International Journal of Advanced Robotic Systems
January-February 2020: 1–8
© The Author(s) 2020
DOI: 10.1177/1729881419897678
journals.sagepub.com/home/arx
SAGE

Feedforward-plus-proportional–integral–derivative controller for agricultural robot turning in headland

Peichen Huang, Zhigang Zhang and Xiwen Luo

Abstract

Traveling along straight lines and headland turning are two common motions during the automatic guidance of agricultural machines. However, most studies focus on accurately following parallel tracks in the field and seldom consider the maneuvers at the end of each row. Moreover, numerous studies have mainly focused on planning the global trajectories in the entire field to increase field efficiency, and very few works are related to maneuver generation and vehicle control in headlands. In this study, a feedforward-plus-proportional–integral–derivative controller for a differential drive robot in headland turning was developed. The feedforward-plus-proportional–integral–derivative controller consisted of a feedforward and a feedback loop. The feedforward loop was designed based on the heading errors of a lookahead point on the planning path. The feedback loop was designed based on radial errors to improve the tracking accuracy. Field comparison tests of feedforward, feedback, and feedforward-plus-proportional–integral–derivative controllers were conducted. Experimental results showed that the feedforward-plus-proportional–integral–derivative had better tracking results and took less turning time.

Keywords

Feedforward-plus-proportional–integral–derivative control, headland turning, differential drive robot, dual-antenna GNSS, path tracking

Date received: 21 August 2018; accepted: 5 December 2019

Topic: Mobile Robots and Multi-Robot Systems
Topic Editor: Nak-Young Chong
Associate Editor: Euntai Kim

Introduction

Mobile robots play a significant role in many agricultural applications as they reduce human labor and enhance the operational safety. The need for autonomous navigation systems of mobile robots has been recognized in different agricultural tasks such as planting, spraying, fertilizing, cultivating, harvesting, thinning, weeding, and inspection.[1,2] Traveling in straight lines and headland turning are two common motions during the automatic guidance of an agricultural machine. Completely autonomous guidance would require the robot's capability to make turns in the headland and enter the next operation row. Several factors such as space availability, vehicle limitations, obstructions, and time taken to turn have to be taken into account when controlling robot during headland turning. However, most studies focus on accurately following parallel tracks in the field, and seldom consider

Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, South China Agricultural University, Guangzhou, China

Corresponding author:
Zhigang Zhang, Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, South China Agricultural University, Guangzhou 510642, China.
Email: zzg208@scau.edu.cn

the maneuvers at the end of each row.[3–6] Miller et al. have reported that the space required to turn at the headland and the time spent in turning significantly affect the field efficiency. They also report that the damage of crops owing to improper headland turns affects productivity.[7] To benefit from fully automated solutions and, therefore, reduce the operator's workload, the problem of maneuver automation in headlands needs to be studied with meticulous care.

Kise et al. tested headland turns using a spline function path planning method on a tractor using a global positioning system-based guidance system. The method guided the vehicle along the desired path with an error of less than 20 cm in lateral deviation.[8] Hague et al. used vision-based guidance in combination with dead reckoning to guide a vehicle through crop rows. The guidance system detected the end of crop rows using the absence of crop-row information from the vision and distance information in dead reckoning. The vehicle was able to execute a U-turn at the headland.[9] Subramanian et al. developed a turning maneuver for an autonomous vehicle from a row into the adjacent row at the headlands of citrus groves. The results showed that the vehicle could turn into the headland at the end of a row, navigated the headland, and turned into the adjacent row without any contact with the alleyway trees.[10] Cariou et al. addresses the problem of path generation and motion control for the autonomous maneuver in headland of a farm vehicle with a trailed implement. A reverse turn planner is firstly investigated, based on primitives connected together to easily generate the reference motion. Then, both steering and speed control algorithms are presented to accurately guide the vehicle–trailer system.[11] In the literature, numerous studies have mainly focused on planning the global trajectories in the whole field to increase field efficiency,[12–19] and very few works are related to maneuver generation and vehicle control in headlands. Most of the previous research focuses on wheel-type robots owing to their applicability to a wide range of challenging situations by developing control techniques. However, crawler-type tractors are also used in some agricultural fields because of their high traction and low ground pressure.[20] Moreover, many of the previous methods either use open-loop control, which is quite sensitive to measurement errors and disturbances, or are highly model dependent, making the controllers very complex and hard to implement in practice.

