method is easy to implement and used for the switchback turning. However, the final heading or positioning error is significant due to the sensor error and control delay. If you use the target path, the robot vehicle is controlled according to it. Thus, it is relatively robust to the sensor error or the control delay. Instead, the target path and a controller to follow the path are more important for the final state. This study focuses on the target path creation because a highly accurate turning method is required for the robot vehicle with a towed implement. Note that the robot vehicle only goes forward.

5.1.2. Path planning problem

Path planning problems for agricultural robots are classified into two types: point-to-point path planning and coverage path planning [53]. The point-to-point path planning generates a path that does not collide with obstacles when given initial and final positions, which is sometimes called a collision-free path planning problem in the robotics sector. The coverage path planning, on the other hand, generates a set of paths that cover all the surfaces without duplication and do not collide with obstacles when a field is given. In the agricultural sector, coverage path planning is much more studied because it is necessary to plan work paths on a field. This study, however, focused on the point-to-point path planning method to consider the turning path for a robot vehicle in an orchard. When planning path by determining the steering input at each time, the issues are how to give the steering input and how to model the behavior of the vehicle based on the steering input. Noguchi and Terao, 1994 and Noguchi and Terao, 1997 report that a series of steering inputs are determined using the genetic algorithm (GA) [54],[55]. In [55], the order of steering inputs at 1-second intervals for a maximum of 100 seconds is semi-optimized using the GA so that the final position and heading of the vehicle match the desired ones. Additionally, based on data on a tractor traveling on an asphalt road, the behavior of the vehicle in response to steering inputs was modeled by nonlinear regression using the neural network (NN). Kise et al., 2002 uses the spline curve to generate a forward turning path [56]. The steering input was assumed to be circular motion with constant input, and turning was performed by switching between three circles. If the curvature of the path generated by the spline function is smaller than the minimum turning radius of the tractor, it is recalculated. Linker and Blass, 2008 [57] used the A^* (pronounced A star) search, first proposed in 1968 [58], to plan a path in an orchard. A^* search is one of the graph search algorithms widely used for path planning in the robotics sector. In the case of considering a robot moving on a plane, the field is divided into cells (sometimes called grids or nodes), and the cells around the starting point that can be moved are opened. At this time, the movement is restricted by removing cells that cannot be moved due to the vehicle kinematics or obstacle existing. This method basically sets two costs: a moving cost according to moving from the current to the next cell, and a heuristic cost that is usually set by distance from the next to the goal cell, and the cell is sequentially selected with the lowest costs. Linker and Blass, 2008 [57] simulated with the cell size of 1 m by 1 m and a vehicle of 2.5 m by 1.0 m. They considered the moving cost on slopes in addition to the combination of forward and backward movement and steering input. The vehicle behavior was also considered by including the heading to the goal point as well as the distance in the heuristic cost. However, the vehicle model is too simple. The steering input is a maximum of 15 degrees in 5 degrees increments, and change in vehicle heading is expressed by simply substituting the steering input angle without considering the vehicle speed, wheelbase, slippage. Also, as they mention in their paper, although the A^* method calculates an optimal solution, there are many weighting parameters for costs specified by the user, and the results are influenced by their tuning.

5.1.3. Purpose of this study

Except for Noguchi et al.'s example of modeling vehicle motion using the NN, there are few studies that consider vehicle slippage. It is important whether a vehicle actually can follow the generated path. Furthermore, there are few examples of path planning taking into consideration the behavior of a towed implement. This study considered two methods of turning path planning in an orchard with a trellis type of trees. One method was based on just connecting arcs and straight lines without considering a towed implement (Strategy A). The other method considered a kinematic model of a vehicle and a di-wheeled towed implement so both the vehicle and the implement do not collide with obstacles such as the trellis, the trees, or the farmland borders (Strategy B). In Strategy B, the pattern of steering inputs was determined by a semi-optimized method using the GA. However, the variables were not the steering angles but the time of three patterns of steering input: straight, maximum left, and maximum right. Compared to Noguchi et al.'s method of solving a series of steering inputs using GA, the method in this study has the advantage of being able to eliminate unnecessary solutions and reduce calculation costs, as well as being able to suppress changes in steering inputs in advance. Moreover, the slip ratio was introduced for the vehicle behavior to be modeled based on actual driving data in an orchard. The two strategies were compared and verified.

5.2. Materials and methods

5.2.1. Turning strategies

5.2.1.1. 2 arcs (Strategy A)

Strategy A only considered a minimum turning radius of the front vehicle and the shape of the orchard. The turning path in Strategy A is basically composed of two arcs of the turning radius *r* and straight lines. Considering the shape of the orchard is not an accurate rectangle, points extended from the end of the paths skipped to headland distance d_h were utilized. Now we consider that the robot vehicle turns from the path *n* to the path *n*+*k*+1 as shown in Fig. 8. Note that *k* is the number of skips. Fig. 8 shows the example when $n = 1$ and $k = 4$. Let line $l_{2,1}$ be a line passing through the endpoint ω_2 of the path *n* and the second endpoint ω_1. Let points $\{\omega_3,\ \omega_4, \ldots,\omega_{3+k-1}\}$ be extended points of the skipped paths. Here, assuming $k > 2$, let $l_{1,1}$ be the line passing through the point ω_3 and the point ω_4, and let C_1 be a circle with radius *r* that is tangent to straight line $l_{1,1}$ and $l_{2,1}$, and connect the line and the arc at the point of contact. Similarly, the straight line passing through the starting point ω_{n+k+2} of path $n+k+1$ and the next point ω_{n+k+3} is $l_{1,2}$, and passing through the point ω_{n+k+1} and the point ω_{n+k} is $l_{2,2}$, and an arc is generated and connected. The turning path created in this way only takes into account the turning radius of the front vehicle, the width of the headland, and the shape of the tree rows. Furthermore, control points are not considered, and if the center of gravity is taken as the control point, there is no guarantee that the front vehicle or the rear implement will not collide with a tree rows. All we have done is create a path that the vehicle can travel by making full use of the width of the headland.

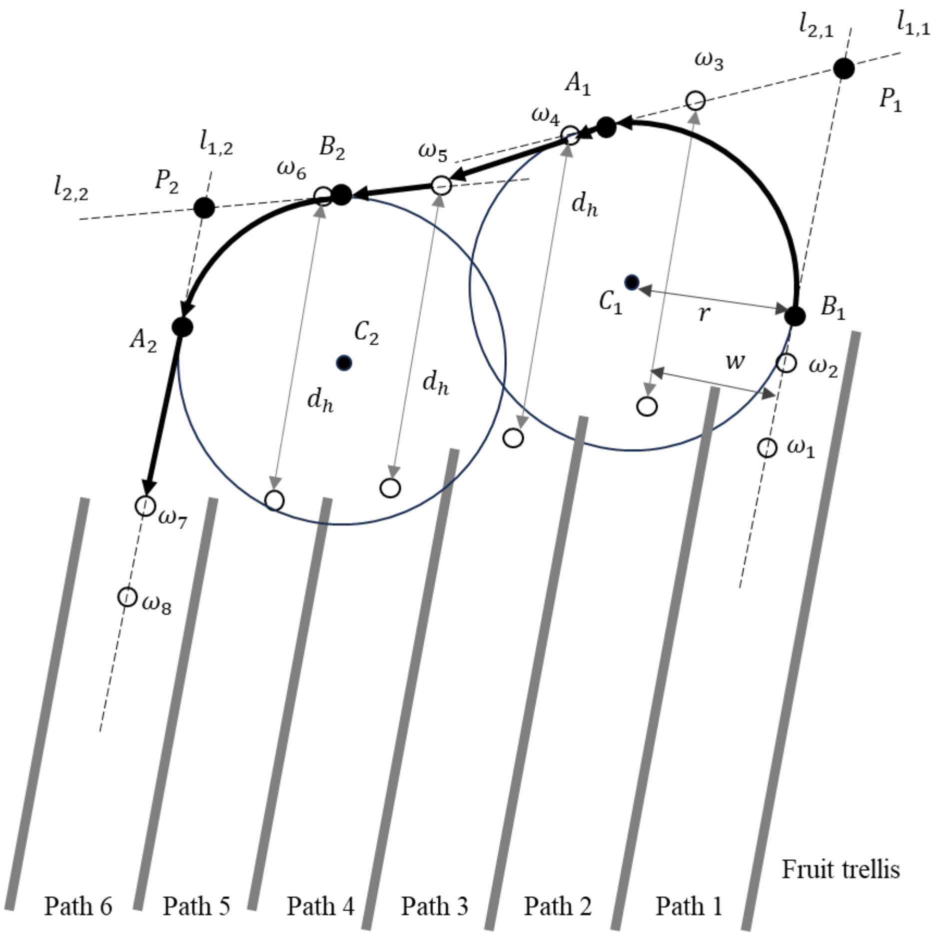


Fig. 42 Overview of the strategy A

5.2.1.2. Model based (Strategy B)

To solve the above problem, we need a turning path that allows the vehicle in front and the towed implement to travel without colliding with the tree rows. In other words, this problem is attributed to the problem of searching for a path in which the vehicle can travel from a starting point to a goal point, while neither the front vehicle nor the towed implement can travel without crashing the farmland boundary and the tree rows.

