

## Article

# Row End Detection and Headland Turning Control for an Autonomous Banana-Picking Robot

Peichen Huang <sup>1</sup>, Lixue Zhu <sup>2,\*</sup> , Zhigang Zhang <sup>3</sup> and Chenyu Yang <sup>2</sup>

<sup>1</sup> College of Automation, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China; pchuang@zhku.edu.cn

<sup>2</sup> College of Mechanical and Electrical Engineering, Zhongkai University of Agriculture and Engineering, Guangzhou 510225, China; yangchenyu@zhku.edu.cn

<sup>3</sup> Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, South China Agricultural University, Guangzhou 510642, China; zzg208@scau.edu.cn

\* Correspondence: zhulixue@zhku.edu.cn; Tel.: +86-189-9833-9289

**Abstract:** A row-following system based on machine vision for a picking robot was designed in our previous study. However, the visual perception could not provide reliable information during headland turning according to the test results. A complete navigation system for a picking robot working in an orchard needs to support accurate row following and headland turning. To fill this gap, a headland turning method for an autonomous picking robot was developed in this paper. Three steps were executed during headland turning. First, row end was detected based on machine vision. Second, the deviation was further reduced before turning using the designed fast posture adjustment algorithm based on satellite information. Third, a curve path tracking controller was developed for turning control. During the MATLAB simulation and experimental test, different controllers were developed and compared with the designed method. The results show that the designed turning method enabled the robot to converge to the path more quickly and remain on the path with lower radial errors, which eventually led to reductions in time, space, and deviation during headland turning.

**Keywords:** row end detection; turning control; headland; picking robot



**Citation:** Huang, P.; Zhu, L.; Zhang, Z.; Yang, C. Row End Detection and Headland Turning Control for an Autonomous Banana-Picking Robot. *Machines* **2021**, *9*, 103. <https://doi.org/10.3390/machines9050103>

Academic Editor:  
Antonios Gasteratos

Received: 26 April 2021  
Accepted: 17 May 2021  
Published: 18 May 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Bananas are one of the main economic fruits in South China with large planting fields. However, the process of picking fruits is still principally manually implemented, which leads to issues such as high labor intensity, high costs, and low efficiency [1,2]. To free the farmers from heavy work, reduce repetitive operations, and avoid the harm caused by certain operations, there is a great need for fruit-picking robots [3,4]. A fruit-picking robot is a type of robot that is designed to move through an entire field and pick fruits on its own [5–13]. In our previous study, a row-following system based on machine vision for a banana-picking robot was developed. However, during the headland turning test, it was found that the visual navigation system could not accurately guide the robot to turn and enter the next row due to the unreliable perceptual information provided by the camera near the headland. To fill this gap and achieve a fully automated picking robot, the headland turning method should be studied and developed. Moreover, the optimization of headland turning control can reduce time, space, and fuel consumption during headland turning and, in turn, improve the overall field efficiency of agricultural field equipment [14–17].

A major challenge in improving headland turning navigation is the generation of a feasible turning trajectory based on route planning and motion planning, which numerical studies have mainly focused on [18]. In early studies, the main criteria, such as the minimum turning radius, the maximum lateral acceleration, and steering speed, were

adopted using the Spline function, Bezier curve, etc., to generate the turning paths [19]. Recently, studies have been directed towards computing the turning paths based on numerical optimization. Tu et al. developed kinematic models of the tractor and the tractor-implement systems and formulated headland turning time-optimal problems by incorporating the models and operational constraints. Different types of turning were tested by simulation based on the MATLAB TOMLAB/SNOPT toolbox [20]. Evans et al. developed a mathematical model for computing the route with the minimum in-field non-working time for a single harvester in irregularly shaped fields utilizing a headland field pattern. A genetic algorithm was used for solving the optimization problem. The results showed that the optimized travel path reduced the non-working travel between 13.8 and 31.5% [21]. Vahdanjoo et al. designed a coverage-planning algorithm for capacitated field operations in agricultural fields. By applying the meta-heuristic algorithm, the optimal traversal sequence of fieldwork tracks under the criterion of minimizing the non-working traveled distance during the headland turnings was found. Moreover, a simulation model was designed to generate the traveled distance of the operation [22]. One of the major disadvantages of the numerical optimization method is the high computational time, which is the key factor in real-time applications. The other drawback of this method is the uncertainty of the solution convergence.

Another challenge in headland turning is the motion control, as turning is often considered as curve path tracking. Approaches to path tracking control include backstepping predictive control, fuzzy neural control, sliding mode control, etc. [23]. However, studies of control algorithms for headland turning have not adequately addressed the issue. Wang et al. proposed an adaptive turning algorithm for a four-wheel robot tractor in the headland. Both the slide movement of the robot and the steering rate were used for estimating the turning radius in real-time. Moreover, after each turn, the vehicle model was tuned based on the results to optimize the controller [24]. An intuitive and simple method based on geometry has also been considered and adopted in headland turning control. Huang et al. developed a headland turning controller based on the pure pursuit algorithm for a rice transplanter. Paddy field test results indicated that by determining the appropriate parameters of the model, the transplanter could succeed in turning accurately and repeatedly at the headland [25]. Recently, Yin et al. developed a steering control algorithm for a rice transplanter. A compound fuzzy PID (proportional-integral-derivative) controller was designed to adjust the real-time data of the PID parameters for automatic steering control. The experimental results showed that the maximum lateral and heading deviations of headland turning were 11.5 cm and 5 degrees, respectively [26].

The objective of this paper is mainly focused on the sensor-based detection of row exits and turning control, which have not been addressed adequately in the literature. Moreover, an additional fast posture adjustment algorithm was designed and added to the start and end of the turning procedures in order to decrease the deviation occurring after row following and headland turning. The main contributions are as follows:

1. A complete method for an agricultural robot turning at the headland was developed.
2. The preview distance, which has a significant effect on the performance of fast posture adjustment, was studied and its optimal range was also determined by test.
3. A simulation and experimental tests of the designed method were carried out, and it was compared with different methods in terms of different aspects, such as space, turning time, and deviation.

The remainder of this paper is divided as follows: Section 2 contains the details of the headland turning algorithms. Section 3 details the simulation and experimental test results and provides the discussion. Finally, Section 4 states the conclusions of this study.

## 2. Materials and Methods

### 2.1. System Architecture

A picking robot mainly consists of three parts: a recognition system, a picking system, and a moving system. The moving system is composed of a robot chassis, dual-antenna

GNSS (Global Navigation Satellite System), industrial computer, microcontroller, and speed sensor. RS232 is used to realize data communication between a top-level navigation decision system and a low-level execution system. Real-time images are captured by a camera and processed by an industrial computer to guide the robot in following tree rows and detecting headlands. Once a headland is detected by the vision system, the robot automatically executes a posture adjustment and turns at the headland based on the GNSS information. The banana-picking robot is presented in Figure 1.