Proportional–integral–derivative (PID) controllers are the most employed controllers in industry and robot field owing to their relative simplicity and satisfactory performance for a wide range of processes they are capable to provide.[21] However, for robot's headland turning, most of the path planning maneuvers consist of curve paths.[16] If the robot tracks such a path, it must perform direction changes very often, which requires the controller to correct tracking error in rapid respond. PID controllers take corrective action only after the errors were generated. This limitation will lead to large errors or even instabilities during the robot headland turning. One of the most significant improvements both for the set point following[22] and for the disturbance rejection task[23,24] is to incorporate feedforward control with knowledge about the system and using the PID to control error. Some feedforward plus feedback control schemes had been designed for path trackings, such as path curvature and longitudinal force inputs,[25] driver steering command,[26,27] explicit functions of the reference path,[28] and a certain "lookahead" point in front of the vehicle[29–31] were used to design the feedforward controllers, whereas dynamic feedback linearization,[32] fuzzy logic,[33] and adaptive PID[34] were used to design feedback controllers. Same with the previous work,[29,30] a lookahead point was chosen to design the feedforward controller in this article. Specially, the orientation error of the lookahead point was used to make the controller more suitable for robot headland turning. Meanwhile, considering a simple but effective approach to implement the feedback controller. A proportional–integral algorithm was used to make it simpler and steadier in the steady-state in the case of noisy data compared with the traditional PID.

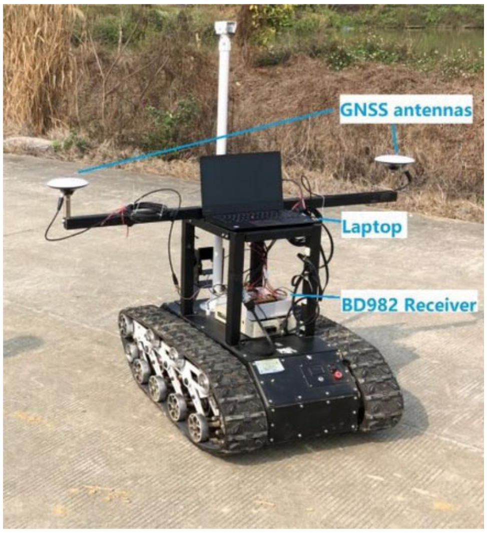

Figure 1. Research platform (Komodo-01).

The main contribution of this article was applied to the feedforward-plus-proportional–integral–derivative (FPID) controller for solving the agricultural robot headland turning problem. The FPID controller designed in this study is model-independent and easy to implement while still improves path tracking accuracy. Field comparison tests were conducted to examine the controllers' performance regarding headland tracking accuracy, turning time, and next row lateral errors.

Materials and methods

Research platform and navigation hardware

A crawler-type robot platform was used in this research. Figure 1 shows the research platform, a Komodo-01. Table 1 presents its specifications. Global Navigation

Table 1. Specifications of the tracked robot.

Dimensions	Power rating	Gearbox ratio	Track width	Max speed	Wheel diameter	Climbing capacity
650 × 350 × 240 mm^3	2 × 650 W	15:1	150 mm	6 km/h	158 mm	40°

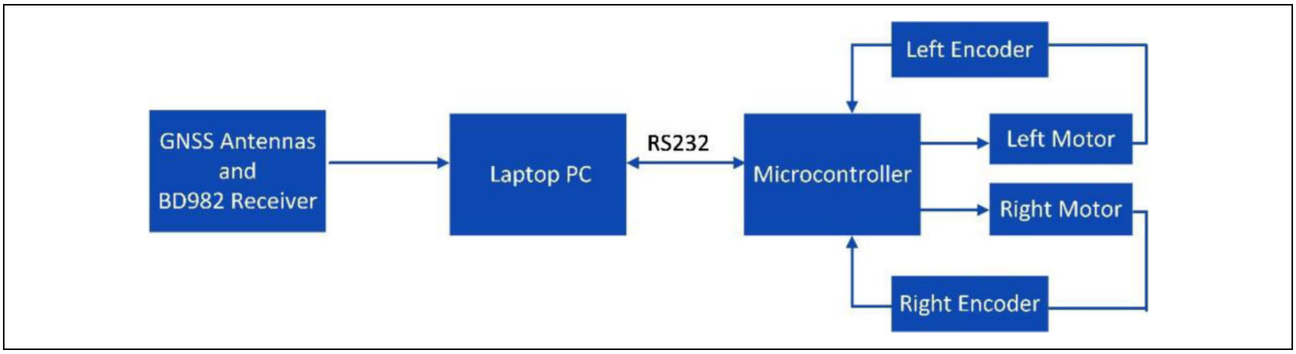

Figure 2. Schematic diagram of the robot platform.

Satellite System (GNSS) antennas and a Trimble BD982 receiver were used as navigation sensors to determine position errors to evaluate the autonomous navigations. To obtain the absolute position of the robot platform, a base station was used with a positioning accuracy of ± 2 cm. A laptop PC (Lenovo, T460s) was used for performing high-level algorithm processing and managing the overall guidance system. Qt Creator 5.7 was used for implementing the algorithms for navigation and graphical user interface for analyzing the algorithms' behavior in real time. A microcontroller was used for low-level control operations such as converting PC's digital command signal to analog signal to operate the left and right motor and processing the encoder data. The PC and microcontroller were communicated through RS232. Figure 2 shows a schematic diagram of the robot platform.