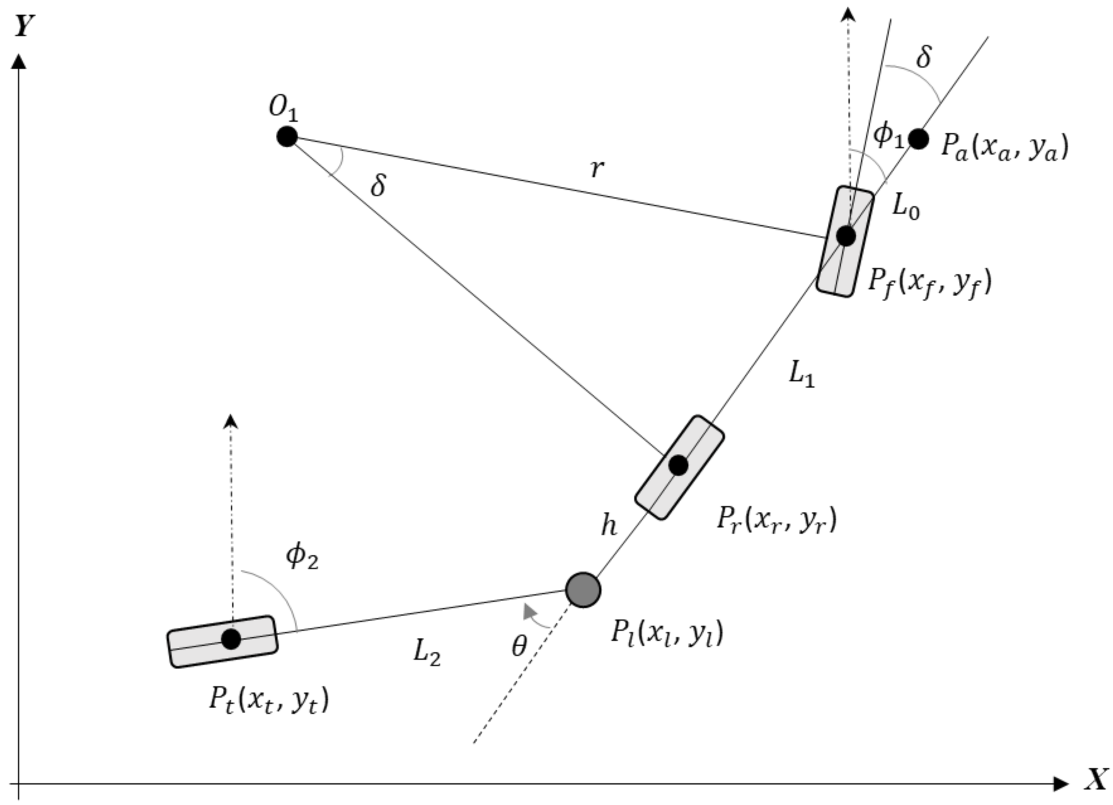


Fig. 43 A bicycle model of a tractor-trailer

From the equivalent bicycle model in Fig. 43, the behavior at the point of the front wheel $P_f(x_f, y_f)$ of the front vehicle in the world coordinates is expressed as Eqs. (36)(37)(38). Note that ϕ_1 is the heading of the front vehicle, δ is the steering angle of the front vehicle, v is the vehicle speed, L_1 is the wheelbase of the front vehicle.

$$\dot{x_f} = vsin(\phi_1 + \delta) \tag{36}$$

$$\dot{y_f} = vcos(\phi_1 + \delta) \tag{37}$$

$$\dot{\phi}_1 = v\frac{\sin\delta}{L_1} \tag{38}$$

The relationship between the position of the forefront of the front vehicle $P_a(x_a, y_a)$, the point of the rear wheel of the front vehicle $P_r(x_r, y_r)$, the linkage point between the vehicle and the implement $P_l(x_l, y_l)$, the point of implement's wheel $P_t(x_t, y_t)$, and P_f is as follows.

$$\begin{bmatrix} x_a \\ y_a \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \end{bmatrix} + L_0 \begin{bmatrix} sin\phi_1 \\ \cos\phi_1 \end{bmatrix} \tag{39}$$

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \end{bmatrix} - L_1 \begin{bmatrix} sin\phi_1 \\ \cos\phi_1 \end{bmatrix} \tag{40}$$

$$\begin{bmatrix} x_l \\ y_l \end{bmatrix} = \begin{bmatrix} x_f \\ y_f \end{bmatrix} - (L_1 + h) \begin{bmatrix} sin\phi_1 \\ \cos\phi_1 \end{bmatrix} \tag{41}$$

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x_l \\ y_l \end{bmatrix} - L_2 \begin{bmatrix} sin\phi_2 \\ \cos\phi_2 \end{bmatrix} \tag{42}$$

$$\phi_2 = \phi_1 + \theta \tag{43}$$

Here θ is the angle between the front vehicle and the towed implement, called the linked angle. [59] indicated θ is represented as Eq. (44).

$$\dot{\theta} = -\frac{v}{L_2}sin\,\theta - \frac{v\,tan\,\delta}{L_1}\left(\frac{h}{L_2}cos\,\theta + 1\right) \tag{44}$$

[60] also used a similar equation that trigonometric functions in Eq.(44) were linearized as a kinematic model of the towed implement. They compared a dynamic model derived from force with the kinematic model, and concluded that there was no difference between those

models when the speed was 0 – 4.5 m/s and input steering rate was 0 – 1 rad/s. Fig. 44 shows the tested vehicle and the towed implement: $L_0 = 0.3$, $L_1 = 2.1$, $h = 0.62$, $L_2 = 1.55$.

Fig. 44 An overview of the vehicle and the towed implement

Using the above model, we obtain the vehicle and implement trajectories and search for a turning path that allows both the frontmost position of the vehicle and the towed implement to travel without touching the tree rows and the farmland boundary. As you can see from the above model, the trajectory of the vehicle is determined by the speed v and the steering angle δ. Therefore, if arbitrary v and δ that the vehicle can achieve are given for each control cycle, a trajectory can be generated through repeated calculations. However, when considering arbitrary inputs without any constraints, the number of combinations becomes enormous. Therefore, we considered using empirical rules to narrow down the possible behavior of vehicles by creating patterns as shown in Fig. 45. The start point is the final point of the current path, and the goal point is the first point of the next path. The boundary is a point extending from the start point and goal point by the length of the headland L, and is a line passing through these two points. Fig. 45 shows a schematic diagram of a left turn, which consists of seven sections: going straight, turning left, going straight, turning right, turning left, turning right, and going straight. When each time is defined as t_k $(k = 1, 2, \dots, 7)$, the path's shape changes by adjusting the time t_k. Input steering angle is changed like Fig. 46. Generally, there is a delay between inputting the steering angle and reaching that state, so the steering angle is not rectangular but trapezoidal, as shown in Fig. 46.

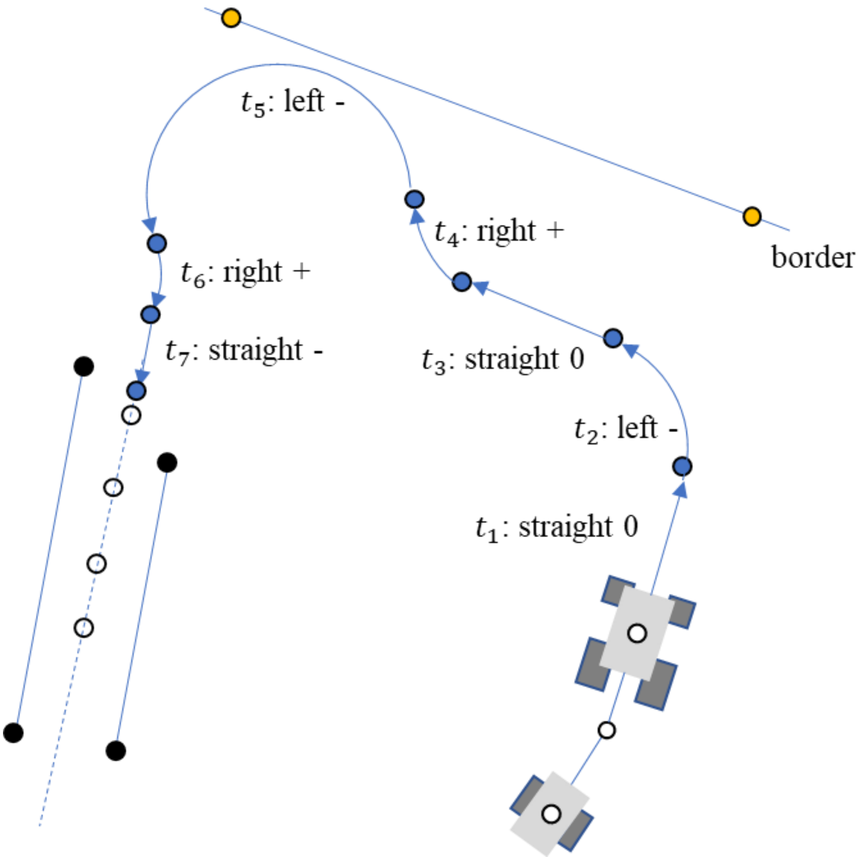


Fig. 45 A diagram of the patterned path trajectory in the case of turning to left