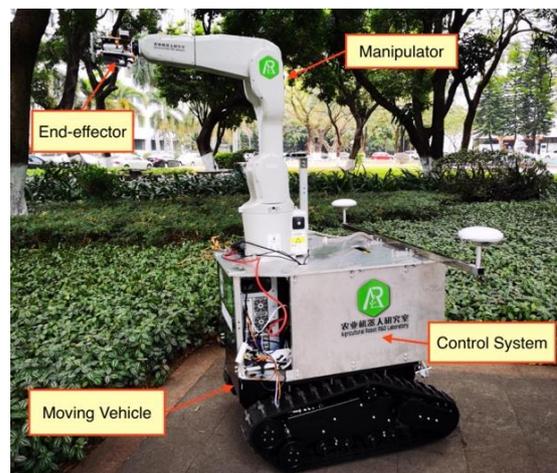


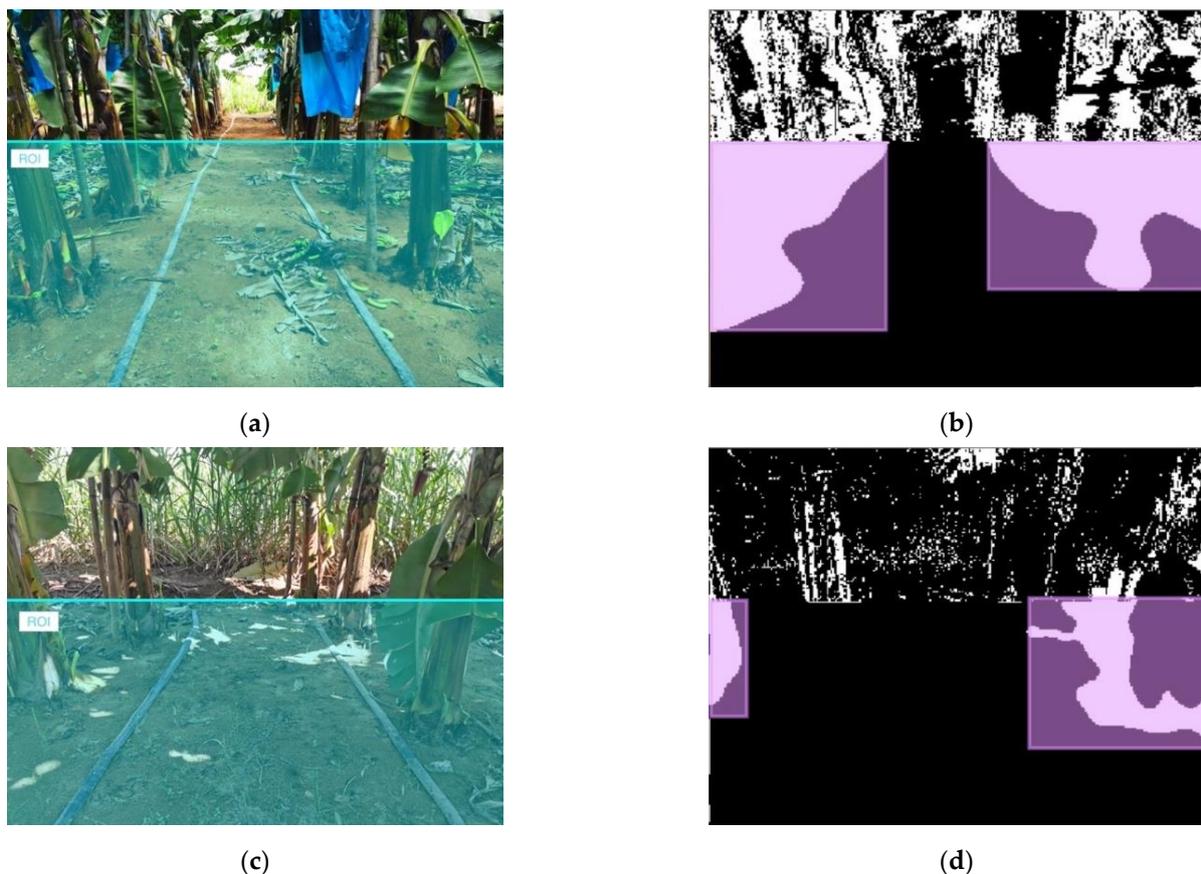
Figure 1. Banana-picking robot.

## 2.2. Headland Detection Based on Vision

The first step is performed using machine vision to detect the approach of the robot to a headland. Figure 2a,b show the original and processed image in the middle of the navigation of the tree rows, while Figure 2c,d show the corresponding images when the robot was close to the end of the tree rows. Image binarization and noise reduction were carried out in advance. The purple rectangular areas represent the tree rows detected in the ROI (Region of Interest). It can be observed that as the robot continued to drive towards the end of the row, the area of the tree row detected in the ROI would gradually decrease. Based on the above analysis, this study took the areas of the tree row detected in the ROI as the headland detection criterion. The areas of the tree row in the binary image were calculated in every frame, and the detection criterion is shown as Equation (1):

$$\frac{1}{2} \sum_{i=0}^{N_l} (x_i y_{i+1} - y_i x_{i+1}) + \frac{1}{2} \sum_{j=0}^{N_r} (x_j y_{j+1} - y_j x_{j+1}) \leq \eta \quad (1)$$

where  $N_l$  and  $N_r$  are the total number of pixels in the left and right tree row regions of the ROI.  $(x_i, y_i)$  and  $(x_j, y_j)$  represent the coordinates of pixel  $i$  and pixel  $j$  in the left and right tree row regions, respectively.  $\eta$  is the detection threshold, which was set according to the camera's field of view and the width and height of the tree rows. When  $\eta$  was set low, the detection range would decrease, and the robot needed to drive close enough to the end of the tree rows to implement switching to the next step of navigation. During our experimental test,  $\eta$  was set to enable the robot to automatically switch to the next step in advance, at about 2 m relative to the last row.



**Figure 2.** Row end detection based on machine vision. (a) Original image in the middle of the row; (b) binary image in the middle of the row; (c) original image at the end of the row; (d) binary image at the end of the row.

### 2.3. Posture Adjustment Based on GNSS

From Section 2.2, it was found that the camera could not provide reliable navigation near the headland. Therefore, there is a great need for another sensing device to accomplish the navigation before starting headland turning. GNSS provides absolute posture measurements and has been widely used for many agricultural tasks [27]. Therefore, posture adjustment based on GNSS was designed in this step. Moreover, to imitate the way in which humans drive, a preview point on the ground was also generated and used for designing the posture adjustment algorithm. According to the preview information, the robot could then be controlled in advance to avoid a large deviation before headland turning. Four modules were included in the posture adjustment algorithm: preview point generation, lateral deviation calculation, a PI (proportional-integral) controller, and a robot kinematic model. As Figure 3 shows, the preview point  $(x_G, y_G)$  with a look-ahead distance  $l$  was first generated based on the current robot posture measured by GNSS. The preview point lateral deviation  $\Delta l$  was then calculated and converted into a robot differential-drive  $\Delta v$  using the PI control algorithm. Finally, the left and right driving signals were calculated based on the robot kinematic model to control the actuator.

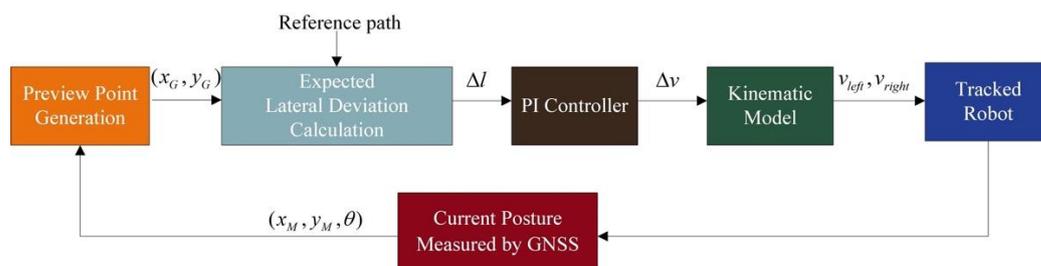


Figure 3. Schematic diagram of robot posture adjustment.

