

LOCALIZATION AND MOTION CONTROL IMPLEMENTATION FOR AN AGRICULTURAL MOBILE ROBOT

Mohd Saiful Azimi Mahmud, Mohamad Shukri Zainal Abidin*, Zaharuddin Mohamed

Control and Mechatronics Department, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

Article history

Received
25 October 2016
Received in revised form
28 June 2017
Accepted
10 September 2017

*Corresponding author
shukri@fke.utm.my

Graphical abstract



Abstract

In robot navigation experiment, a localization and motion control system is required to secure the agricultural robot motion in the environment. However, the high cost of localization system and complex structure of motion controller has limited the low cost agricultural mobile robot development. In this paper, a low-cost localization system and simple motion control system is presented. The localization system has been implemented using a dead reckoning method by accessing an incremental encoder's reading. A simple cascaded motion control system based on proportional feedback kinematics controller and PI based controller was used to control the mobile robot motion. The performances of different turning methods: U turn and a π turn, were compared for lane changing, based on completion time, controller's error and distance travelled. Simulation test of robot motion was conducted using a Simulink3d animation in MATLAB software. An experimental test in a real greenhouse environment was conducted to verify the simulation performance in motion control and localization system. The experimental and simulation results have shown that a U turn has the best turning performance with 69.1 % better efficiency in experimental mode and it is recommended to be applied in agricultural field.

Keywords: Kinematics, localization, cascade control, dead reckoning, simulation

Abstrak

Dalam ujian navigasi robot, sistem penyetempatan dan pengawal gerakan robot diperlukan bagi memastikan pergerakan robot pertanian terjaga. Walau bagaimanapun, kos sistem penyetempatan yang agak tinggi dan struktur pengawal gerakan yang kompleks telah mengehak perkembangan robot pertanian kos rendah. Dalam manuskrip ini, sistem penyetempatan kos rendah dan pengawal gerakan robot yang ringkas telah dibentangkan. Sistem penyetempatan robot telah dijalankan menggunakan kaedah perhitungan mati dengan mengambil data daripada pengekod tambahan. Sistem pengawal gerakan robot selari yang ringkas digunakan berdasarkan pengawal sistem maklum balas berkadar dan system pengawal PI. Penilaian cara pusingan robot: pusingan U dan pusingan π telah dibandingkan berdasarkan masa selesai, ralat pengawal dan jarak perjalanan. Kajian simulasi bagi pergerakan robot telah dijalankan menggunakan animasi Simulink3d dalam perisian MATLAB. Kajian eksperimen dalam kawasan rumah hijau telah dijalankan bagi mengesahkan hasil kajian simulasi dalam sistem kawalan gerakan robot dan sistem penyetempatan. Hasil keputusan eksperimen menunjukkan bahawa pusingan U mempunyai prestasi pusingan yang lebih baik dengan 69.1% kecekapan dalam eksperimen dan disyorkan bagi navigasi di dalam rumah hijau

Kata kunci: Kinematik, penyetempatan, pengawal selari, perhitungan mati, simulasi

© 2017 Penerbit UTM Press. All rights reserved

1.0 INTRODUCTION

Agricultural field operations are recently becoming driven by technology as the operations are complex, diverse and labour-intensive [1]. As the predicted world population to be over 10 million in 2050, the agricultural productivity has continuously and significantly increase over time and thus the agricultural operations need to be enhanced [2]. In the past decades, enormous changes has been seen and with the evolution of new technologies, automation control and robotics has been proven to produce a higher agricultural productivity with lower production cost.

In agriculture, robots are needed to perform operations such as inspection [3-4], cultivation [5], transplanting [6], spraying [7] and selective harvesting [8-9]. Despite the tremendous amount of research, commercial application of robots in complex environment are not yet available [10]. Some of the applications of the agricultural environment are still in development stage [11-12]. The main reason behind the delay is the lack of robot implementation in the real agricultural environment as most of the invention were tested in the lab or simulated environment. Therefore, the implementation of mobile robot is needed to reduce the delay of producing a commercial agricultural robot.