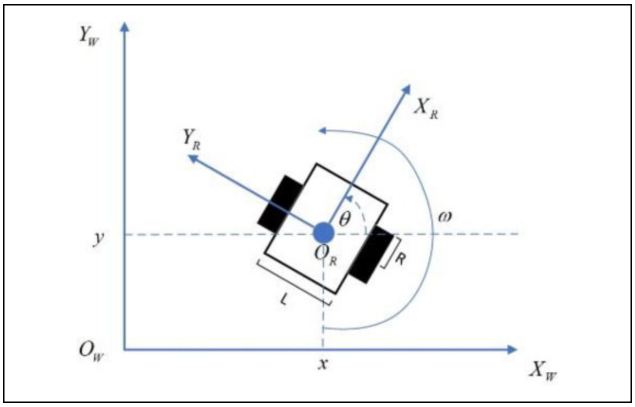

Figure 3. Kinematics model of differential drive robot.

Kinematics of the differential drive mobile robot

Kinematics is the study of motion without considering the forces. A tracked mobile robot can be seen as a special case of a wheeled robot with the differential drive. The difference is the robot's better maneuverability in rough terrain and its higher friction in turns, owing to its tracks and multiple points of contact with the surface.[20]

As Figure 3 shows, the configuration of the robot platform is represented by the generalized coordinate constraints $q = (x, y, \theta)$, where (x, y) is the position and θ is the heading of the center of the axis of the wheels $O_R$with respect to a global inertial frame $\{O_W, X_W, Y_W\}$. Let $\{O_R, X_R, Y_R\}$ be the robot frame.v_x and v_y are robot's linear velocities in the robot frame's x-direction and y-direction, separately. L is the distance between the wheels, $R$is the radius of the wheels, v_l is the left wheel angular velocity, v_r is the right wheel angular velocity, and ω is the heading rate. The kinematic model in the robot frame is given by

$$\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ -\frac{R}{L} & \frac{R}{L} \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \tag{1}$$

and the kinematic model in the world frame is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ \omega \end{bmatrix} \tag{2}$$

v_l, v_r, and ω are related as (3) and (4)

$$v_l = \frac{2v_x - \omega L}{2R} \tag{3}$$

$$v_r = \frac{2v_x + \omega L}{2R} \tag{4}$$

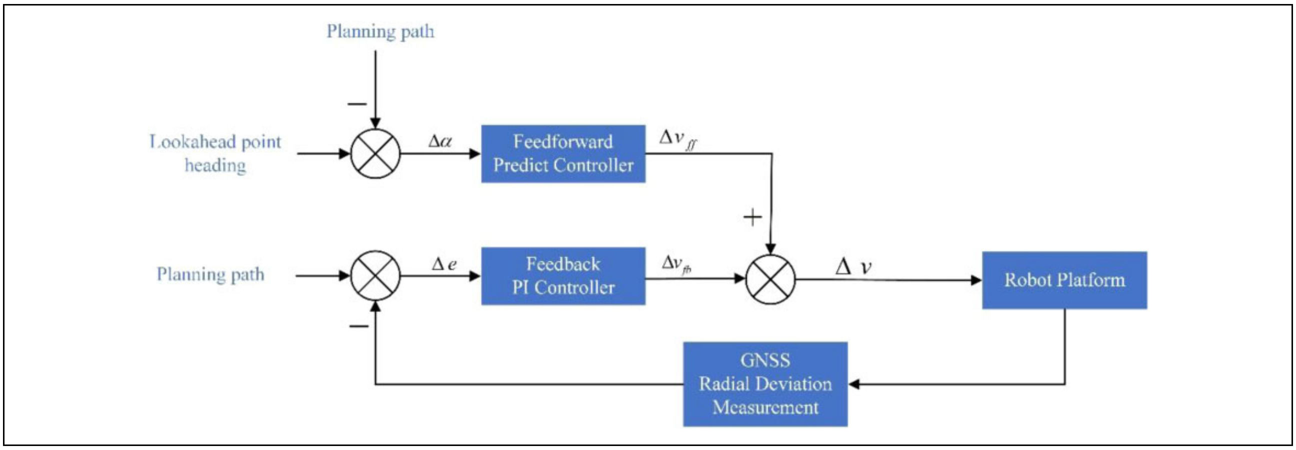

Figure 4. Block diagram of the FPID controller. FPID: feedforward-plus-proportional–integral–derivative.

from the equations above, if $v_l = v_r$, the robot moves straight along X_R; If $v_l = -v_r$, the robot is stationary and spinning.