### 2.3.1. Kinematics Model

A tracked mobile robot could be simplified as a wheeled robot with differential drive [28]. As Figure 4 shows, the current robot posture is defined as  $(x, y, \theta)$  in the world frame  $\{O_w, X_w, Y_w\}$ , while the robot vehicle frame is defined as  $\{O_r, X_r, Y_r\}$ . The kinematic model in the robot frame is given by Equation (2):

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ -\frac{R}{L} & \frac{R}{L} \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \tag{2}$$

where  $R$  is the radius of the wheels, and  $L$  is the distance between the left and right wheels.  $v$  is the robot linear velocity along  $X_r$ .  $v_l/v_r$  is the left/right wheel angular velocity, and  $\omega$  is defined as the robot heading rate. The kinematic model in the global frame is given by Equation (3):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{3}$$

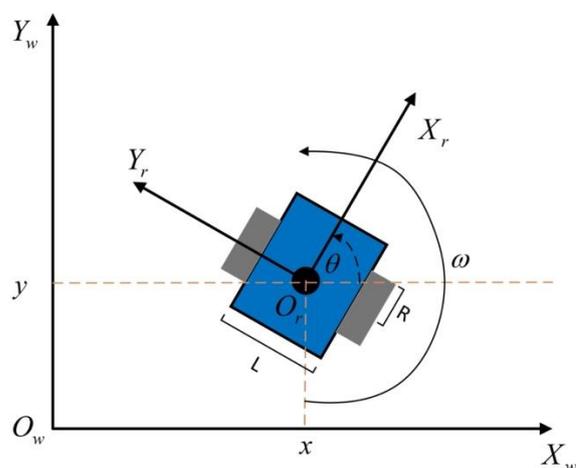


Figure 4. Differential-drive robot kinematics model.

Equation (4) can be derived from Equation (2) as follows:

$$\begin{bmatrix} v_l \\ v_r \end{bmatrix} = \begin{bmatrix} \frac{1}{R} & -\frac{L}{2R} \\ \frac{1}{R} & \frac{L}{2R} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4}$$

### 2.3.2. Preview Point Generation

As Figure 5 shows,  $\{W\}$  represents the world frame. The origin of frame  $\{M\}$  locates at the center of the primary antenna, while the origin of frame  $\{G\}$  locates at the preview point  $P$  on the ground in the robot’s moving direction.  $(x_M, y_M, z_M)$  is the coordinates of

the primary antenna in the world frame measured by GNSS, and  $(x_p, y_p, z_p)$  is defined as the coordinates of the preview point.  $\{G\}$  is different from  $\{M\}$  only by a translation.

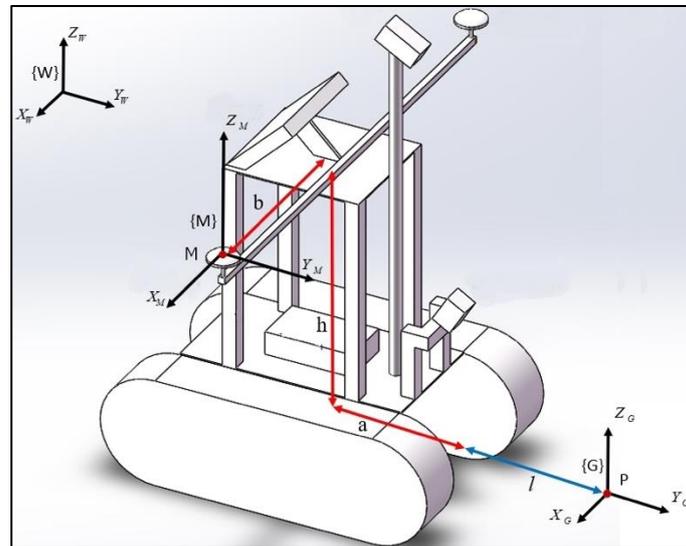


Figure 5. Preview point generation in 3D space.

Based on the theory of spatial descriptions and transformations, the coordinates of  $P$  can be obtained through Equation (5):

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} x_M \\ y_M \\ z_M \end{bmatrix} + {}^W_M R_{XYZ}(\theta_r, \theta_p, \theta_y) \cdot \begin{bmatrix} b \\ a+l \\ h \end{bmatrix} \quad (5)$$

where  $b$ ,  $h$ ,  $a + l$  represents the distance from the preview point to the primary antenna.  $\theta_r$ ,  $\theta_p$  and  $\theta_y$  are the roll, pitch, and yaw angles measured by GNSS, respectively.  ${}^W_M R_{XYZ}(\theta_r, \theta_p, \theta_y)$  is a rotation matrix that describes the orientation of the frame  $\{M\}$  with respect to frame  $\{W\}$ .

### 2.3.3. Posture Adjustment Controller Design

According to the differential drive robot kinematics, by changing the speed of two wheels, the robot is able to spin in place, move in a straight line, or follow a predefined trajectory. In our study, differential drive  $\Delta v(t)$  was designed as the output, and the lateral error of the preview point  $\Delta l$  was designed as the input of the controller. The control law is presented as Equation (6):

$$\Delta v(t) = k_p \Delta l(t) + k_i \int_0^t \Delta l(\tau) d\tau \quad (6)$$

where  $k_p$  and  $k_i$  are the proportional and integral gain. From Equations (2) to (6), the expected linear velocity of the left and right track  $v_{lexp}(t)$ ,  $v_{rexp}(t)$  could be calculated as follows:

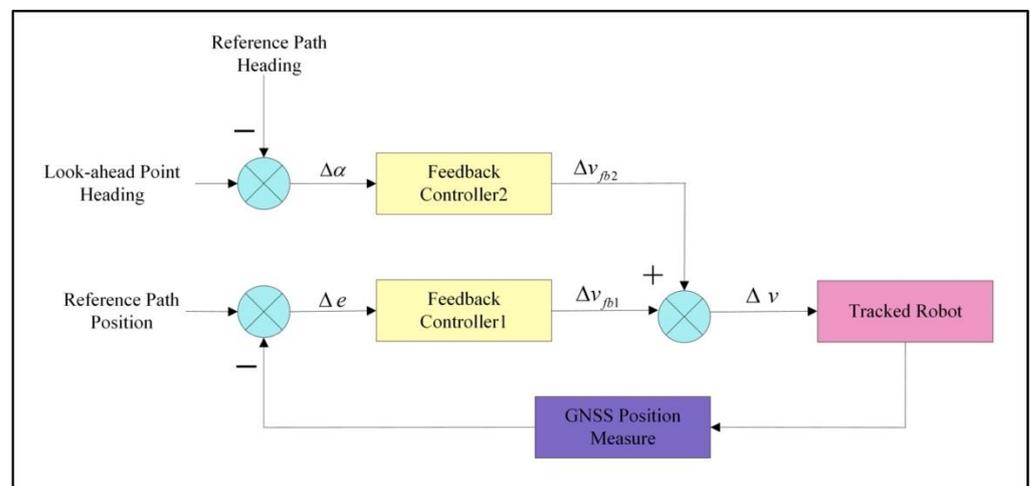
$$\begin{bmatrix} v_{lexp}(t) \\ v_{rexp}(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & -1 & -1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ k_p \Delta l(t) \\ k_i \int_0^t \Delta l(\tau) d\tau \end{bmatrix} \quad (7)$$

where  $v(t)$  is the robot's moving velocity. Once the calculated expected value exceeds the maximum value of the motor velocity  $v_{l_{max}}$ , a new linear velocity is then recalculated using Equation (8):

$$v_{i\_new} = \begin{cases} v_{iexp} - \left( \max(v_{lexp}, v_{rexp}) - v_{i\_max} \right) & \text{if } \max(v_{lexp}, v_{rexp}) > v_{i\_max} \\ v_{iexp} - \left( \min(v_{lexp}, v_{rexp}) + v_{i\_max} \right) & \text{if } \max(v_{lexp}, v_{rexp}) < -v_{i\_max} \\ v_{iexp} & \text{if } \max(v_{lexp}, v_{rexp}) \in [v_{i\_max}, -v_{i\_max}] \end{cases} \quad (i = l \text{ or } r) \quad (8)$$