In robot localization, several methods has been applied in agricultural field operation. Localization system such as Global Positioning System (GPS) [13], Real-Time Kinematics GPS (RTK-GPS) [14], Geographic Information System [15], Machine vision [16] and LIDAR based system [17] has been applied to the agricultural mobile robot system. Comparing from all the method mentioned, RTK-GPS method has been proven to achieve the highest accuracy in robot localization [18]. However, this method needs an extremely high cost as the sensor's price is very expensive. Therefore, this method was hardly used in agricultural application.

Machine vision was the most common method in guiding and localizing the mobile robot in agriculture. However, as the environment become more complex, the vision system become unstable and the mobile robot may collide with the environment. LIDAR based localization system provides an accurate mapping and mobile robot location. However, the accuracy of the system depends on the quality of the sensor. Therefore, in order to develop an accurate localization system, an expensive LIDAR sensor is needed.

For mobile robot motion control, methods such as an asymptotic stable controller [19-20], an adaptive controller [21], and a feedback linearization controller [22], have been designed. They have been proven to be a robust and effective controller for robot motion planning. However, the high computational costs of these methods have made the hardware implementation to become harder, as they need a high end system to be executed. A simple and easier control method, such as a proportional feedback kinematic controller as proposed by [23], has been proven to be a good motion tracking controller, as it was developed based on the robot kinematics. This

method has been widely used in motion planning applications, such as in [24].

Recently, a four-wheel-drive (4WD) autonomous greenhouse mobile robot platform was designed in [25]. It uses a vision system and equipped with a CCD camera. It was driven by a signal generated from Atmega128 controller to control the motor. However, only lab test was presented and the control system was not explained in detail. Therefore, the performance of control system was not validated as the real environment structure is more complex compared to the lab environment.

A path tracking of agricultural mobile robot has been evaluated in crop based environment in [26]. An adaptive PID, model reference adaptive controller and fuzzy controller was compared. For turning method, U-type turning based is used and the model reference adaptive controller was having the best performance in comparison. However, it is not possible to define clearly which is the best option for the robot as the difference is not much significant. In addition, only simulation result was shown and the controller may behaves differently in experimental implementation.

In this paper, an inexpensive localization system based on dead reckoning method will be implemented and tested in agricultural environment. A simple cascaded control system based on proportional feedback kinematics controller and PI controller will be used to control the robot motion and velocity. Two types of path will be compared based on different types of headland turn: U-Turn and π -Turn in terms of time taken, controller error and distance travelled. Simulation in virtual environment and experimental tests will be conducted to verify the system performance in addition to evaluate the path quality.

2.0 METHODOLOGY

Figure 1 shows an overview of the robot's trajectory control system. The block diagram shown is similar to the kinematic based motion control proposed in [27].

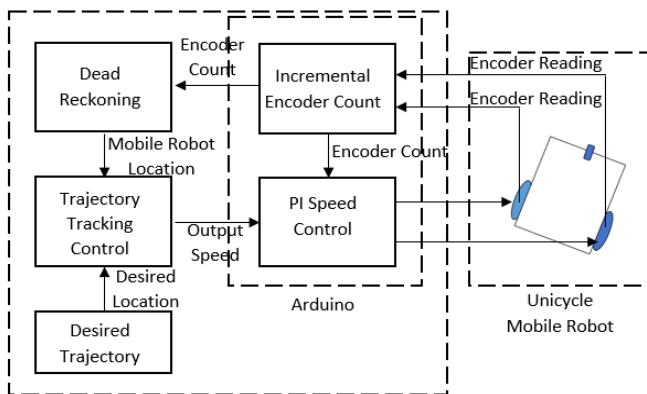


Figure 1 Cascaded mobile robot control system

Based on Figure 1, two types of controller were involved. The first controller was a proportional integral speed controller that was implemented on an Arduino Mega 2560 board. The second controller was implemented in a computer to control the robot motion. It was called as a feedback kinematic controller that was proposed in [23]. In this paper, the Arduino board was used in order to act as an interface between the mobile robot and the main controller (computer). For the localization, a dead reckoning method was chosen by accessing the odometry details.