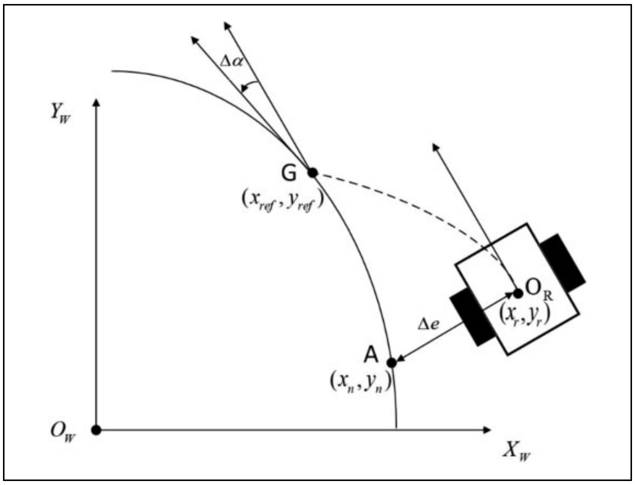

Figure 5. FPID controller geometry. FPID: feedforward-plus-proportional–integral–derivative.

Headland turning controller design

The FPID consists of a feedforward loop and a conventional feedback loop. Input to the feedforward is the heading deviation $\Delta\alpha$ of a lookahead path point. Input to the feedback is the radial path deviation Δe. The total steering command is the sum of the feedback and feedforward outputs. The block diagram of the FPID controller structure is shown in Figure 4.

The FPID controller geometry used in this work is represented in Figure 5. Consider a differential drive robot taking a left turn on a flat surface. The planning path is created as a circular arc and represented as a subset of the points referred to as "navigation points." $(x_r(t), y_r(t), \theta_r(t))$ stands for the robot current position in global coordinates, which is determined by GNSS. Point $A(x_n(t), y_n(t))$ represents the closest path point to the robot center. The FPID controller can be implemented as detailed below:

1. Find the current position of the vehicle in the global coordinate system $(x_c(t), y_c(t), \theta_c(t))$;
2. Find the closest point $A(x_n(t), y_n(t))$ on the planning path to the robot center and calculate the error Δe

$$\Delta e(t) = \pm\sqrt{(x_n(t) - x_r(t))^2 + (y_n(t) - y_r(t))^2} \qquad (5)$$

As illustrated in Figure 5 for the case the robot taking a left headland turning, $\Delta e > 0$, and for the case where the robot is on the left side of the path, $\Delta e < 0$.This error is used for the feedback loop of the FPID controller;

3. Starting from point A, the lookahead point $G(x_{\text{ref}}(t), y_{\text{ref}}(t))$ is found by moving along the path for a constant number of navigation points. The tangent heading to the path at point G, $\theta_{\text{ref}}(t)$, is calculated by the neighbor point $(x_{\text{refn}}(t), y_{\text{refn}}(t))$ next to G. Then, the feedforward loop input error $\Delta\alpha(t)$ is given by

$$\Delta\alpha(t) = \arctan\left(\frac{y_{\text{refn}}(t) - y_{\text{ref}}(t)}{x_{\text{refn}}(t) - x_{\text{ref}}(t)}\right) - \theta(t) \qquad (6)$$

4. Based on the kinematics model, the robot platform with differential wheel control turns by using different wheel speeds. Let $\Delta v = v_r - v_l$. The control law Δv was then calculated using the proportional-integral (PI) control based on heading errors and radial errors, expressed as equation (7)

$$\begin{aligned}\Delta v(t) = {} & (K_{\text{FFP}} \times \Delta\alpha(t) + K_{\text{FFI}} \times \int_0^t \Delta\alpha(\tau)d\tau) \\ & + (K_{\text{FBP}} \times \Delta e(t) + K_{\text{FBI}} \times \int_0^t \Delta e(\tau)d\tau)\end{aligned} \qquad (7)$$

where K_{FFP} is the feedforward proportional gain, a tuning parameter; K_{FFI} is the feedforward integral gain, a tuning parameter; K_{FBP} is the Feedback proportional gain, a tuning parameter; K_{FBI} is the Feedback integral gain, a tuning parameter; $\Delta\alpha(t)$ is the heading errors of lookahead point at time t; and $\Delta e(t)$ is the radial errors at time t.