#### 2.4. Headland Turning Control Method

In closed-loop motion control applications, it is common to use a feedback controller, such as the PID, to generate the control output. During headland turning, the robot often has to track the curve path with a large curvature. To reduce the error more quickly and keep it smaller, rather than relying on a single PID alone, another feedback controller based on preview point heading errors was designed. Figure 6 presents the designed controller structure. The radial deviation  $\Delta e$  of the robot center relative to the reference path was inputted into feedback-loop-1, while the preview point heading deviation  $\Delta\alpha$  was inputted into feedback-loop-2. The outputs of the two loops were then summed as the final steering command.



**Figure 6.** Turning controller structure diagram.

Figure 7 displays the geometry details of the designed headland turning controller. The robot is tracking a curve path on a flat surface in the case of turning left. The curve path tracking algorithm is described as follows:

The current robot posture  $(x_r(t), y_r(t), \theta_r(t))$  in the world coordinate system is located based on GNSS.

Along the reference path, the nearest point A  $(x_a(t), y_a(t))$  relative to the robot center is found and the distance is calculated as the input of feedback-loop-1:

$$\Delta e(t) = \pm \sqrt{(x_a(t) - x_r(t))^2 + (y_a(t) - y_r(t))^2} \quad (9)$$

where the sign of  $\Delta e$  is defined based on which side the robot is on. In our study, for the case of left headland turning,  $\Delta e > 0$ .

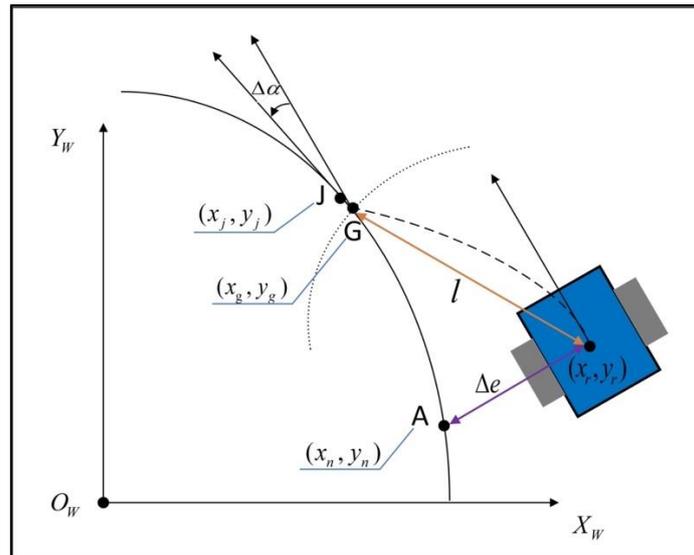


Figure 7. Turning controller geometry.

Starting from point A, the preview point G  $(x_g(t), y_g(t))$  is found using the preview distance  $l$ . The expected heading is then calculated according to point G and the adjacent point J  $(x_j(t), y_j(t))$ . Given the robot current heading  $\theta_r(t)$ , the input of the feedback-loop-2  $\Delta\alpha$  is then calculated by Equation (10):

$$\Delta\alpha(t) = \arctan\left(\frac{y_j(t) - y_g(t)}{x_j(t) - x_g(t)}\right) - \theta_r(t) \quad (10)$$

Based on the kinematics model,  $\Delta v(t)$ , as the output of the designed controller, can then be calculated:

$$\Delta v(t) = \left(k_{fb\_p1}\Delta e(t) + k_{fb\_i1} \int_0^t \Delta e(\tau) d\tau\right) + \left(k_{fb\_p2}\Delta\alpha(t) + k_{fb\_i2} \int_0^t \Delta\alpha(\tau) d\tau\right) \quad (11)$$

where  $k_{fb\_p1}$ ,  $k_{fb\_i1}$ ,  $k_{fb\_p2}$ , and  $k_{fb\_i2}$  are the feedback-loop-1 and feedback-loop-2 gains.

### 3. Results and Discussion

#### 3.1. Posture Adjustment Experiment

During the test, the robot velocity was set to 0.5 m/s, and the update rates of the GNSS system were set to 20 Hz. For the initial tuning of the controller, the look-ahead distance was fixed at 3 m, and the Ziegler Nichols method was adopted, in which only the proportional gain was given and increased until sustained oscillations occurred. The desired gain values could then be calculated and further tuned. In our test, it was found that satisfactory gains were  $k_p = 0.3$ ,  $k_i = 0.03$ . The influence on the tracking performance of changing the look-ahead distance was also studied. A total of five test runs with different look-ahead distances were executed in the same starting position. The posture adjustment performances were recorded, as shown in Figure 8.

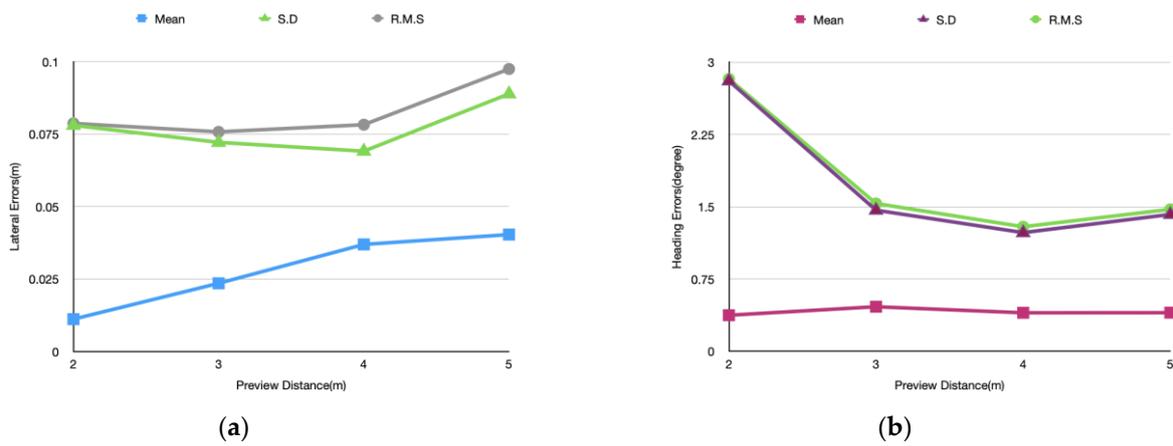


Figure 8. Posture adjustment performance. (a) Lateral errors; (b) heading errors.