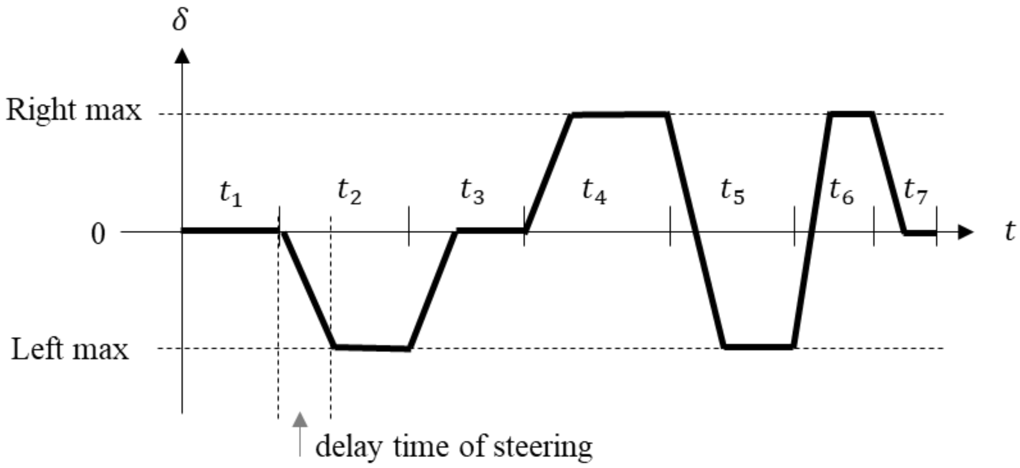


Fig. 46 Input steering angle

Consider how to determine seven variables $t_1, t_2, \ldots, t_7$. At the goal point, the position and heading of the vehicle should be along the following path, and the front vehicle and the towing implement must not collide with the tree rows or the farm boundary while its traveling. Therefore, the evaluation function *J* is set as Eq.(45). Note that a is a weighting coefficient, $\varepsilon_{vehicle}$ [m] is the lateral error between the forefront of the front vehicle and the goal point, $\varepsilon_{trailer}$ [m] is the lateral error between the towed implement and the goal point, φ [deg] is the heading error between the vehicle heading and the path heading, and ζ is a variable that takes a binary value of ∞ if contact occurs between the front vehicle or the towed implement and the tree rows or farm boundary, and 0 otherwise. A margin width ρ [m] was given, and if the distance was less than that, it was considered a collision. In this study, $\rho = 0.6$ and $a = 0.1$.

$$J = \varepsilon_{vehicle} + \varepsilon_{trailer} + a\varphi + \zeta \tag{45}$$

By iterated calculations in discrete times (0.1 [s]) of $0 \leq t_k \leq 10$ [s], a combination of seven variables $t_1, t_2, \ldots, t_7$ that minimizes this evaluation function is derived. However, even though the variables were reduced to seven, the number of combinations would be too large to perform the brute force approach, which would take time. In this study, we used the genetic algorithm (GA) to solve this optimization problem. In the GA, the population was 500, the number of generations was 200, the elite selection rate was 20%, the tournament selection rate was 20%, and of the remaining 60%, the probability of mutation was 30% and the probability of hybridization was 70%. When the evaluation function became smaller than a threshold, it was considered to have converged, and when it did not converge, calculations were repeated under the same conditions.

5.2.2. Control method

Section 5.2.1 presented a method for generating a turning path. We will explain how to control the vehicle, so it follows the path. Since the path was created based on the vehicle model, if the model accuracy was high, it would be possible to control it even in an open loop. However, in off-road and sloping environments, the model can have an error because of the slippage. Furthermore, it is difficult to keep the speed strictly constant, resulting in errors. Therefore, as shown in Fig. 47, we use a controller that feeds back the errors between the created path and the current state. This method, used in [49], determines the steering input by giving proportional gains a and b to the lateral deviation d [m] and the heading error $\Delta\phi$ [deg] (Eq. (46)).

$$\delta = ad + b\Delta\phi \tag{46}$$

Note that since the heading error $\Delta\phi$ depends on the look ahead distance L [m], there are three tuning parameters: the proportional gains a, b and L. If L becomes smaller, the vehicle will follow a curve line more precisely, but if it is too small, it will diverge. In this study, we used a brute force method to find a combination of three parameters that would minimize the root mean square (RMS) error in the lateral deviation in simulation. Based on prior information, we empirically found that $0 < a < 80$, $0 < b < 2.0$ and $0.3 < L < 5.0$, so we optimized by increasing a in 1 increment and b and L in 0.1 increments within this range.

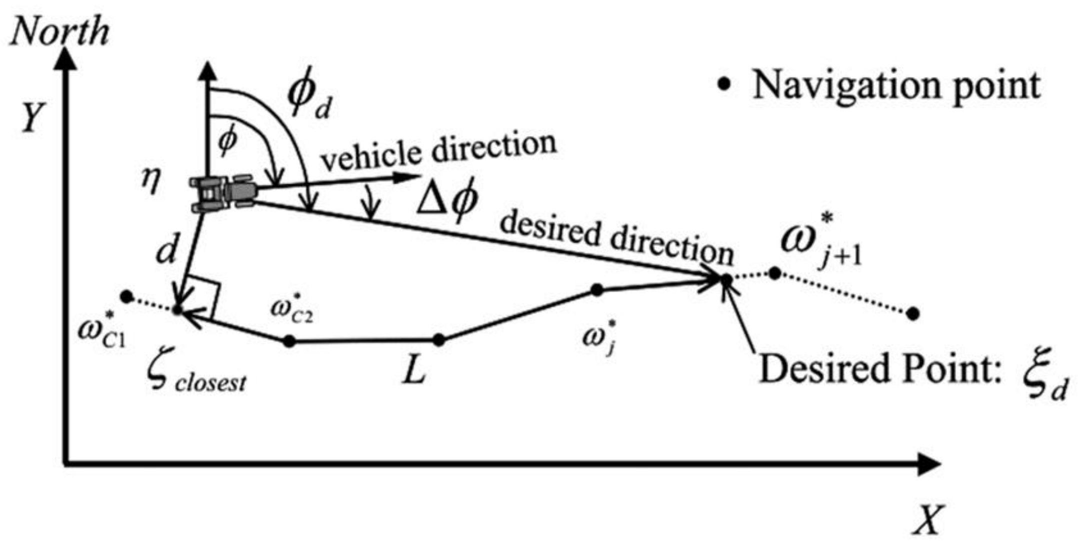


Fig. 47 Definition of a navigation map (Source: [50])

5.2.3. Vehicle property

Considering the vehicle behavior based on the model as described above, an important point is whether the vehicle behaves in accordance with the model in the real environment. Particularly, sensor errors in steering angle and errors due to the slippage can be observed. Since these errors largely depend on vehicle characteristics, we clarified the characteristics of the tested vehicle.