From (3), (4), and (7)

$$\begin{aligned} v_{\text{ld}} = V_{\text{set}} - \frac{1}{2}\Big(&K_{\text{FFP}} \times \Delta\alpha(t) + K_{\text{FFI}} \times \int_0^t \Delta\alpha(\tau)d\tau \\ &+ K_{\text{FBP}} \times \Delta e(t) + K_{\text{FBI}} \times \int_0^t \Delta e(\tau)d\tau\Big) \end{aligned} \tag{8}$$

$$\begin{aligned} v_{\text{rd}} = V_{\text{set}} + \frac{1}{2}\Big(&K_{\text{FFP}} \times \Delta\alpha(t) + K_{\text{FFI}} \times \int_0^t \Delta\alpha(\tau)d\tau \\ &+ K_{\text{FBP}} \times \Delta e(t) + K_{\text{FBI}} \times \int_0^t \Delta e(\tau)d\tau\Big) \end{aligned} \tag{9}$$

where V_{set} is the robot forward velocity; v_{ld} is the left track desired linear velocity; and v_{rd} is the right track desired linear velocity.

In this study, the FPID controller focuses more on steering than on controlling the linear velocity. Thus, equation (10) was applied to computing a new motor linear velocity if the original value exceeds a motor's maximum forward linear velocity

$$v_{\text{nl}} = \begin{cases} v_{\text{ld}} - (\max(v_{\text{ld}}, v_{\text{rd}}) - vl_{\max}) & \text{if } \max(v_{\text{ld}}, v_{\text{rd}}) > vl_{\max} \\ v_{\text{ld}} - (\min(v_{\text{ld}}, v_{\text{rd}}) + vl_{\max}) & \text{if } \min(v_{\text{ld}}, v_{\text{rd}}) < -vl_{\max} \\ v_{\text{ld}} & \end{cases} \tag{10}$$

where $vl_{\max}$ is the left motor's maximum forward linear velocity; v_{nl} is the new left track desired linear velocity; and v_{nr} can be calculated in a similar way.

Results and discussion

To test whether the proposed algorithm works practically on the differential drive robot platform and to compare the performance of the FPID, feedforward, and feedback controllers, several experiments are made for the robot platform. The FPID, feedforward, and feedback controllers are implemented in C++ and tested on a laptop computer with a Core i5 2.5 GHz processor and 4G RAM. The GNSS updates the robot position at about 20 Hz and the controller runs at 10 Hz on average. Experiments were conducted to test on flat paved ground. The robot was run at an average speed of 0.8 m/s for the straight path and it was slowed down to an average of 0.2 m/s for the headland turning. The feedforward and feedback controllers were actually the feedforward and feedback loops of the FPID controller, respectively, and they were tuned to an optimal level by trial and error after the other loop was disconnected. In the feedforward controller tuning process, the lookahead distance was fixed and the parameters were then tuned manually. The feedback controller was also using the manual tuning method. Two straight reference paths were obtained by GNSS and a semicircle arc with 1-m radius reference path was created using the Dubins algorithm based on planning points obtain by GNSS.[35] For each controller, three turnings were performed to ensure replication, resulting in a total of nine test turns.

Headland turning consists of two parts. In the first part of the process, the robot starts tracking the predefined path using the controllers presented in this work. Radial errors were calculated using equation (11). The yaw angle of the robot was monitored using the GNSS. In the second part of the process, when the robot had turned by more than 165° with respect to the initial heading angle before turning, the straight-line path-tracking controller was used instead. The last 15° of rotation was performed in this part using the straight path-tracking algorithm so that the vehicle does not turn more owing to momentum. Lateral errors instead of radial errors were calculated based on the robot center to the straight path. The details of the controller design were presented in reference.[36] Therefore, any lateral error in entering the new row is corrected using the straight-line path-tracking algorithm

$$r(t) = D(t) - r_{\text{ref}} \tag{11}$$

where $D(t)$ is the distance from robot center to semicircle center, $D > 0$ for the case where the robot is to the right of the path; and r_{ref} is the radius of reference path.

Figure 6(a) shows the comparison of the performance of the feedforward, feedback, and FPID controllers in the first part of the process. The average radial errors of the feedforward, feedback, and FPID controllers were, respectively, 0.3844, 0.6867, and 0.3925 m, while the average standard deviations (SDs) were 0.4050, 0.7295, and 0.2872. The average root mean squares (RMSs) were 0.5574, 1.0005, and 0.4865, while the maximum radial errors were 1.3709, 2.2953, and 0.8284 m. It was noted that, in the first part of the turning process, the feedback controller had the worst performance. The feedback control reacted only to the process error and took corrective action only after the disturbances had affected the process and generated an error.[37] This time lag before the deviation of output occurs will enlarge the errors or even create instability of the system during the first part of turning. In contrast, the feedforward and FPID controllers predicted the steering error using the lookahead point and took corrective action before the disturbance entered into the process. Therefore, both the feedforward and FPID controllers had better performance than the feedback controller in this part. However, a pure feedforward controller is an open-loop control and is not self-correcting. If the input adjustments failed to produce a correct output, then the process will continue to produce the wrong output. The FPID balances the capability of the