During the test, the lateral errors and heading errors were controlled, on average, below 0.03 m and 0.4 degrees, respectively. Figure 8 also shows that there was a reasonable range of the preview distance, which enabled the robot to smoothly regain the path with less oscillation while maintaining low deviation. By trial and error, the reasonable range was found to be 2–3 m. Furthermore, some of the lateral error trajectories are presented in Figure 9, which clearly shows that, with 18–22 cm initial lateral errors, after driving 2–3 m, the posture adjustment algorithm could control the deviation within 5 cm.

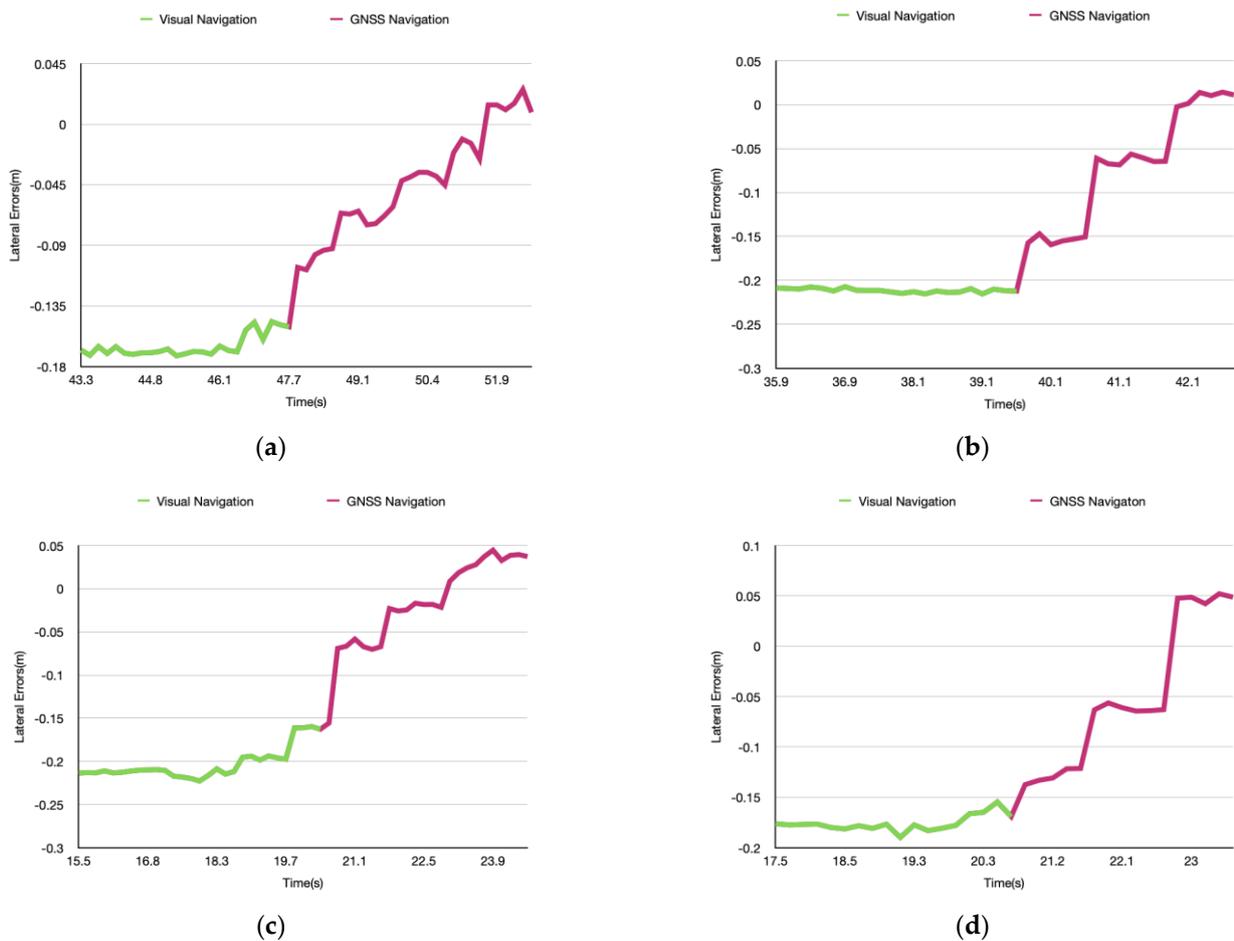
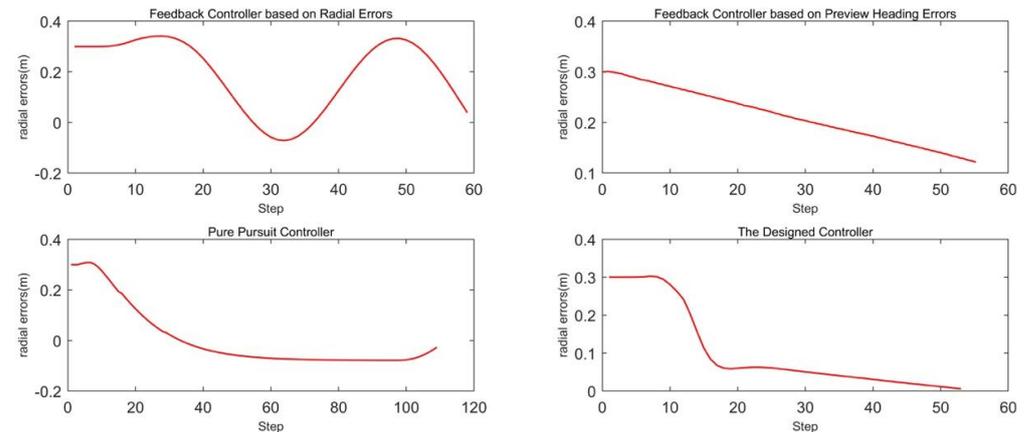


Figure 9. Some lateral error trajectories during the test. (a) Test 1; (b) Test 2; (c) Test 3; (d) Test 4.

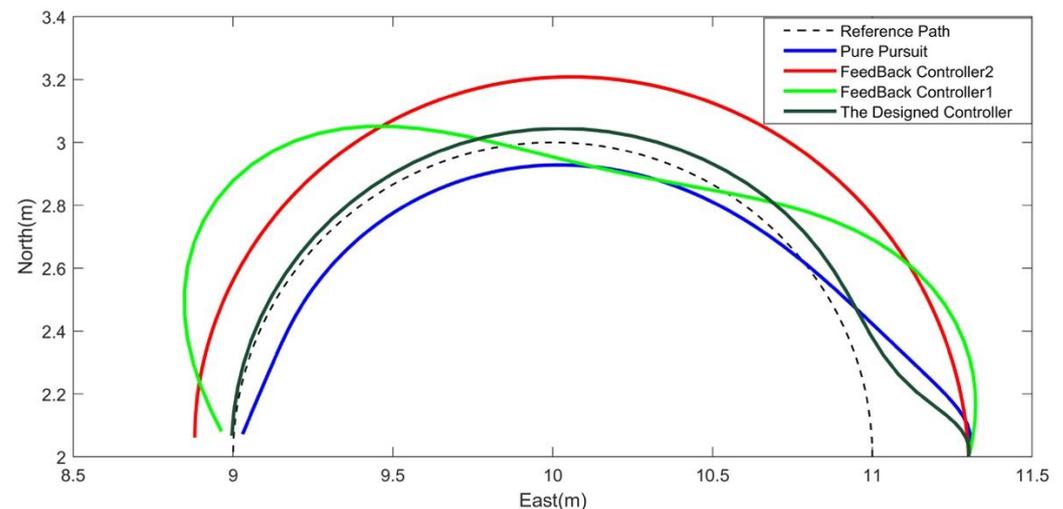
### 3.2. Headland Turning Experiment

#### 3.2.1. Simulation

For comparison, a Pure Pursuit controller, a feedback controller based on robot center radial errors, a feedback controller based on preview point heading errors, and the designed controller presented in this paper were implemented and tested using the same reference path in simulation. A semicircular arc with a radius of 1 m was generated as the reference path. The initial posture deviation was set with 0.3 m lateral errors and 15 degree heading errors. The robot velocity was set to 0.3 m/s. Figures 10 and 11 show the radial errors and robot trajectories of each controller during the turning.