2.1 Unicycle-Like Mobile Robot Model

In this paper, a unicycle-like mobile robot model has been chosen as it is the most common type of mobile robot that was used in various applications, such as in surveillance, floor cleaning and in autonomous wheelchair applications. In agricultural application, unicycle mobile robot has been used in [6] as a ploughing mobile robot. Figure 2 shows a unicycle-like mobile robot model by De La Cruz *et al.* [21].

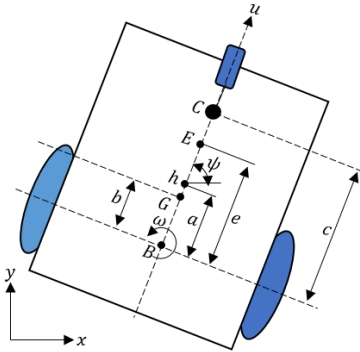


Figure 2 Unicycle-Like mobile robot structure

In order to ensure that the chosen robot model was able to traverse in agricultural environment, the position of the castor has been moved to the back position. It is mainly due to the uneven surface and it is easier for the mobile robot to pull the castor wheel instead of pushing the castor wheel in an uneven surface.

Based on Figure 2, u indicates the linear velocity of the mobile robot and ω indicates the angular velocity of the mobile robot. a indicates the distance between the point of interest and the central point that links the wheels, G is the robot's center of mass, C is the castor wheel position, E is the tool position of the robot, h is the point of interest in the inertial frame and ψ is the robot's orientation. The robot kinematics model is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} \quad (1)$$

The kinematics model in Equation 1 was derived from the transformational matrix between the robot's frame and the inertial frame. The unicycle-like mobile robot was defined as a non-holonomic robot, as the differential equation was not integrable into the final

position. Thus, the distance that each wheel traveled was not sufficient for the final position's calculation.

2.2 Feedback Kinematics Controller

The controller equation was derived from the inverse kinematics model that was obtained in Equation 1. The inverse kinematics is given by:

$$\begin{bmatrix} u \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\frac{1}{a} \sin \psi & \frac{1}{a} \cos \psi \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \quad (2)$$

From Equation 2, the kinematics based controller for the trajectory tracking was designed by applying the kinematics control law. Thus, the full kinematics based controller is given by:

$$\begin{bmatrix} u_{ref}^c \\ \omega_{ref}^c \end{bmatrix} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\frac{1}{a} \sin \psi & \frac{1}{a} \cos \psi \end{bmatrix} \begin{bmatrix} \dot{x}_d + l_x \tanh\left(\frac{k_x}{l_x} \tilde{x}\right) \\ \dot{y}_d + l_y \tanh\left(\frac{k_y}{l_y} \tilde{y}\right) \end{bmatrix} \quad (3)$$

Based on Equation 3, $\tilde{x} = x_d - x$ and $\tilde{y} = y_d - y$ are the errors in the current position of the XY axes, respectively. k_x and k_y are the gains of the controller. (x, y) and (x_d, y_d) are the current and the desired coordinates of the point of interest. The \tanh function was used as an error limiter when the analytical saturation of the velocity was included and where $l_x \in \mathcal{R}$ and $l_y \in \mathcal{R}$ were the saturation constants for the controller.