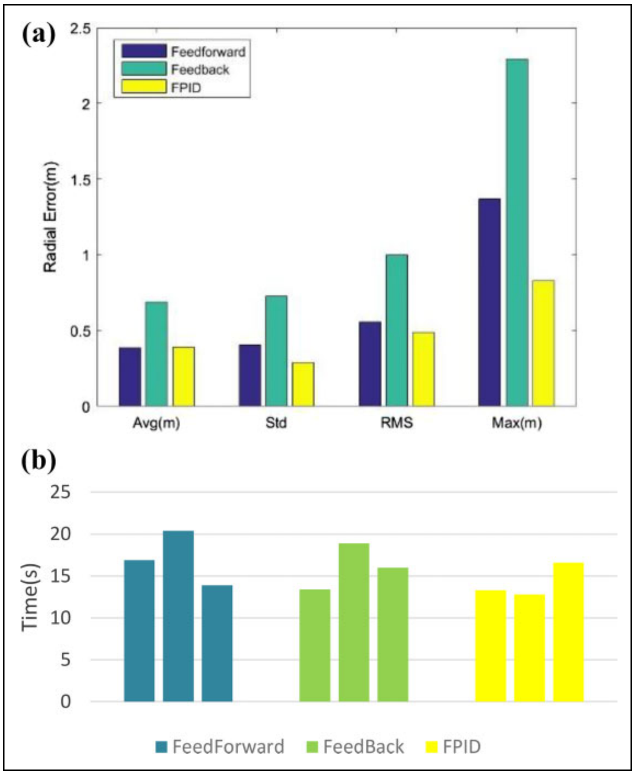

Figure 6. (a) Comparison of performance in the first part of turning. (b) Turning time comparison.

feedforward controller to take preemptive control actions for a heading deviation, while it permits the traditional feedback control loop to provide set point tracking capability and rejects all other disturbances and external influences that are not measured. In addition, the feedforward loop in the FPID minimizes the level of compensation required by the steering feedback, thus reducing the tracking errors and allowing for less overall control effort. Overall, the FPID controller results in lower tracking errors compared with the feedforward and feedback controllers in the first part of turning. Moreover, as Figure 6(b) shows, it was also found that under the same robot forward velocity and optimal control gain for each controller condition, the FPID controller had less turning time than that of the other controllers in the first part of turning.

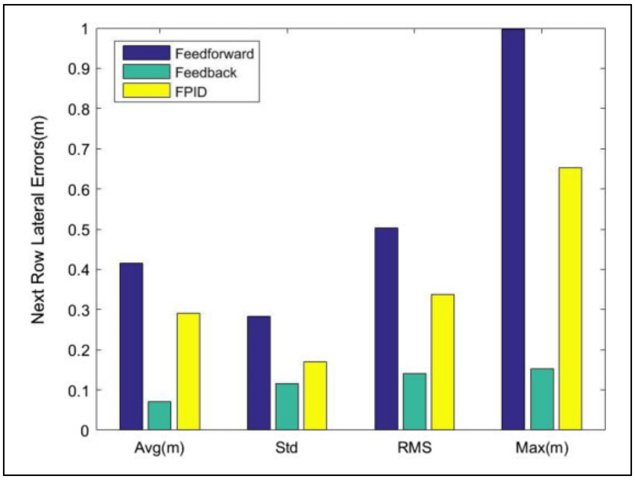

Figure 7. Comparison of performance in the second part of turning.

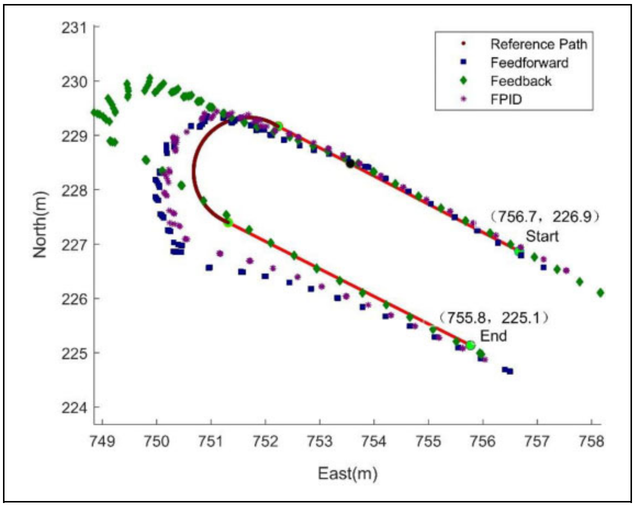

Figure 8. Reference path and the actual position of the robot.