**Figure 10.** Radial error comparison during the simulation.



**Figure 11.** Curve tracking trajectory comparison.

As shown in Figure 11, feedback controller 1 was designed based on the radial errors, while feedback controller 2 was designed based on the preview point heading errors. It was found that feedback controller 1 had the worst performance. By contrast, the other three controllers that used the preview information and took corrective action in advance achieved better performance. However, it should also be noted that the feedback controller 2 relied only on the preview heading information and could not effectively eliminate the radial deviation during the whole headland turning, resulting in a 0.12 m deviation at the end of the headland turning process. In contrast to feedback controller 1 and feedback controller 2, the designed controller had a good capability of reducing both the preview heading deviation and the radial errors at the same time. Pure Pursuit is a popular algorithm based on geometry for robot path tracking, which has been widely used in mobile robot path tracking [29]. Even though the Pure Pursuit controller performed well

during the simulation, it was nonetheless found that the designed controller enabled the robot to converge to the path more quickly and remain on the path with lower radial errors than Pure Pursuit. After finishing the headland turning, the radial deviation was  $-0.027$  m for the Pure Pursuit controller and  $0.006$  m for the designed controller.

### 3.2.2. Experimental Test

The path planning included two straight lines for the posture adjustment and a semicircle arc with a radius of  $1$  m for the headland turning, which was created using the Dubins algorithm. The whole headland turning process consists of three parts. First, after finishing headland detection, the robot switches to the posture-adjusting mode. Second, the robot executes curve path tracking based on the designed controller. During the actual test, to avoid turning more due to momentum, after the yaw angles reached  $165$  degrees, the robot automatically switched to the third part in advance. In this part, the robot executed the posture adjustment based on the next straight line reference path. During the test, the robot moved on the flat land and surface soils were naturally hard and dense. After finishing headland detection, the speed was reduced from  $0.5$  m/s to  $0.2$  m/s.

Figure 12 shows the performance in each part of the headland turning process. During the test, it was found that the robot could detect the headland about  $2$  m in advance before reaching the end of the final tree rows. In the first part, the average lateral error was  $0.053$  m, and the standard deviation was  $0.08$ . In the second part, the radial error was controlled within  $0.45$  m on average, and the maximum error was controlled within  $0.8$  m. In the last part, with an initial lateral error of  $0.47$  m, the lateral error was controlled at  $0.29$  m on average. The turning accuracy, headland space, and turning time of the designed controller were measured and compared with the feedback controller based on the radial errors and feedback controller based on the preview point heading errors. The amount of headland space for turning can be related to the distance of the height and width defined in Figure 13 in the case of turning left. The comparison results are shown in Table 1 and Figure 14. Overall, it was clearly found that the designed controller took less time and required less space during the turn. Three of the robot trajectories during the test are shown in Figure 15. In general, based on the headland turning method presented in this study, the robot could automatically detect the headland  $2$  m away from the last row, execute a posture adjustment based on GNSS, implement headland turning, and enter the next tree row.

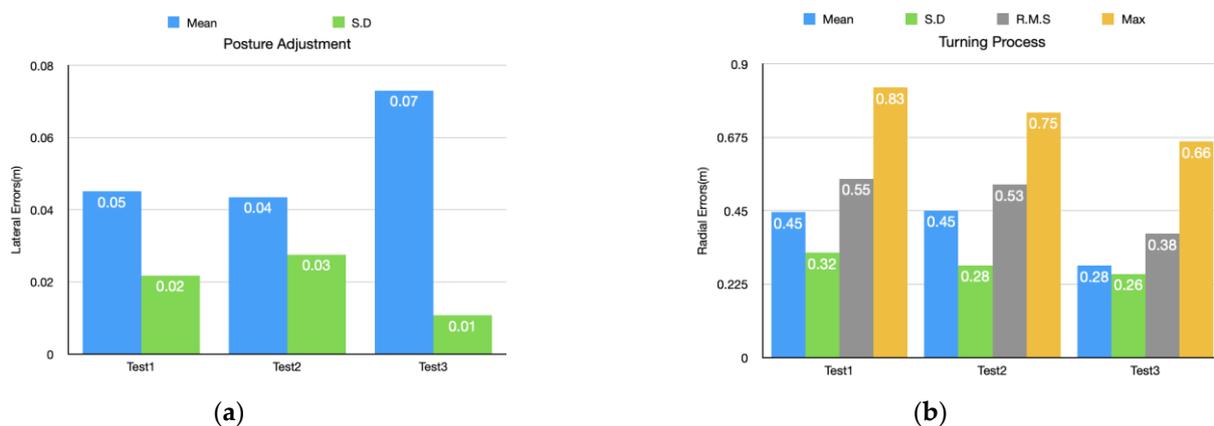
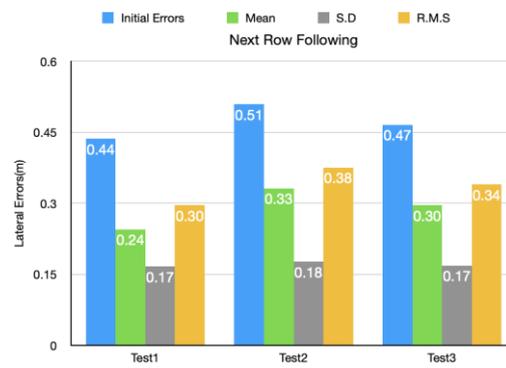


Figure 12. Cont.



(c)

Figure 12. Performance in the headland turning experiment. (a) First part; (b) second part; (c) third part.

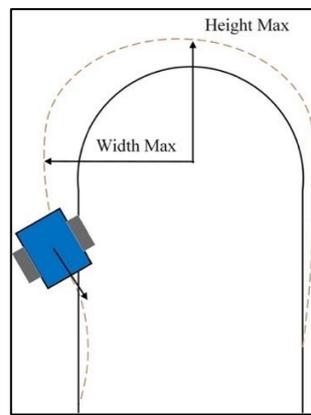
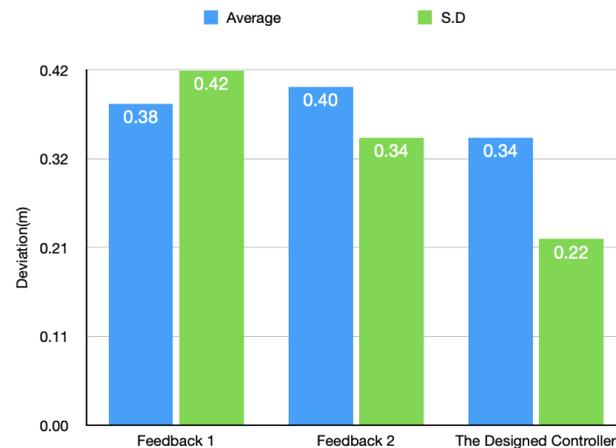


Figure 13. Performance measures relating to the space required for headland turning (in the case of a left turn).