2.3 Mobile Robot Localization

There were two types of mobile robot localization: local and global. Local techniques aim at compensating for the odometric errors occurred during the robot's navigation and which require the initial location of the robot to be approximately known. Global techniques can localize a robot without any prior knowledge about its position. In this paper, local localization has been implemented by using the dead reckoning method. The equation for calculating the mobile robot's position and orientation in the inertial axis is given by:

$$x_k = x_{k-1} + \left(\frac{(R_r + R_l) \times W_r \times \pi}{T_s} \right) \cos \left(\frac{W_r}{W_d} (R_r - R_l) \right) \quad (4)$$

$$y_k = y_{k-1} + \left(\frac{(R_r + R_l) \times W_r \times \pi}{T_s} \right) \sin \left(\frac{W_r}{W_d} (R_r - R_l) \right) \quad (5)$$

$$\psi_k = \psi_{k-1} + \left(\frac{W_r}{W_d} \right) (R_r - R_l) \quad (6)$$

Based on equations 4 to 6, x_k , y_k and ψ_k are the current locations and orientation of the mobile robot while x_{k-1} , y_{k-1} and ψ_{k-1} indicate the previous locations and orientation of the mobile robot. R_r and R_l indicate the number of wheel rotations for the right and left wheels, respectively. W_r indicates the radius of the wheel, W_d is the axial distance between the wheels, and T_s is the sampling time for the encoder.

2.4 Mobile Robot Trajectory

Many techniques have been developed to optimize the field operation that focusing on minimizing operational time, cost and maundering over field area [28]. In agriculture, sharp turn is rarely used as it can cause severe damage to the soil structure. Therefore, soft turning type is needed which reduces the soil damage over headlands area [29].

Several turning methods such as U turn, π turn, Ω turn and Hook turn were used for the agriculture vehicle [30]. However, as the space between the headlands and the border of the greenhouse is limited, U turn and π turn is selected to be used in the greenhouse as it takes the least space to do the turning. Figure 3 shows the U turn and π turn respectively.

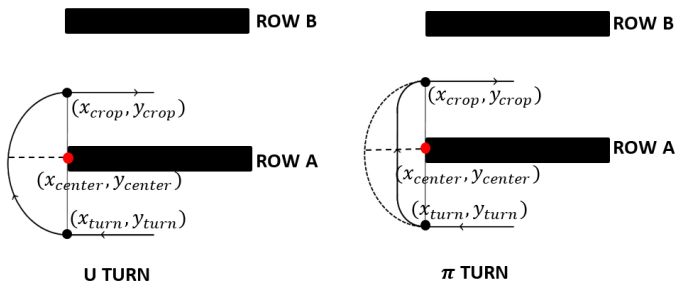


Figure 3 Robot headland pattern

Based on Figure 3, the coordinate of turning centre was computed based on the midpoint between turning point, (x_{turn}, y_{turn}) and crop point (x_{crop}, y_{crop}) . x_{crop} was computed based on the furthest obstacles in x-coordinate in the current lane. y_{crop} was obtained based on midpoint between crop in Row A and Row B in y-coordinate. U turn and π turn was computed based on Equation of Circle:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (7)$$

Where the circle has a centre of (a, b) and radius r . The formula for U turn is:

$$x = x_{center} + r \cos(\theta), y = y_{center} + r \sin(\theta), \quad \theta = 0, \dots, 180. \quad (8)$$

And for π turn:

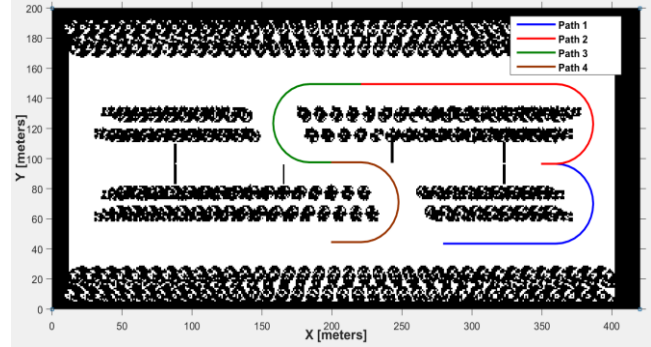
$$x = x_{center} + \frac{r}{2} \cos(\theta) + \frac{r}{2}, y = y_{center} + \frac{r}{2} \sin(\theta) + \frac{r}{2}, \quad \theta = 0, \dots, 90 \quad (9)$$

$$x = x_{center} + \frac{r}{2} \cos(\theta) - \frac{r}{2}, y = y_{center} + \frac{r}{2} \sin(\theta) - \frac{r}{2}, \quad \theta = 91, \dots, 180 \quad (10)$$