Figure 7 shows the comparison of performance in the second part of turning in the second part of the process. The average lateral errors of the feedforward, feedback, and FPID controllers were 0.4157, 0.0701, and 0.2909 m, respectively, while the average SDs were 0.2839, 0.1154, and 0.1705. The average RMSs were 0.5026, 0.1412, and 0.3371, while the maximum radial errors were 0.9979, 0.1532, and 0.6524 m. From the plots of the second-part performance comparison, it was found that, in general, the path tracking of the feedback controller had a better performance than that of the other two controllers. One of the main reasons for this was that large radial errors occurred when the robot came close to the top of the headland turning owing to the feedback controller's time lag errors correction. This large error causes a very large differential drive control for the robot's left and right tracks and makes the robot's heading to change suddenly, which causes that the second part of turning becomes active earlier than that with the other two controllers. Therefore, the robot enters to the straight path-tracking mode with less lateral errors. However, owing to the large radial error in the first part of turning, as shown in Figure 8, the feedback control mode occupied the largest headland space. In contrast, the FPID occupied the smallest headland space. By integrating the feedforward and feedback controllers, the resulting FPID controller showed its capability in compensating both the lookahead point heading errors and radial tracking errors effectively.

Conclusions

An FPID headland-turning controller for a differential drive robot was developed in this study. The controller consisted of a feedforward loop and a feedback loop. The feedforward loop was designed based on the heading errors of a lookahead point on the planning path and the feedback loop was designed based on radial errors to improve the steering accuracy. Field comparison tests of the feedforward, feedback, and FPID controllers were conducted. Experimental results showed that the FPID could provide accurate steering control and could reduce the radial tracking error from a finely tuned feedforward and feedback loop. Moreover, the FPID headland turns decrease the proportion of the field area that constitutes the headland area and, consequently, the proportion of low-productivity area, which results in economic benefits. In our study, the robot headland-turning velocity was set as fixed. However, it was found that the feedforward loop in the FPID steering command becomes increasingly sensitive as the vehicle speed increases. A common fix for this issue is to scale the lookahead distance with the vehicle speed. This should be done in future work. Moreover, attention must be paid to providing an accurate kinematic model that considers more variations, such as the sideslip factor.

Acknowledgement

The authors are grateful to the student Zhao Yunxia and Lin Guchen for their help to the experiments.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the International S&T Cooperation Program of China (2015DFG12280) and the Science and Technology Planning Project of Guangdong Province, China (2016B020205003).

ORCID iDs

Peichen Huang https://orcid.org/0000-0002-1023-1342
Zhigang Zhang https://orcid.org/0000-0002-8232-4900

References

1. Shalal N, Low T, McCarthy C, et al. A review of autonomous navigation systems in agricultural environments. In: *2013 Society for Engineering in Agriculture Conference: Innovative Agricultural Technologies for a Sustainable Future*, Barton, Western Australia, 22–25 September 2013, pp. 10.
2. Moorehead SJ, Kise M, and Reid JF. Autonomous tractors for citrus grove operations. In: *2nd International Conference on Machine Control & Guidance*, Bonn, 9-11 March 2010, pp. 309–313.
3. Thanpattranon P, Ahamed T, and Takigawa T. Navigation of an autonomous tractor for a row-type tree plantation using a laser range finder-development of a point-to-go algorithm. *Robotics* 2015; 4(3): 341–364.
4. Andersen JC, Ravn O, and Andersen NA. Autonomous rule-based robot navigation in orchards. *IFAC Proceed Vol* 2010; 43(16): 43–48.
5. Radcliffe J, Cox J, and Bulanon DM. Machine vision for orchard navigation. *Comput Ind* 2018; 98: 165–171.
6. Torres-Sospedra J and Nebot P. A new approach to visual-based sensory system for navigation into orange groves. *Sensors* 2011; 11(4): 4086–4103.
7. Miller MA, Steward BL, and Westphalen M L. Effects of multi-mode four-wheel steering on sprayer machine performance. *Trans ASAE* 2004; 47(2): 385–395.
8. Kise M, Noguchi N, Ishii K, et al. Enhancement of turning accuracy by path planning for robot tractor. In: *Automation Technology for Off-Road Equipment Proceedings of the 2002 Conference. American Society of Agricultural and Biological Engineers*, Chicago, IL, USA, 26–27 July 2002, pp. 398.
9. Hague T, Marchant JA, and Tillett ND. Autonomous robot navigation for precision horticulture. In: *Proceedings of International Conference on Robotics and Automation*, Albuquerque, NM, USA, 25–25 April 1997, vol. 3, pp. 1880–1885. IEEE.
10. Subramanian V and Burks TF. Autonomous vehicle turning in the headlands of citrus groves. In: *2007 ASAE Annual Meeting. American Society of Agricultural and Biological Engineers*, Minnesota, 17-20 June 2007, pp. 1.
11. Cariou C, Lenain R, Thuilot B, et al. Autonomous maneuver of a farm vehicle with a trailed implement: motion planner and lateral-longitudinal controllers. In: *2010 IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, 3–7 May 2010, pp. 3819–3824. IEEE.
12. Backman J, Piirainen P, and Oksanen T. Smooth turning path generation for agricultural vehicles in headlands. *Biosyst Eng* 2015; 139: 76–86.
13. Bochtis DD and Vougioukas SG. Minimizing the non-working distance travelled by machines operating in a headland field pattern. *Biosyst Eng* 2008; 101(1): 1–12.
14. Thamrin NM, Arshad NHM, Adnan R, et al. A turning scheme in the headland of agricultural fields for autonomous robot. *ARPN J Eng Appl Sci* 2006; 11(5): 3265–3269.
15. Sabelhaus D, Röben F, zu Helligen LPM, et al. Using continuous-curvature paths to generate feasible headland turn manoeuvres. *Biosyst Eng* 2013; 116(4): 399–409.
16. Sabelhaus D, Schulze Lammers P, Helligen Zu LPM, et al. Path planning of headland turn manoeuvres. *Landtechnik* 2015; 70(4): 123–131.
17. Rovira-Más F, Saiz-Rubio V, and Millot C. Sonar-based aid for the execution of headland turns by a vineyard robot. In: *2016 ASABE Annual International Meeting. American Society of Agricultural and Biological Engineers*, Orlando, FL, USA, 17–20 July 2016, pp. 1.
18. Xue JL and Grift TE. Agricultural robot turning in the headland of corn fields. *Appl Mech Mater* 2011; 63: 780–784. Trans Tech Publications.
19. Noguchi N, Ishii K, and Terao H. Turning function based on dynamic path creation for agricultural mobile robot. *IFAC Proceed Vol* 2001; 34(19): 191–195.

