Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 51-17 (2018) 112–117

Path Tracking Control of a Differential-Drive Tracked Robot Based on Look-ahead Distance

Peichen Huang.* Zhigang Zhang.* Xiwen Luo.*
Jian Zhang .* Peikui Huang*

**Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, South China Agricultural University, Guangzhou 510642, China (Tel: +86 13570495304; e-mail: zzg208@ scau.edu.cn).*

Abstract: In this paper, a PI path tracking controller based on look-ahead point information for a differential-drive crawler-type robot was developed. The robot is equipped with Global Navigation Satellite System (GNSS) antennas and a receiver module which are used as navigation sensors to obtain real-time estimates of the position and heading errors of the robot. To control the position of the robot with respect to the target path, a look-ahead point was generated by using Euler Transformation. The look-ahead point lateral errors were then controlled by the designed controller. Forward and backward path tracking tests with different look-ahead distances were conducted to examine the controller performance of the robot and the effect of changing look-ahead distance. Through experiment results, it was noted that the lateral errors mean value was within 0.05 m for forward path tracking and 0.06 m for backward path tracking while the heading errors mean value was within 0.5 degree for forward path tracking and 1 degree for backward path tracking. The proposed algorithms were suitable for the differential-drive robot platform for forward and backward path tracking with a reasonable look-ahead distance.

Keywords: crawler-type robot, differential-drive, path tracking, Dual-antenna RTK-GPS, look-ahead distance.

1. INTRODUCTION

Mobile robots play a significant role in many agricultural applications since they reduce human labour and enhance the operation safety. The need for autonomous navigation systems of mobile robots has been recognised in different agricultural tasks such as planting, spraying, fertilizing, cultivating, harvesting, thinning, weeding, and inspection (Shalal, N,2013). Robot automation is a growing interest among the agricultural community. Many feasibility studies were made on autonomous agricultural vehicles.

The basic task in mobile robot motion control is to accurately follow a given trajectory. Different control strategies have been researched in the past few decades. In the work investigated by Subramanian et al. (2006), a PID controller was developed to minimize the path error and steer the tractor through the alleyway of a citrus grove using the information from the vision system and laser radar. In some agricultural application, Neural Network(NN) and Genetic Algorithm(GA) have been applied to control the motion of the mobile robot. Zhu et al. (2005) developed a NN vehicle model for estimating vehicle behaviour on sloping terrain. A training method combined with GA and back propagation algorithm was used to train the NN vehicle model. The tractor was successfully guided along a predetermined path. There are also path tracking algorithms using Fuzzy-Logic has been implement on agricultural robot. Martín et al. (2010) described the implementation of a fuzzy steering controller for safe obstacle avoidance in unmanned navigation of a robot tractor under different ground conditions.

Most of the previous research focus on wheel-type robot due to their applicability to the wide range of challenging situations with developing control techniques. A crawler-type robot is a vehicle with tracks instead of wheels. Crawler-type tractors are also used in some agricultural fields because of their high traction and low ground pressure. Faik et al. (2017) designed A PID and kinematic based backstepping controller for a differential-drive mobile robot to be able to track a desired trajectory. Matlab/Simulink results showed a good tracking performance and stability for the different initial positions. Ryosuke et al. (2011) modified a commercial crawler-type tractor into a robot tractor that can be navigated autonomously by using RTK-GPS and IMU as navigation sensors. Experiment results showed that the robot could finish the multipath navigation successfully. Murakami et al. (2004) designed a teleoperation system for a hydro-static transmission (HST) drive crawler-type robotic vehicle. The vehicle could travel in a straight line with a maximum lateral error of 0.3 m by using supervisory control. Michihisa Iida et al. (2013) construct an advanced harvesting system by using a combine robot. A real-time kinematic global positioning system (RTK-GPS) and a GPS compass are used as navigation sensors. A Kalman filter that estimates the lateral and heading errors to control steering. Through field

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2018.08.072

experiments, the combine robot is capable of automatically following a target spiral path and harvesting rice crops.

In this paper, we consider the problem of controlling the path tracking of a differential-drive crawler-type robot. The robot is equipped with Global Navigation Satellite System (GNSS) antennas and a receiver module which are used as navigation sensors to obtained real-time estimates of the position and heading errors of the robot. To control the position of the robot with respect to the target path, a look-ahead point was generated by using Euler Transformation. Then an PI controller was designed to control the lateral errors of the look-ahead point. Forward and backward path tracking tests with different look-ahead distance were conducted to examine the controller performance of the robot and the effect of changing look-ahead distance.