5.2.3.1. Steering sensor calibration

The tested vehicle was equipped with a potentiometer to sense the steering angle, so that the motor was controlled according to the input steering angle to reach the desired angle. However, the maximum steering angle and 0 point of the potentiometer are not accurate because they are adjusted manually. Therefore, a constant radius turning at a speed of 1 m/s on a flat, hard concrete road surface was conducted with recording the trajectory using the GNSS, to estimate the actual steering angle. The input steering angles were 15 degrees, 30 degrees, and the maximum steering angle of 35 degrees, and a total of 6 trials were performed to the left and right. AsteRx SB3, Septentorio was used as the GNSS receiver. The positioning accuracy of it is 6 cm by the CLAS, one of the PPP-RTK method [61]. Here, as shown in Fig. 48, the GNSS antenna was mounted ahead of the front wheel, so the measured turning radius was R_3. From the kinematic model, R_3 was expressed as Eq. (47).

$$R_3 = \frac{L_0 + L_1}{sin\,\alpha} \tag{47}$$

Here, since the relationship between R_1 and R_3 is Eq. (48), the actual steering angle δ can finally be derived as shown in Eq. (49). Note that the actual steering angle at this time is a value including tire slippage actually, but considering the hard concrete road surface and low speed, the slippage is assumed to be small and ignored here.

$$R_1 = R_3 \cos \alpha \tag{48}$$

$$\delta = \tan^{-1} \frac{L_1}{R_1} \tag{49}$$

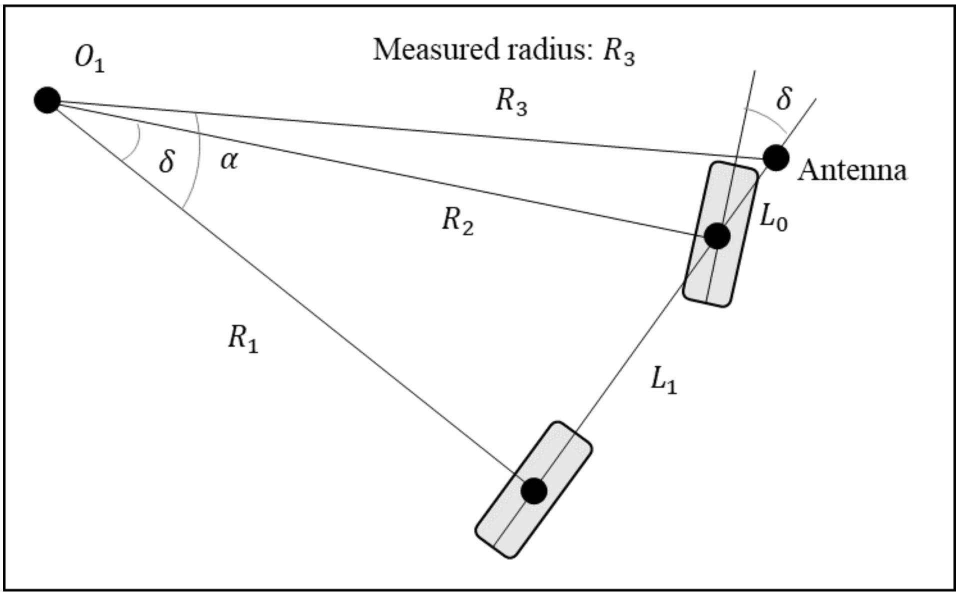


Fig. 48 A bicycle model. The GNSS antenna was ahead of the front tire.

In addition, in the simulation, realizing the steering response speed is important. The steering rate and the dead time were estimated from the actual steering input and output relationship. On the other hand, as mentioned above, the steering motor is feedback-controlled according to the potentiometer, so the input and output relationship was expected to be a second order or higher delay system. The steering input and output system was modeled by the second order + the dead time (Eq. (50)). The unknown coefficients ζ, ω_n and L of the second order lag system and dead time were repeatedly calculated numerically, and the values at which the average error was the lowest were used.

$$G(s) = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2} e^{-Ls} \tag{50}$$

5.2.3.2. Slip characteristics

In orchards, the vehicle travel on surfaces covered with soil and grass, so the slippage can be expected. In particular, we concerned that the actual turning radius will be larger than the modeled turning radius due to slipping when turning due to mud. The vehicle may slip due to the force by the towed implement due to the slope. In this case, the turning path created in Section 5.2.1 becomes unfollowable. Therefore, we conducted a traveling test in an orchard environment, recorded the traveling trajectory using the GNSS, and verified the actual amount of slippage. In this study, following [62], a slip ratio γ as shown in Eq. (51) was introduced. Given the slip angle β, it is proportional to the actual steering angle δ multiplied by the slip ratio γ. $\gamma = 1$ indicates no slip, and $\gamma < 1$ indicates the understeer characteristics.

$$\beta = \gamma\delta \tag{51}$$

The estimated trajectory substituted β for δ in Eqs. (36), (37) and (38) was compared with the actual traveling trajectory by changing the slip ratio γ in 0.001 increments to achieve the best match. The vehicle speed was measured using the GNSS. Additionally, to confirm whether the slip ratio changes with inclination, an IMU (VN100, VectorNav, USA) was installed in the vehicle and the inclination angle was measured.

5.3. Results and discussion

5.3.1. Steering characteristics

Fig. 49 shows the results of the steering sensor calibration performed in Section 5.2.3.1. The horizontal axis in Fig. 49 is the input steering angle, and the vertical axis is the actual steering angle derived from the constant radius turning. The maximum value of the input steering angle was 35 degrees according to the configuration, but actually it was around 30 degrees. The actual steering angle was calibrated by substituting the steering angle for *x* in Eq. (52).

$$\hat{\delta} = 0.8679x - 0.3082 \tag{52}$$

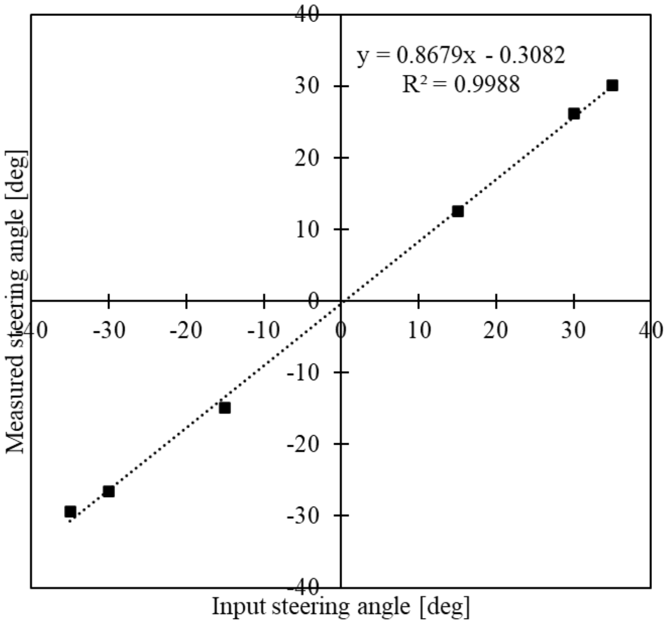


Fig. 49 Relationship between input and measured steering angles

Next, Fig. 50 shows the relationship between input and output of steering angle. Fig. 51 shows the estimated output for the linear and the second-order lag systems. The unknown coefficients of the linear equation were the steering rate 28 [deg/s] and the dead time 0.3 [s]. The unknown coefficients of the second-order lag system were the damping coefficient $\zeta = 2.81$, the natural angular frequency $\omega_n = 0.58$, and the dead time $L = 0.1$. The average error of the linear system was 0.9 degrees and that of the second-order lag system was 1.2 degrees. The true output showed that the response speed was fast and overshoots when responding to a step input, which indicates that it is a third-order or higher-order lag system. This can be the cause of the approximation error due to second-order lag. In this simulation, we used the linear formula with smaller errors.