Table 1. Headland turning performance comparison.

	Turning Time (s)	Headland Heigh Max (m)	Headland Width Max (m)
Feedback 1	17.4	3.30	1.15
Feedback 2	18.6	2.37	2.0
Designed	13.0	1.83	1.65

Feedback 1 was based on the radial errors. Feedback 2 was based on the preview heading errors.



Feedback 1 was based on the radial errors.  
Feedback 2 was based on the preview heading errors

Figure 14. Deviation comparison during the experiment test.

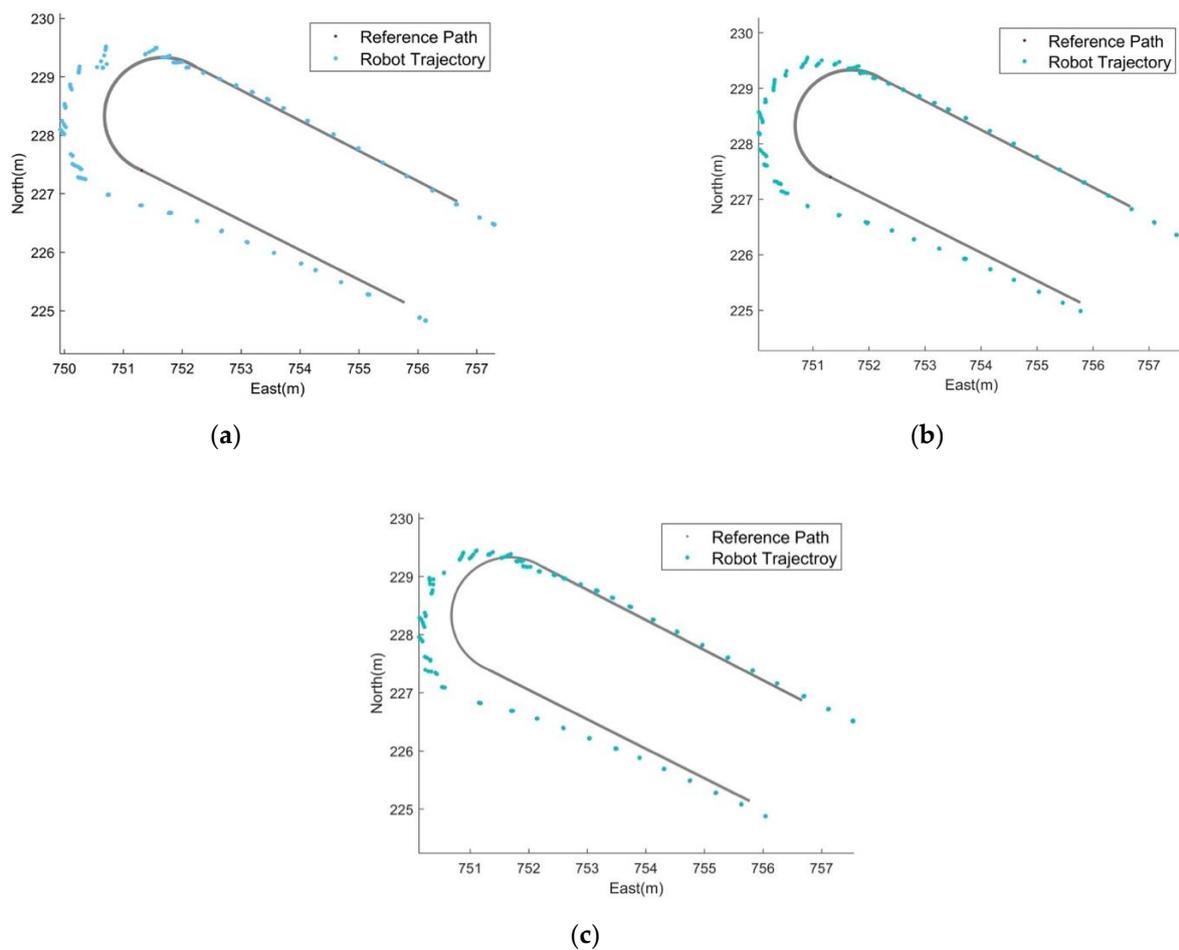


Figure 15. Robot trajectories during the test. (a) Test 1; (b) Test 2; (c) Test 3.

#### 4. Conclusions

Based on our previous research, a complete headland turning method for a banana-picking robot was developed. Since the camera cannot provide effective sensing information when it is close to the headland, a more reliable dual-antenna GNSS system was adopted for the robot posture adjustment and headland turning. Three steps were executed during the headland turning. First, a headland detection method based on machine vision was designed, which enabled the robot to automatically detect the headland about 2 m away from the last row. Second, to reduce the deviation caused by the vision navigation system before headland turning, a posture adjustment was executed based on the precise preview information provided by GNSS. The experimental results show that, with a reasonable preview distance, the lateral deviation could be controlled within 0.03 m, and the heading deviation could be controlled within 0.4 degrees during this step. Third, a turning controller was developed for curve path tracking, which was designed based on the preview point heading errors and robot center radial errors. For the turning control comparison, a Pure Pursuit controller, a feedback controller based on radial errors, a feedback controller based on preview point heading errors, and the designed controller were implemented and tested. The simulation results show that the designed controller achieved the best performance among the controllers, with a radial deviation of 0.006 m after finishing headland turning. An experimental test was also conducted, and the results show that the robot could automatically detect a headland about 2 m away from the last row, execute a posture adjustment, implement headland turning, and enter the next tree row.

In the future, a comparison with some other experimental studies should be done and the following further research should also be conducted. The headland detection algorithm

based on machine vision needs to be improved. Based on our test, it was found that once the light intensity fluctuated or deep shadows occurred, the visual detection algorithm needed to be readjusted to pick out important features consistently. Deep learning, which has the potential to enable robots to learn to perform many complex visual perception tasks, is worth studying and applying in this step. To enable robots to turn more precisely, common path planning for headland patterns, such as the double round corner, the loop, or switch-back turns, should be considered and tested. During the test, it was also found that the preview distance and robot speed had a significant effect on the performance of headland turning. Establishing a precise model that describes the relationship between the optimal preview distance and speed should be researched. Furthermore, it was found that there existed sliding effects when the robot moved on the land during the headland turning. To reduce the effects, specific methods are applied: A tracked mobile robot has been adopted since it has better maneuverability in rough terrain and higher friction in turns due to its tracks and multiple points of contact with the surface. After finishing headland detection, the speed of the robot was reduced from 0.5 m/s to 0.2 m/s; rather than relying on a single PID alone, another feedback controller based on preview point heading errors was designed. In future work, the sliding effects can be further incorporated into an extended kinematics model to enhance the headland turning accuracy.

**Author Contributions:** Conceptualization, P.H. and Z.Z.; methodology, P.H. and L.Z.; software, P.H. and C.Y.; validation, P.H., L.Z. and Z.Z.; formal analysis, C.Y.; investigation, L.Z.; resources, Z.Z.; data curation, C.Y.; writing—original draft preparation, P.H.; writing—review and editing, L.Z.; visualization, P.H.; supervision, L.Z. and Z.Z.; project administration, L.Z.; funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the Science and Technology R&D Projects in Key Fields of Guangdong Province [2019B020223003], Guangdong Agricultural Technology Research and Development Project [2018LM2167], and Guangdong Province Modern Agricultural Industrial Technology System Innovation Team Project [Guangdong Agricultural Letter [2019] [No. 1019]].