2. MATERIALS AND METHODS

2.1 Research platform and navigation hardware

A crawler-type robot platform was used in this research. Figure1 shows the research platform, a Komodo-01. Table.1 shows its specification. GNSS antennas and a Trimble BD982 receiver were used as navigation sensors to determine lateral and heading errors to evaluate the autonomous navigations. To obtain the absolute position of the robot platform, a base station was used with 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 GUI for analysing the algorithms behaviour 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 schematic diagram of the robot platform.

Fig.1. Research platform(Komodo-01)

Table 1. Specifications of a tracked robot

Model	Komodo-01	Driving System	Crawler
Dimensions	650*350*240, mm	Max Speed	6km/s
Power Rating	2*650W	Wheel Diameter	158, mm
Gearbox Ratio	15:1	Max Load	100, Kg
Track Width	150, mm	Climbing Capacity	40, degree

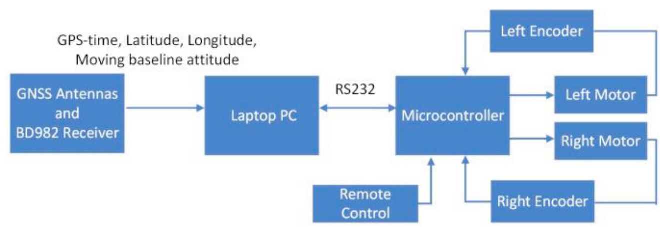

Fig.2. Schematic diagram of a robot platform

2.2 Kinematics of the Differential Drive Mobile Robot

Kinematic is the study of the motion without considering the forces. The purpose of the kinematic modelling is to derive robot velocities as a function of the driving wheels' velocities in predefined. A tracked mobile robot can be seen as a special case of a wheeled robot with differential drive. The difference is the robot's better manoeuvrability in rough terrain and its higher friction in turns, due to its tracks and multiple points of contact with the surface(Bräunl,2006).

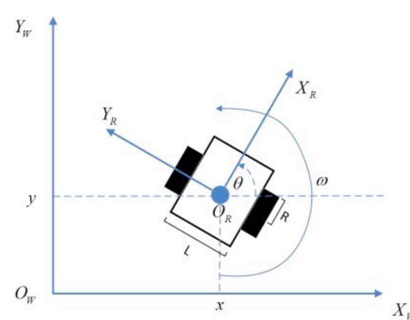

Fig.3. Kinematics model of differential-drive robot

As Figure 3 shows, the configuration of the robot platform is represented by the generalized coordinates constraints $q=(x,y,\theta)$,where (x,y) is the position and θ is the heading of the centre 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 is robot's linear velocity in robot frame's x-direction, L is distance between the wheels, R is 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:

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

The kinematic model in the world frame is given by:

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

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

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

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

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.

2.3 Controller Design

Path tracking is the process concerned with how to determine speed and steering settings at each instant of time in order for the robot to follow a certain path. In this paper, the robot platform was tended to think of as chasing a point on the path some distance ahead of it which was like the way human drive. The lateral error was then calculated using this point. As Figure 4 shows, *M* point stands for the primary antenna which its absolute position was obtained firstly. Then the position of the look-ahead point *G*, which was on the ground, was calculated by Euler Transformation using *M* point position and attitude.

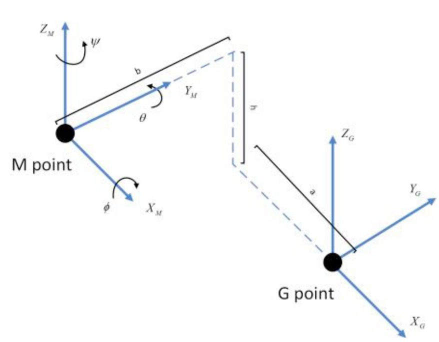

Fig.4. Project M point to G point using Euler Transformation

The values of parameters *b*, *h* was obtained by measuring, θ, ϕ, φ was obtained by GNSS receiver and look-ahead distance *a* was set manually.