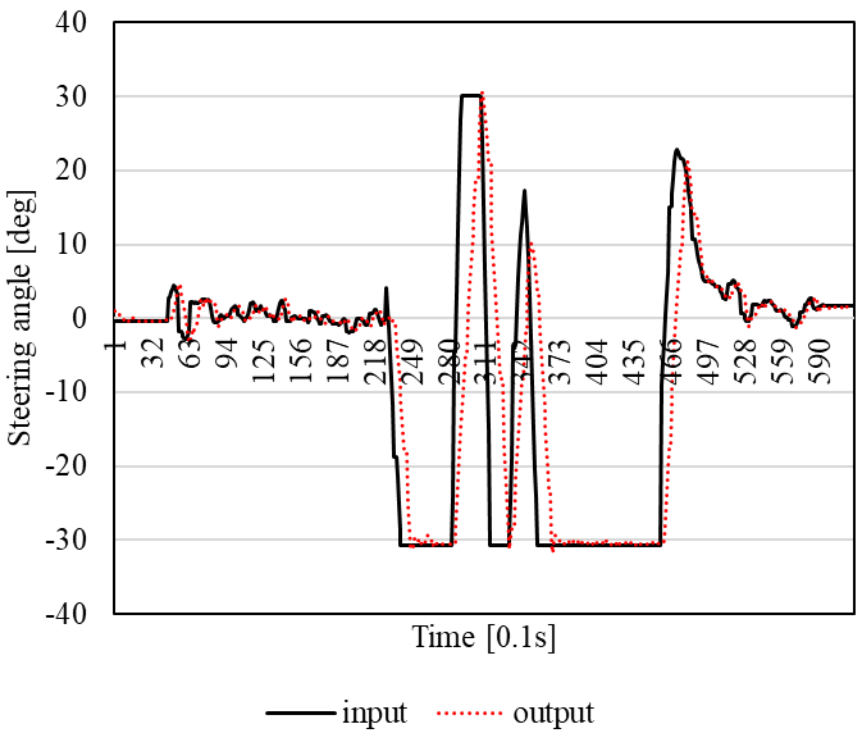


Fig. 50 Relationship between the input and the true output

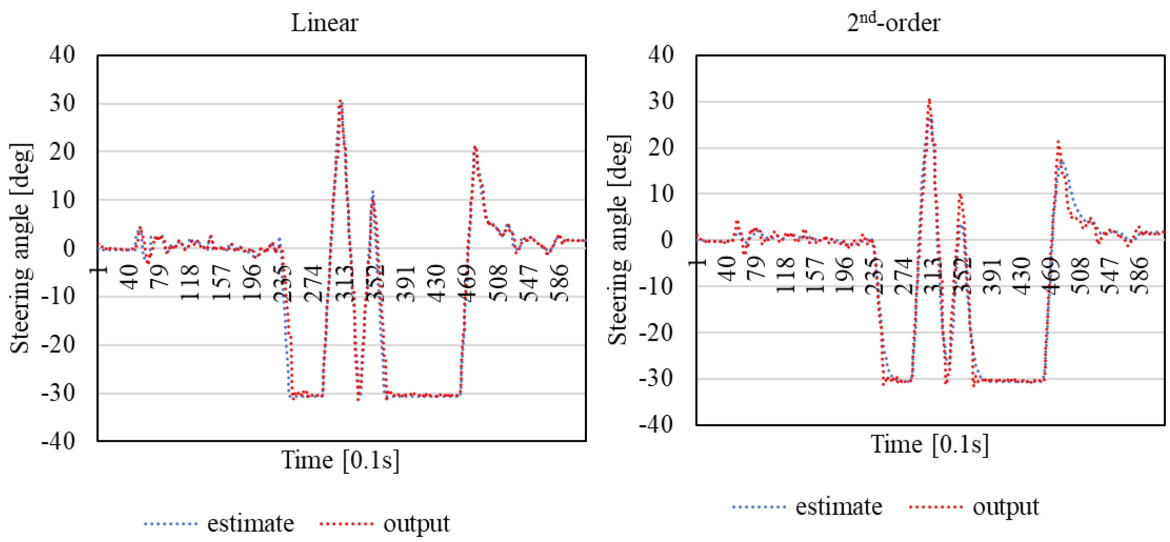


Fig. 51 Estimated output of the linear and 2nd-order

5.3.2. Slip characteristics

In the traveling test regarding slip ratio in Section 5.2.3.2, Fig. 52 shows the vehicle behavior when the maximum steering angle was input. In the figure, A and B were at different locations. In A, the slope was -4.8 degrees in the roll direction and -3.8 degrees in the pitch direction, and in B, the slope was 7.2 degrees in the roll direction and -6.9 degrees in the pitch direction. Note that the roll angle is positive when pointing to the right, and the pitch angle is positive when pointing upward. In both A and B, the vehicle turned left. Although the direction of inclination was different in both sites, the slip ratio was $\gamma = 0.961$ in A, $\gamma = 0.965$ in B, showing almost no difference, indicating understeer characteristics. The mean absolute errors were 0.036 m in A and 0.020 m in B. Although the longer the estimation section is, the larger the error is due to accumulation errors, the slip ratio γ was effective to express slippage during turns. The slip ratio γ can express the behavior at the maximum steering angle with slopes of 10 degrees or less, such as in the tested orchard.

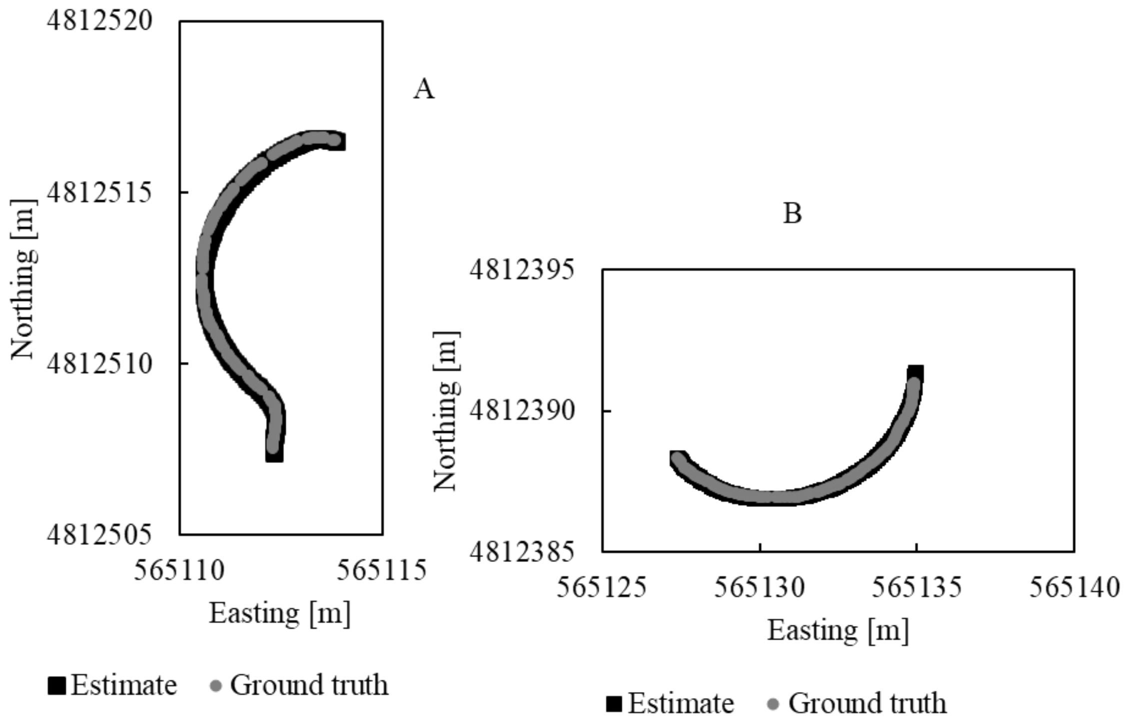


Fig. 52 Traveled and estimated trajectories with a constant steering angle

Table 19 Results of traveling tests

	A	B
Roll max [deg]	-1.7	7.2
Roll min [deg]	-4.8	-1.4
Pitch max [deg]	2.5	3.6
Pitch min [deg]	-3.8	-6.9
γ [-]	0.961	0.965
MAE [m]	0.036	0.020

On the other hand, when the steering angle changed significantly, we found that the slip ratio γ did not adequately represent the slippage. Fig. 53 shows the results of estimation with the slip ratio $\gamma = 0.961$, where the steering angle changed significantly from the maximum left to the maximum right. The traveling distance was almost the same as A in Fig. 52. However, as you can see from the diagram, the error became bigger when the first left turn was completed, and the vehicle moved to the right turn. Fig. 54 shows the change in steering angle and the error between the estimated and true trajectories. Although the error remained less than 0.05 m during the first left turn, the error suddenly increased when moving to the next right turn (the section where the steering angle increased from -30 to 30). This shows that the slip angle changed not only according to the position of the steering angle, but also according to the steering angle rate. Therefore, when generating a turning path, the change in the steering angle should be minimized. Moreover, if there is a large amount of steering change, the vehicle behavior will be unstable due to control delay, so the path should have as little steering change as possible.

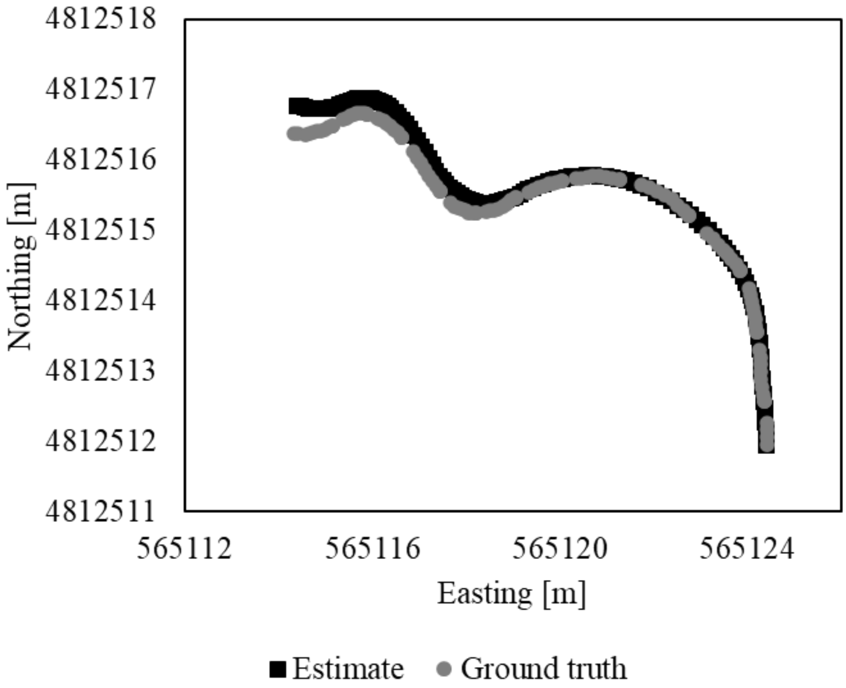


Fig. 53 Traveled and estimated trajectory when the steering angle varied

Fig. 54 The estimation error according to the steering angle

5.3.3. Model accuracy

To verify the accuracy of the kinematic model of the towed implement created in Section 5.2.1.2. Fig. 55 shows the traveled trajectory with the towed implement in the orchard. The ground truth was the actual measured value of the GNSS, and the red dot was the estimated trajectory using Eqs. (42) and (43) from the link angle calculated in Eq. (44). The vehicle speed was the measured value by the GNSS, and the vehicle heading was obtained by fusing the GNSS with the IMU using the least squares method ([63]). The vehicle traveled including the maximum left steering angle and the maximum right steering angle. The error between the estimated and true position of the towed implement was 0.06 m on average and 0.19 m on the maximum, indicating that the behavior of the towed implement could be expressed by the kinematic model. The reason why the maximum error became slightly large was due to the error in the vehicle heading. In this experiment, the GNSS positioning frequency was not maintained regularly, resulting in data dropouts, which resulted in large errors in heading estimation sometimes, especially near the start of a turn. This greatly affected the accuracy of position estimation of the towed implement. The model of the towed implement used this experiment seemed to be highly dependent on the accuracy of the vehicle heading.