20. Bräunl T. *Embedded robotics: mobile robot design and applications with embedded systems*. Berlin: Springer Science & Business Media, 2008.
21. Veronesi M and Visioli A. Automatic feedforward tuning for PID control loops. In *2013 European Control Conference (ECC)*, Zürich, Switzerland, July 17–19 2013, pp. 3919–3924, IEEE.
22. Visioli A. A new design for a PID plus feedforward controller. *J Process Control* 2004; 14(4): 457–463.
23. Petersson M, Årzén KE, and Hägglund T. A comparison of two feedforward control structure assessment methods. *Int J Adapt Control Signal Process* 2003; 17(7–9): 609–624.
24. Vilanova R, Arrieta O, and Ponsa P. IMC based feedforward controller framework for disturbance attenuation on uncertain systems. *ISA Trans* 2009; 48(4): 439–448.
25. Shladover SE, Desoer CA, Hedrick JK, et al. Automated vehicle control developments in the PATH program. *IEEE Trans Vehicul Technol* 1991; 40(1): 114–130.
26. Mammar S and Koenig D. Vehicle handling improvement by active steering. *Vehicle Syst Dynam* 2002, 38(3): 211–242.
27. Raharijaona T, Duc G, and Mammar S.Robust control and μ-analysis with application to lateral driving assistance. In: *Proceedings of IEEE Intelligent Transportation Systems*, Shangai, China, 12–15 October 2003, pp. 247–252.
28. Starrenburg JG, Van Luenen WTC, Oelen W, et al. Learning feedforward controller for a mobile robot vehicle. *Control Eng Pract* 1996; 4(9): 1221–1230.
29. Shin D H, Singh S, and Lee JJ. Explicit path tracking by autonomous vehicles. *Robotica* 1992; 10(6): 539–554.
30. Hingwe P and Tomizuka M. A variable look-ahead controller for lateral guidance of four wheeled vehicles. In: *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, Philadelphia, PA, USA, 26–26 June 1998, vol. 1, pp. 31–35, IEEE.
31. Rossetter EJ. *A potential field framework for active vehicle lanekeeping assistance*. Stanford: Stanford University, 2003.
32. Oriolo G, De Luca A, and Vendittelli M. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Trans Control Syst Technol* 2002; 10(6): 835–852.
33. Asif M and Junaid KM. Feedforward and feedback kinematic controllers for wheeled mobile robot trajectory tracking. *J Automat Control Eng* 2015; 3(3):178–182.
34. Wei SU. A model reference-based adaptive PID controller for robot motion control of not explicitly known systems. *Int J Intell Control Syst* 2007; 12(3): 237–244.
35. Dubins LE. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am J Math* 1957; 79(3): 497–516.
36. Huang P, Zhang Z, Luo X, et al. Path tracking control of a differential-drive tracked robot based on look-ahead distance. *IFAC-Papers Online* 2018; 51(17): 112–117.
37. Franklin GF, Powell JD, Emami-Naeini A, et al. *Feedback control of dynamic systems*. Reading: Addison-Wesley, 1994.