Based on the kinematics model, the robot platform with differential wheel control turns by using different wheel speeds. Let $\Delta v = v_r - v_l$. Δv was then calculated using PI control based on lateral error shown as (6).

$$\Delta v(t) = K_p * Pe(t) + K_I * \int_0^t Pe(\tau) d\tau \tag{6}$$

Where,

K_P : Proportional gain, a tuning parameter

K_I : Integral gain, a tuning parameter

$Pe(t)$: Look-ahead point G's lateral error at time t

From (1) and (6),

$$v_{ld} = V_{set} - \frac{1}{2}(K_p * Pe(t) + K_I * \int_0^t Pe(\tau) d\tau) \tag{7}$$

$$v_{rd} = V_{set} + \frac{1}{2}(K_p * Pe(t) + K_I * \int_0^t Pe(\tau) d\tau) \tag{8}$$

Where,

V_{set} : Robot moving velocity, setting manually

v_{ld} : Left track desired linear velocity

v_{rd} : Right track desired linear velocity

In this paper, the PI controller focus more on steering than on controlling the linear velocity. So, (10) was applied to computing a new motor linear velocity if the original value exceeds a motor's maximum forward linear velocity.

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

Where,

$vl_{\max}$: Left motor's maximum forward linear velocity

v_{nl} : New Left track desired linear velocity

Then v_{nr} can be calculated in a similar way.

3. EXPERIMENT

In order to test whether the proposed algorithm works practically on the differential-drive robot platform and how changing the look-ahead distance effects the path tracking results, several experiments are made for the robot platform. The algorithm is implemented in C++ and tested on a laptop computer with Core i5 2.5 GHz and 4G RAM. The GNSS updates the path errors at about 20 Hz and the proposed algorithm runs at 10 Hz on average. The robot platform moving velocity was set at 0.8m/s in all experiments.

At first, Ziegler Nichols method was adopted for initial tuning of the PI controller. In this tuning process, the system was given a small proportional gain. Integral gains were removed and the look-ahead distance was fix at 3 m. The gain was increased until sustained oscillations were obtained when $K_P = 0.6$. The required proportional gain K_P and integral gain K_I were calculated using the Ziegler Nichols formula. The parameters were further tuned. Finally, the optimum PI gains were found to be: $K_P = 0.3$, $K_I = 0.03$.

Single-line path navigation test was then conducted on a road to test the accuracy of steering control. Since sometimes robot have to drive backwards, for example, in the narrow region or headland. The path tracking performance of the backward paths should be guaranteed as well as that of the forward paths (Kim M, 2016). Therefore, forward path tracking and backward path tracking tests were performed to evaluate the algorithm.

Finally, experiments were conducted to test effects of changing the look-ahead distance. For both forward and backward path tracking, the robot was manually driven to the same starting position and then run at five different look-ahead distances, 1.5 m, 2 m, 3 m, 4 m, 5 m, result in a total 10 test run. The lateral errors and heading errors were recorded and analysed.

4. RESULTS AND DISCUSSION

The results of the robot forward and backward path tracking performance from the test are shown in Figure5 and Figure6. It was noted that the lateral errors mean value was within 0.05 m for forward path tracking and 0.06 m for backward path tracking while the heading errors mean value was within 0.5 degree for forward path tracking and 1 degree for backward path tracking. From the performance comparison plots, it was also found that the forward path tracking had a better performance than the backward path tracking in general. One of reasons for this was that the designed PI controller's parameters was tuned under the robot moving forward circumstances. Therefore, the tuning parameters may be a slightly different from the parameters for moving backward. Overall, the proposed algorithm was suitable for the differential drive robot platform for forward and backward path tracking with a reasonable look-ahead distance.