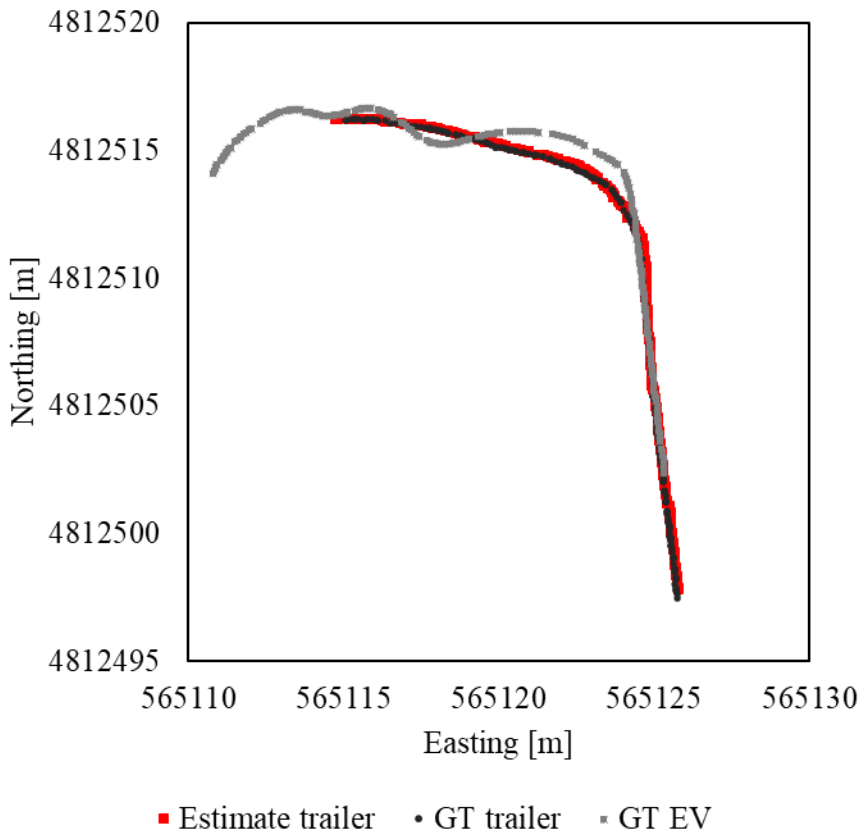
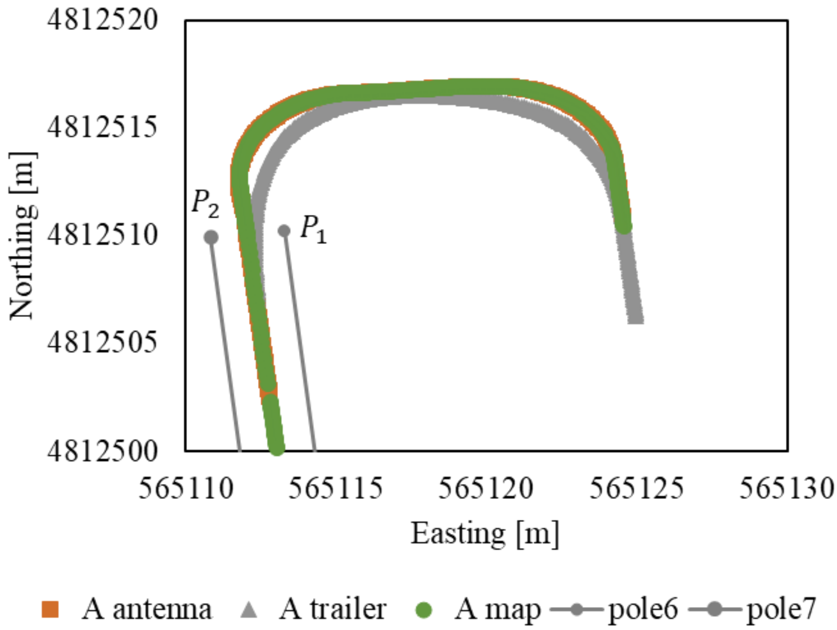


Fig. 55 Trajectories of ground truth (GT) of the EV and the trailer, and estimated trailer

5.3.4. Verification of two turning strategies

Fig. 56 shows the traveling results on two turning paths devised in Section 5.2.1 in simulation according to the controller in Section 5.2.2. As is clear from the figure, since Strategy A only considered the turning radius of the front vehicle, the towed implement passed inside. On the other hand, Strategy B, which was optimized on the vehicle model, considered the difference between the front vehicle and the towed implement. The evaluation values based on Eq. (45) were $J = 2.66$ for Strategy A and $J = 2.20$ Strategy B when the vehicle passed poles P_1 and P_2. In Strategy A, the lateral error of the towed implement was more than twice as large as that in Strategy B, which leads the evaluation value to be large. Additionally, if the distance L_2 from the linked point to the towed implement was made longer than 1.55 m, the towed implement traveled more inner side. When L_2 was 3.5 m, the towed implement contacted with the pole in Strategy A, while with Strategy B, no contact with the pole was observed. The computation time of the GA for Strategy B was approximately 5 minutes. Since real-time path planning was not possible with Strategy B, it was necessary to create turning paths in advance. Furthermore, since the GA calculates a sub-optimal solution, the results were not always applicable. For example, the GA sometimes returned the local solutions, such as entering the next row without colliding anywhere. These inappropriate local solutions must be checked and excluded by humans.



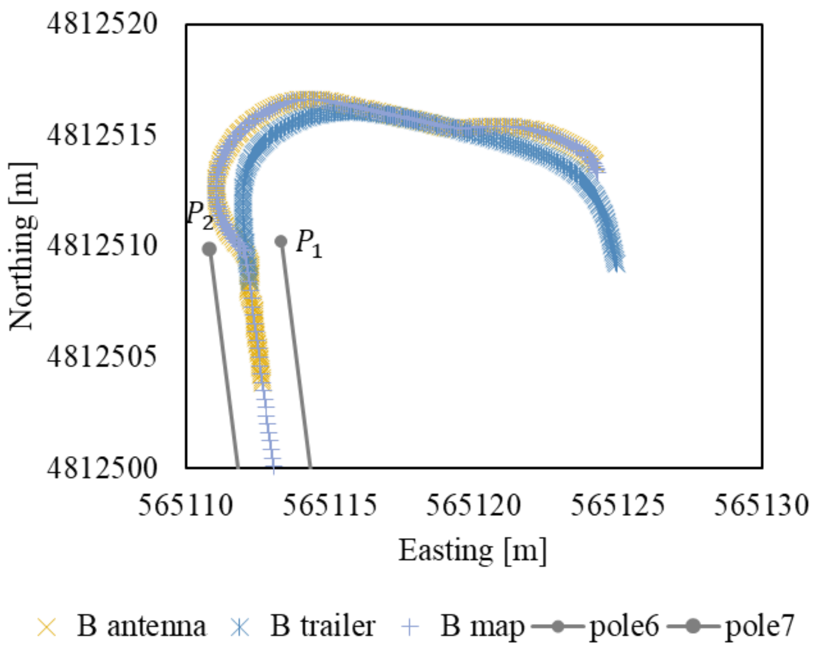


Fig. 56 The trajectories of the front antenna, the trailer, and the navigation map for the strategy A and B

5.4. Conclusion of this chapter

In this study, we considered the turning strategy for a farming vehicle with a towed implement in an orchard. The models of a vehicle and a towed implement were derived and verified those accuracy. A sub-optimal solution was obtained using the GA for the turning path considering the towed implement and the vehicle's posture at the time of the next tree entry. The turning path created by the GA enabled the robot vehicle to safely turn without contacting tree rows and farm boundaries. Furthermore, if the towed implement becomes larger, a turning path that did not consider the towed implement resulted in a collision with a tree because it traveled more inner side than the front vehicle, but a turning path considering the towed implement allowed the robot vehicle to turn safely.