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the privacy policy of the organization.

**Acknowledgments:** The authors are grateful to the students, Zhao Yunxia and Lin Guchen, for their help with the experiments.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Zhang, M.; Ji, Y.; Li, S.; Cao, R.; Xu, H.; Zhang, Z. Research Progress of Agricultural Machinery Navigation Technology. *Trans. CSAM* **2020**, *51*, 1–18. [[CrossRef](#)]
2. Zheng, Y.; Chen, B.; Lyu, H.; Kang, F.; Jiang, S. Research progress of orchard plant protection mechanization technology and equipment in China. *Trans. CSAE* **2020**, *36*, 110–124. [[CrossRef](#)]
3. Zhang, W.; Gong, L.; Chen, S.; Wang, W.; Miao, Z.; Liu, C. Autonomous Identification and Positioning of Trucks during Collaborative Forage Harvesting. *Sensors* **2021**, *21*, 1166. [[CrossRef](#)] [[PubMed](#)]
4. Kang, H.; Zhou, H.; Wang, X.; Chen, C. Real-Time Fruit Recognition and Grasping Estimation for Robotic Apple Harvesting. *Sensors* **2020**, *20*, 5670. [[CrossRef](#)]
5. Tang, Y.; Chen, M.; Wang, C.; Luo, L.; Li, J.; Lian, G.; Zou, X. Recognition and localization methods for vision-based fruit picking robots: A review. *Front. Plant Sci.* **2020**, *11*, 510. [[CrossRef](#)]
6. Chen, M.; Tang, Y.; Zou, X.; Huang, K.; Huang, Z.; Zhou, H.; Wang, C.; Lian, G. Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology. *Comput. Electron. Agric.* **2020**, *174*, 105508. [[CrossRef](#)]
7. Bonadies, S.; Gadsden, S.A. An overview of autonomous crop row navigation strategies for unmanned ground vehicles. *Eng. Agric. Environ. Food* **2019**, *12*, 24–31. [[CrossRef](#)]
8. Lin, G.; Tang, Y.; Zou, X.; Wang, C. Three-dimensional reconstruction of guava fruits and branches using instance segmentation and geometry analysis. *Comput. Electron. Agric.* **2021**, *184*, 106107. [[CrossRef](#)]

9. Wang, H.; Dong, L.; Zhou, H.; Luo, L.; Lin, G.; Wu, J.; Tang, Y. YOLOv3-Litchi Detection Method of Densely Distributed Litchi in Large Vision Scenes. *Math. Probl. Eng.* **2021**, *2021*, 1–11. [[CrossRef](#)]
10. Li, J.; Tang, Y.; Zou, X.; Lin, G.; Wang, H. Detection of Fruit-bearing Branches and Localization of Litchi Clusters for Vision-based Harvesting Robots. *IEEE Access* **2020**, *8*, 117746–117758. [[CrossRef](#)]
11. Lin, G.; Tang, Y.; Zou, X.; Cheng, J.; Xiong, J. Fruit detection in natural environment using partial shape matching and probabilistic Hough transform. *Precis. Agric.* **2020**, *21*, 160–177. [[CrossRef](#)]
12. Chen, M.; Tang, Y.; Zou, X.; Huang, K.; Li, L.; He, Y. High-accuracy multi-camera reconstruction enhanced by adaptive point cloud correction algorithm. *Opt. Lasers Eng.* **2019**, *122*, 170–183. [[CrossRef](#)]
13. Wang, C.; Tang, Y.; Zou, X.; Luo, L.; Chen, X. Recognition and matching of clustered mature litchi fruits using binocular charge-coupled device (CCD) color cameras. *Sensors* **2017**, *17*, 2564. [[CrossRef](#)]
14. Guevara, L.; Michałek, M.M.; Cheein, F.A. Headland turning algorithmization for autonomous N-trailer vehicles in agricultural scenarios. *Comput. Electron. Agric.* **2020**, *175*, 105541. [[CrossRef](#)]
15. Bulgakov, V.; Pascuzzi, S.; Beloev, H.; Ivanovs, S. Theoretical Investigations of the Headland turning Agility of a Trailed Asymmetric Implement-and-Tractor Aggregate. *Agriculture* **2019**, *9*, 224. [[CrossRef](#)]
16. Roshanianfard, A.; Noguchi, N.; Okamoto, H.; Ishii, K. A review of autonomous agricultural vehicles (The experience of Hokkaido University). *J. Terramech.* **2020**, *91*, 155–183. [[CrossRef](#)]
17. Chen, J.; Qiang, H.; Wu, J.; Xu, G.; Wang, Z. Navigation path extraction for greenhouse cucumber-picking robots using the prediction-point Hough transform. *Comput. Electron. Agric.* **2021**, *180*, 105911. [[CrossRef](#)]
18. Vougioukas, S.G. Agricultural Robotics. *Annu. Rev. Control Robot. Auton. Syst.* **2019**, *2*, 365–392. [[CrossRef](#)]
19. Fountas, S.; Mylonas, N.; Malounas, I.; Rodias, E.; Santos, C.H.; Pekkeriet, E. Agricultural Robotics for Field Operations. *Sensors* **2020**, *20*, 2672. [[CrossRef](#)]
20. Tu, X.; Tang, L. Headland turning optimisation for agricultural vehicles and those with towed implements. *J. Agric. Food Res.* **2019**, *1*, 100009. [[CrossRef](#)]
21. Evans, J.T.; Pitla, S.K.; Luck, J.D.; Kocher, M. Row crop grain harvester path optimization in headland patterns. *Comput. Electron. Agric.* **2020**, *171*, 105295. [[CrossRef](#)]
22. Vahdanjoo, M.; Madsen, C.T.; Sørensen, C.G. Novel Route Planning System for Machinery Selection. Case: Slurry Application. *Agric. Eng.* **2020**, *2*, 408–429. [[CrossRef](#)]
23. Rokonzaman, M.; Mohajer, N.; Nahavandi, S.; Mohamed, S. Review and performance evaluation of path tracking controllers of autonomous vehicles. *IET Intell. Transp. Syst.* **2021**, *15*, 646–670. [[CrossRef](#)]
24. Wang, H.; Noguchi, N. Adaptive turning control for an agricultural robot tractor. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 113–119. [[CrossRef](#)]
25. Huang, P.; Luo, X.; Zhang, Z. Field verification of the autonomous agricultural machine headland turning control method. *Int. Agric. Eng. J.* **2016**, *25*, 98–105.
26. Yin, J.; Zhu, D.; Liao, J.; Zhu, G.; Wang, Y.; Zhang, S. Automatic Steering Control Algorithm Based on Compound Fuzzy PID for Rice Transplanter. *Appl. Sci.* **2019**, *9*, 2666. [[CrossRef](#)]
27. Pandey, P.C.; Tripathi, A.K.; Sharma, J.K. Chapter 16—An evaluation of GPS opportunity in market for precision agriculture. In *GPS and GNSS Technology in Geosciences*; Petropoulos, G.P., Srivastava, P.K., Eds.; Elsevier: Amsterdam, The Netherlands, 2021; pp. 337–349. [[CrossRef](#)]
28. Bräunl, T. *Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems*; Springer Science & Business Media: Berlin, Germany, 2008; pp. 4–5.
29. Samuel, M.; Hussein, M.; Mohamad, M.B. A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle. *Int. J. Comput. Appl.* **2016**, *135*, 35–38. [[CrossRef](#)]