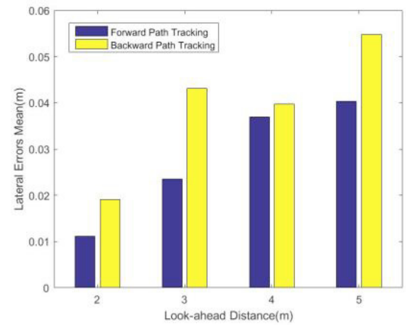

Fig.5(a). Mean comparison

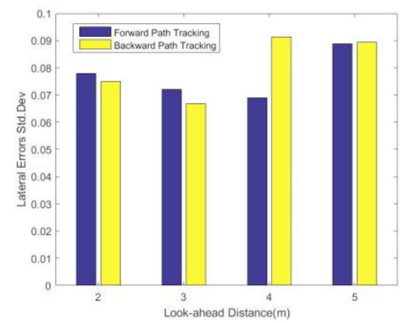

Fig.5(b). Standard deviation comparison

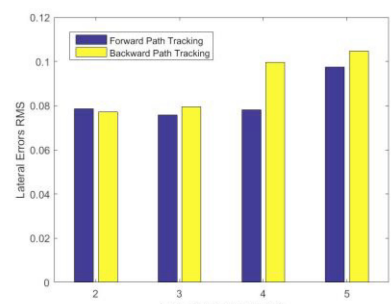

Fig.5(c). R.M.S comparison

Fig.5. Lateral errors comparison of forward and backward path tracking

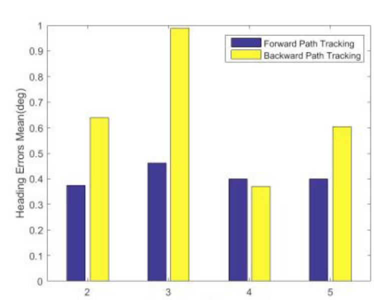

Fig.6(a). Mean comparison

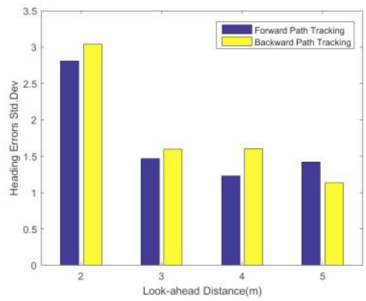

Fig.6(b). Standard deviation comparison

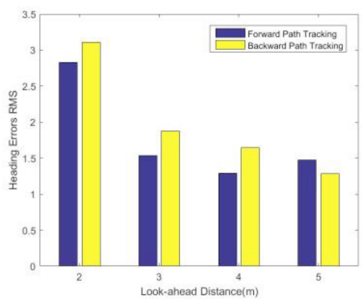

Fig.6(c). R.M.S comparison

Fig.6. Heading errors comparison of forward and backward path tracking

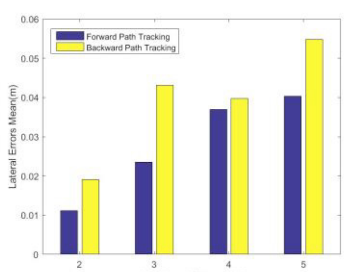

Fig.7(a). Mean Value

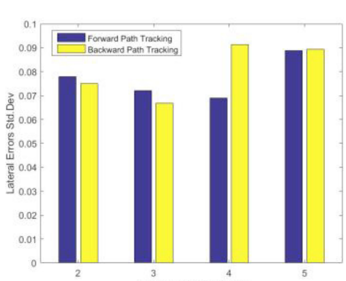

Fig.7(b). Standard deviation

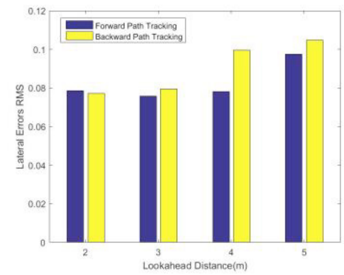

Fig.7(c). R.M.S

Fig.7. Lateral Errors with different look-ahead distances

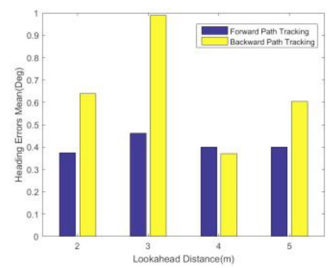

Fig.8(a). Mean Value

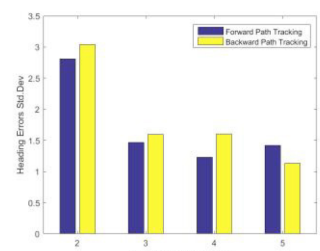

Fig.8(b). Standard deviation

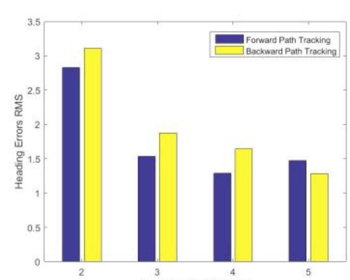

Fig.8(c) R.M.S

Fig.8. Heading Errors with different look-ahead distances

Figure 7 and Figure 8 shows the lateral and heading errors with different look-ahead distance statistics results respectively. It has been noticed that changing the look-ahead distance can have a significant effect on the performance of the algorithm. As Figure8(b) and Figure8 (c) shows, short look-ahead distances cause the robot to regain the path quicker but may result in oscillations about the path. In our experiment, the robot became oscillations and unstable after a short time during both forward path tracking and backward path tracking with look-ahead distance 1.5*m*. It was shown that the robot could not maintain the path with too short look-ahead distance. Within the reasonable range, enlarge the look-ahead distance may cause the robot smooth regaining the path with less oscillations. The standard deviation of heading errors dropped down from 2.8061 to 1.2311 for forward path tracking and 3.0434 to 1.6034 for backward path tracking when the look-ahead distance was enlarged from 2 m to 4 m. However, large look-ahead distance will lead to worse tracking during the same amount of tracking time since the robot had to take more time to regain the path. The forward path tracking mean lateral errors was 0.0112 m, 0.0235 m, 0.0369 m with look-ahead distance 2 m, 3 m, 4 m while the backward path tracking mean lateral errors was 0.0191 m, 0.0431 m, 0.0398 m.