Acknowledgement

In carrying out this research, my supervisor, Professor Noguchi, taught me academic knowledge such as control theory and sensor engineering, as well as important techniques for carrying out research, such as how to write a paper and how to proceed with experiments. Associate Professor Okamoto of the same laboratory and Associate Professor Ishii of the Bioproduction Engineering Laboratory taught me how to create a program and various other technical guidance. Assistant Professor Ospina of the same laboratory supported programming for robot development and instructed English for me. Mr. Kamata, currently at the National Agriculture and Food Research Organization, advised me about research and supported my experiments when I was a master's student. I would like to thank Mr. Morita, a PhD student, Mr. Yamanaka, currently at Kubota, and Mr. Hizatate, a PhD student, who were involved in the development and experiments of the electric vehicle. I am also very grateful to the technical staff, Mr. Sato, Mr. Ichikawa, Mr. Tatebe, Mr. Ojima, and Mr. Wada, who helped manage the farm and served as drivers. I would like to express my deepest gratitude to everyone for their guidance and cooperation.

References

[1] Land survey, Ministry of Agriculture, Forestry and Fisheries. (2023). https://www.maff.go.jp/j/tokei/kouhyou/sakumotu/menseki/index.html.

[2] Household budget survey, Ministry of Internal Affairs and Communications. (2023). https://www.stat.go.jp/data/kakei/longtime/index.html#time.

[3] Export of fruits, Ministry of Agriculture, Forestry and Fisheries. (2021). https://www.maff.go.jp/j/shokusan/export/e_info/attach/pdf/zisseki-17.pdf.

[4] N.T. Agency, Overview of the domestic wine, 2019.

[5] N.T. Agency, Wine industry, 2021.

[6] Overview of Yamanashi, Yamanashi Prefecture Government. (2023). https://www.pref.yamanashi.jp/sabo/114_018.html.

[7] Geographics of Nagano, GEOTECH Co., Ltd. (2023). https://www.jiban.co.jp/tips/kihon/ground/prefecture/nagano.htm.

[8] Hokkaido data book 2021, Hokkaido Government. (2021). https://www.pref.hokkaido.lg.jp/ss/tkk/databook/76967.html.

[9] Forests of Yamagata, Yamagata Prefecture Government. (2023). https://www.pref.yamagata.jp/050011/kurashi/shizen/sinrin/yamagatakennosinrin.html.

[10] N.A. and F.R. Organization, Tree-shaping for labor saving, (2021) 1–110.

[11] K. Ishiyama, S. Honda, H. Imai, Multi-purpose Autonomous Orchard Vehicles, Tree-shaping for Labor Saving, and Vehicle Management IoT System 30 ART for Human Possibilities, 2022.

[12] O.C. Barawid, A. Mizushima, K. Ishii, N. Noguchi, Development of an Autonomous Navigation System using a Two-dimensional Laser Scanner in an Orchard Application, Biosystems Engineering. 96 (2007) 139–149. https://doi.org/10.1016/j.biosystemseng.2006.10.012.

[13] M.H. Jones, J. Bell, D. Dredge, M. Seabright, A. Scarfe, M. Duke, B. MacDonald, Design and testing of a heavy-duty platform for autonomous navigation in kiwifruit orchards, Biosystems Engineering. 187 (2019) 129–146. https://doi.org/10.1016/J.BIOSYSTEMSENG.2019.08.019.

[14] Y. Yanagisawa, R. Matsuura, M. Otani, Y. Ashihara, H. Iwata, S. Doi, S. NAkamura, H. Fukuoka, A. Suda, R. Matsuura, M. Otani, Y. Ashihara, H. Iwata, S. Doi, S. Nakamura, H. Fukuoka, A. Suda, M. Ando, M. Sasaoka, K. Iida, Development of the Transportation Vehicle with a Moving Workbench for Orchard on Sloping Land, The Japanese Journal of the Institute of Industrial Applications Engineers. 9 (2021) 107–117. https://doi.org/10.12792/jjiiae.9.2.107.

[15] A. Yamanaka, Y. Yamasaki, N. Noguchi, Autonomous Navigation System in a Vineyard by Vision Sensors, The Japanese Society of Agricultural Machinery and Food Engineers. 84 (2022) 256–264.

[16] W. Liu, W. Li, H. Feng, J. Xu, S. Yang, Y. Zheng, Overall integrated navigation based on satellite and lidar in the standardized tall spindle apple orchards, Computers and Electronics in

Agriculture. 216 (2024). https://doi.org/10.1016/j.compag.2023.108489.

[17] G. Falco, M. Nicola, M. Pini, G. Marucco, W. De Wilde, A. Popugaev, P. Gay, D.R. Aimonino, Investigation of performance of GNSS-based devices for precise positioning in harsh agriculture environments, 2019 IEEE International Workshop on Metrology for Agriculture and Forestry, MetroAgriFor 2019 - Proceedings. (2019) 1–6. https://doi.org/10.1109/METROAGRIFOR.2019.8909223.

[18] FutureFarming, John Deere: ‘We believe in electric tractors. 100%’ - FutureFarming, (2020). https://www.futurefarming.com/Machinery/Articles/2020/3/John-Deere-We-believe-in-electric-tractors-100-552869E/ (accessed January 30, 2021).

[19] Electrek, John Deere unveils latest all-electric tractor prototype for zero-emission agriculture - Electrek, (2016). https://electrek.co/2016/12/05/john-deere-electric-tractor-prototype/ (accessed January 30, 2021).

[20] T.W. Producer, Are we ready for electric tractors? | The Western Producer, (2019). https://www.producer.com/machinery/are-we-ready-for-electric-tractors/ (accessed January 30, 2021).

[21] M. Gonzalez-de-Soto, L. Emmi, C. Benavides, I. Garcia, P. Gonzalez-de-Santos, Reducing air pollution with hybrid-powered robotic tractors for precision agriculture, Biosystems Engineering. 143 (2016) 79–94. https://doi.org/10.1016/j.biosystemseng.2016.01.008.

[22] J. Caban, J. Vrabel, B. Šarkan, J. Zarajczyk, A. Marczuk, Analysis of the market of electric tractors in agricultural production, MATEC Web of Conferences. 244 (2018) 1–10. https://doi.org/10.1051/matecconf/201824403005.

[23] G.P. Moreda, M.A. Muñoz-García, P. Barreiro, High voltage electrification of tractor and agricultural machinery - A review, Energy Conversion and Management. 115 (2016) 117–131. https://doi.org/10.1016/j.enconman.2016.02.018.

[24] H. Mousazadeh, A. Keyhani, A. Javadi, H. Mobli, K. Abrinia, A. Sharifi, Life-cycle assessment of a Solar Assist Plug-in Hybrid electric Tractor (SAPHT) in comparison with a conventional tractor, Energy Conversion and Management. 52 (2011) 1700–1710. https://doi.org/10.1016/j.enconman.2010.10.033.

[25] Y. Ueka, J. Yamashita, K. Sato, Y. Doi, Study on the development of the electric tractor-Specifications and traveling and tilling performance of a prototype electric tractor, Engineering in Agriculture, Environment and Food. 6 (2013) 160–164. https://doi.org/10.1016/S1881-8366(13)80003-1.

[26] O. Lagnelöv, G. Larsson, A. Larsolle, P.-A. Hansson, Impact of lowered vehicle weight of electric autonomous tractors in a systems perspective, Smart Agricultural Technology. 4 (2023) 100156. https://doi.org/10.1016/J.ATECH.2022.100156.

[27] J. Romero Schmidt, F. Auat Cheein, Assessment of power consumption of electric machinery in agricultural tasks for enhancing the route planning problem, Computers and Electronics in Agriculture. 163 (2019) 104868. https://doi.org/10.1016/J.COMPAG.2019.104868.

[28] L. Cantelli, F. Bonaccorso, D. Longo, C. Donato Melita, G. Schillaci, G. Muscato, A Small Versatile Electrical Robot for Autonomous Spraying in Agriculture, AgriEngineering. 1 (2019) 391–402. https://doi.org/10.3390/agriengineering1030029.

[29] A. Linz, D. Brunner, J. Fehrmann, T. Herlitzius, R. Keicher, A. Ruckelshausen, H.-P. Schwarz, Modelling environment for an electrical driven selective sprayer robot in orchards, Advances in Animal Biosciences. 8 (2017) 848–853. https://doi.org/10.1017/s2040470017000723.

[30] G. Gil, D.E. Casagrande, L.P. Cortés, R. Verschae, Why the low adoption of robotics in the farms? Challenges for the establishment of commercial agricultural robots, Smart Agricultural Technology. 3 (2023) 100069. https://doi.org/10.1016/J.ATECH.2022.100069.