Furthermore, Figure7(b) and Figure7(c) shows that the robot became more oscillations with look-ahead distance 5 m during both forward and backward path tracking. This was because when the robot came closely to the path, even with few heading errors, the lateral errors would be enlarged for the look-ahead point. This would cause very large differential drive for the robot platform. Another factor was the robot's speed. High speed requires large look-ahead distance. The reason for this is that higher speeds requires the robot to start turning at an earlier stage. However, in our project, the speed of the robot was low. So too large look-ahead distance would not be appropriate. From experimental data, there existed a reasonable interval of look-ahead distance. In our case, a good choice of a look-ahead distance is between 2 m~3 m.

6. CONCLUSIONS

In this paper, a PI controller based on look-ahead point information for a differential-drive crawler-type robot motion control was developed. The Look-ahead point was generated by using Euler Transformation and real-time estimates of the position and heading errors of the robot which was obtained through Dual-antenna RTK-GPS. The look-ahead lateral error was then calculated and controlled using PI controller. Forward and backward path tracking with different look-ahead distance tests were conducted and the result shows that the algorithm was suitable for the differential drive robot platform for forward and backward path tracking with a reasonable look-ahead distance.

In future work, attention must be paid to providing an accurate kinemics model which considers more variations. Additional work is required to find the optimal look-ahead distance by considering the robot speed will be the key point for increasing the tracking accuracy

REFERENCES

Faik DEMİRBAŞ, Mete KALYONCU.(2017). Differential drive mobile robot trajectory tracking with using PID and kinematic based backstepping controller. Selcuk University Journal of Engineering, Science & Technology / Selçuk Üniversitesi Mühendislik, Bilim ve Teknoloji Dergisi . Vol. 5 Issue 1, p1-15. 15p.

Iida, M, Masahiko, S, & Uchida, R. (2013). Advanced harvesting system by using a combine robot. Bio-Robotics.Vol.46, pp.40-44.

Kim M, Shin S, Park J. Study on vehicle lateral control for backward driving[C].International Conference on Ubiquitous Robots and Ambient Intelligence. IEEE, 2016:191-193.

Martín, D., Guinea, D., García-Alegre, M., Villanueva, E. & Guinea, D. (2010). Fuzzy steering control of a hydraulic tractor and laser perception obstacle avoidance strategies, Proceedings of the XV Congreso Espanol Sobre Tecnologıas y Logica Fuzzy (ESTYLF), Huelva, Spain, p. 349-355.

Murakami, N., K. Inoue, and S. Miyaura. 2004.Teleoperation platform for HST drive agricultural vehicle. In: Proc. Automation Technology for Off-road Equipment, Kyoto, Japan. Paper701P1004.

Shalal, N. and Low, T. and McCarthy, C. and Hancock, N. (2013) A review of autonomous navigation systems in agricultural environments. In: SEAg 2013: Innovative Agricultural Technologies for a Sustainable Future, 22-25 Sept 2013, Barton, Western Australia.

Subramanian, V., Burks, T. F. & Arroyo, A. (2006). Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation, Computers and Electronics in Agriculture, 53: 130-143.

Subramanian, V., Burks, T. & Dixon, W. (2009). Sensor fusion using fuzzy logic enhanced Kalman filter for autonomous vehicle guidance in citrus groves, Transactions of the ASAE, 52: 1411-1422.

Takai, R., Barawid, O. J., & Noguchi, N. (2011). Autonomous Navigation System of Crawler-Type Robot Tractor.

Thomas Bräunl.(2006).Embedded Robotics: Mobile Robot Design and Applications with Embedded Systems. Chatper7,p102. Springer-Verlag Berlin Heidelberg, Printed in Germany

Zhu, Z., Torisu, R., Takeda, J., Mao, E. & Zhang, Q. (2005). Neural network for estimating vehicle behaviour on sloping terrain, Biosystems Engineering, 91: 403-411.

Ziegler, J.G & Nichols, N. B. (1942). "Optimum settings for automatic controllers" (PDF). Transactions of the ASME. 64: 759–768.