[31] M. Kamiya, Development of Traction Drive Motors for the Toyota Hybrid System, IEEJ Transactions on Industry Applications. 126 (2006) 473–479. https://doi.org/10.1541/IEEJIAS.126.473.

[32] D. Troncon, L. Alberti, Case of study of the electrification of a tractor: Electric motor performance requirements and design, Energies. 13 (2020). https://doi.org/10.3390/en13092197.

[33] E. Sato, Permanent Magnet Synchronous Motor Drives for Hybrid Electric Vehicles, Transactions on Electrical and Electronic Engineering IEEJ Trans. 2 (2007) 162–168. https://doi.org/10.1002/tee.20122.

[34] H. Nakai, H. Ohtani, E. Satoh, Y. Inaguma, Development and Testing of the Torque Control for the Permanent-Magnet Synchronous Motor, IEEE Transactions on Industrial Electronics. 52 (2005) 800–806. https://doi.org/10.1109/TIE.2005.847596.

[35] I. Kobayashi, K. Ogawa, D. Uchino, K. Ikeda, T. Kato, A. Endo, T. Narita, H. Kato, Investigation of Gear Ratios for Hybrid System in Small Race Car Using Lap Time Simulation, IFAC-PapersOnLine. 55 (2022) 442–447. https://doi.org/10.1016/J.IFACOL.2022.10.552.

[36] W.S. Kim, S.Y. Baek, T.J. Kim, Y.S. Kim, S.U. Park, C.H. Choi, Y.J. Kim, Work load analysis for determination of the reduction gear ratio for a 78 kW all wheel drive electric tractor design, Korean Journal of Agricultural Science. 46 (2019) 613–627. https://doi.org/10.7744/kjoas.20190047.

[37] B. Xie, S. Wang, X. Wu, C. Wen, S. Zhang, X. Zhao, Design and hardware-in-the-loop test of a coupled drive system for electric tractor, Biosystems Engineering. 216 (2022) 165–185. https://doi.org/10.1016/J.BIOSYSTEMSENG.2022.02.014.

[38] M. Ichikawa, Dynamics of Wheeled Vehicle, Journal of the Robotics Society of Japan. 13 (1995) 213–218. https://doi.org/10.7210/jrsj.13.213.

[39] T. Kobayashi, E. Katsuyama, H. Sugiura, E. Ono, M. Yamamoto, Study on driving force distribution and power consumption in steady state cornering – Formulation and validation using EV-, FISITA 2014 World Automotive Congress - Proceedings. 45 (2014) 309–314. https://doi.org/https://doi.org/10.11351/jsaeronbun.45.309.

[40] S. Harada, H. Fujimoto, Range Extension Control System for Electric Vehicle during Acceleration and DecelerationBased on Front and Rear Driving/Braking Force

DistributionConsidering Slip Ratio and Motor Loss, IEEJ Transaction on Industry Application. 134 (2013) 6626–6631. https://doi.org/10.1541/ieejias.134.1.

[41] K. Yoshida, Research Issues on Lunar/Planetary Exploration Robots —A Technical Note on Traction Mechanics—, Journal of the Robotics Society of Japan. 27 (2009) 506–509. https://doi.org/https://doi.org/10.7210/jrsj.27.506.

[42] N. Noguchi, Agricultural Vehicle Robot, Journal of Robotics and Mechatronics. 30 (2018) 165–172. https://doi.org/10.20965/jrm.2018.p0165.

[43] C.O. Government of Japan, Quasi-Zenith Satellite System Performance Standard (PS-QZSS-001), 2022. https://qzss.go.jp/en/technical/download/pdf/ps-is-qzss/ps-qzss-003.pdf?t=1658287396868.

[44] Trimble, SPS855 GNSS MODULAR RECEIVER, 2012. https://www.sitech-france.fr/wp-content/uploads/2017/06/Specification-Sheet-SPS855.pdf.

[45] M. Miya, Y. Shima, Start of the operation of the Japanese QZSS Centimeter Level Augmentation Service (CLAS), NAVIGATION. 210 (2019) 6–9. https://doi.org/https://doi.org/10.18949/jinnavi.210.0_6.

[46] H. Wang, N. Noguchi, Navigation of a robot tractor using the centimeter level augmentation information via Quasi-Zenith Satellite System, Engineering in Agriculture, Environment and Food. 12 (2019) 414–419. https://doi.org/10.1016/j.eaef.2019.06.003.

[47] K. Udompant, R. Ospina, Y.-J. Kim, N. Noguchi, Utilization of Quasi-Zenith Satellite System for Navigation of a Robot Combine Harvester, Agronomy. 11 (2021) 483. https://doi.org/10.3390/agronomy11030483.

[48] M. Pini, G. Marucco, G. Falco, M. Nicola, W. De Wilde, Experimental Testbed and Methodology for the Assessment of RTK GNSS Receivers Used in Precision Agriculture, IEEE Access. 8 (2020) 14690–14703. https://doi.org/10.1109/ACCESS.2020.2965741.

[49] M. Kise, N. Noguchi, K. Ishii, H. Terao, Field Mobile Robot navigated by RTK-GPS and FOG (Part 2) - Autonomous operation by applying navigation map -, Journal of the Japanese Society of Agricultural Machinery. 63 (2001) 80–85. https://doi.org/https://doi.org/10.11357/jsam1937.63.5_80.

[50] N. Noguchi, M. Kise, K. Ishii, H. Terao, Field Automation Using Robot Tractor, in: Automation Technology for Off-Road Equipment, ASAE Publication Number 701P0502, Chicago, Illinois, USA, 2002: pp. 239–245.

[51] H. Namie, O. Okamoto, C. Fan, A. Yasuda, Development of a Network-Based RTK-GPS Positioning System using FKP via the Internet, Journal of the Institute of Positioning, Navigation and Timing of Japan. 1 (2010) 9–15. https://doi.org/10.5266/ipntj.1.9.

[52] Y. Yamasaki, N. Noguchi, Optimal route planning and a turning method for an electric vehicle robot in an irregularly shaped vineyard, IFAC-PapersOnLine. 55 (2022) 1–5. https://doi.org/10.1016/J.IFACOL.2022.11.105.

[53] L.C. Santos, F.N. Santos, E.J. Solteiro Pires, A. Valente, P. Costa, S. Magalhães, Path Planning

for ground robots in agriculture: a short review, in: 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta, Delgada, Portugal, 2020: pp. 61–66.

[54] N. Noguchi, H. Terao, Creation of Optimal Route for Agricultural Vehicle and Construction Machinery by Using Genetic Algorithm, Transactions of the Society of Instrument and Control Engineers. 30 (1994) 64–71.

[55] N. Noguchi, H. Terao, Computers and electronics in agriculture Path planning of an agricultural mobile robot by neural network and genetic algorithm, Computers and Electronics in Agriculture. 18 (1997) 187–204.

[56] M. Kise, N. Noguchi, K. Ishii, H. Terao, Field Mobile Robot navigated by RTK-GPS and FOG (Part 3) - Enhancement of Turning Accuracy by Creating Path applied with Motion Constraints -, Journal of the Japanese Society of Agricultural Machinery. 64 (2002) 102–110.

[57] R. Linker, T. Blass, Path-planning algorithm for vehicles operating in orchards, Biosystems Engineering. 101 (2008) 152–160. https://doi.org/10.1016/j.biosystemseng.2008.06.002.

[58] P.E. Hart, N.J. Nilsson, B. Raphael, A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics. 4 (1968) 100–107. https://doi.org/10.1109/TSSC.1968.300136.

[59] R. Torisu, F. Takeda, T. Banno, Equations of the Steering Characteristics of Multi-articulated Vehicles at Low Speeds*, Journal of the Japanese Society of Agricultural Machinery. 52 (1990) 27–34.

[60] M. Karkee, B.L. Steward, Study of the open and closed loop characteristics of a tractor and a single axle towed implement system, Journal of Terramechanics. 47 (2010) 379–393. https://doi.org/10.1016/j.jterra.2010.05.005.

[61] Y. Yamasaki, N. Noguchi, Research on autonomous driving technology for a robot vehicle in mountainous farmland using the Quasi-Zenith Satellite System, Smart Agricultural Technology. 3 (2023) 100141. https://doi.org/10.1016/J.ATECH.2022.100141.

[62] J. Backman, T. Oksanen, A. Visala, Navigation system for agricultural machines: Nonlinear Model Predictive path tracking, Computers and Electronics in Agriculture. 82 (2012) 32–43. https://doi.org/10.1016/j.compag.2011.12.009.

[63] M. Kise, N. Noguchi, K. Ishii, H. Terao, Field Mobile Robot navigated by RTK-GPS and FOG (Part 1) - Estimation of Absolute Heading Angle by Sensor-fusion RTK-GPS and FOG-, Journal of the Japanese Society of Agricultural Machinery. 63 (2001) 74–79.