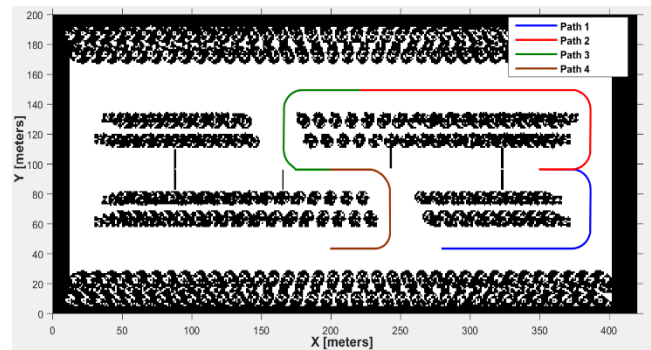
Based on equation (8) to (10), x_{center} and y_{center} denoted the centre of the curve that was

calculated based on the coordinate between two crops in the different turning row.

In order to design the mobile robot path, a crop identification algorithm has been conducted using Mahalanobis distance [31]. After the identification process, the trajectory was formed between the crops by using a probabilistic roadmap [32]. In order for the mobile robot to turn into the next crop row effectively, U turn and π turn were used and compared.



(a) U-Turn trajectory based



(b) π -Turn trajectory based

Figure 4 Trajectory formed based on different turnings

Figure 4 shows the trajectory that has been formed using probabilistic roadmap. Figure 4(a) shows the U-Turn based trajectory and Figure 4(b) shows the π -Turn based trajectory. Based on the figure, the path computed has been divided by 4 different parts. Each part indicates for each checkpoint that need to be achieved by the robot. Based on Figure 5, the measurement has been computed based on the pixel count of the aerial image. A scale of 0.03 has been used to differentiate the measurement between the virtual environment and real environment.

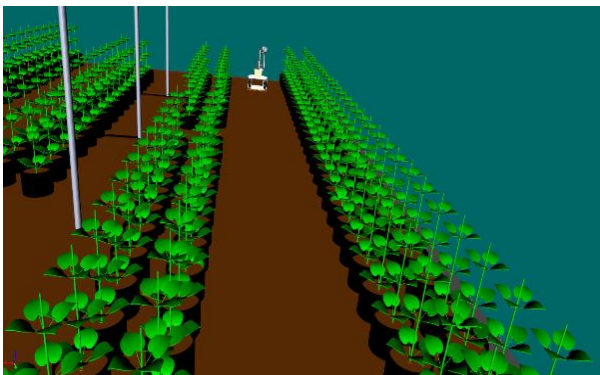
2.5 Simulation Test Setup

In simulation setup, a virtual environment has been designed based on real environment using SolidWorks software. The environment design was then converted into Virtual Reality Modelling Language file (VRML) and simulated in MATLAB. Figure 5(a) shows a real

environment and Figure 5(b) shows the designed virtual environment.



(a)



(b)

Figure 5 Real and simulated environment design

Based on Figure 5, the virtual environment has been design by taking the exact measurements based on the real environment. The similarities of both environment can be shown in Figure 5 in which the simulation process was conducted. The robot motion was simulated by using the unicycle robot model equation shown in equation (1). Simulink3d animation embedded inside a MATLAB software was used as a simulator in this experiment. This simulation environment has been simulated by using MATLAB 2015a software embedded inside an Intel Core i7 2.7 GHz notebook.

2.6 Experimental Test Setup

In experimental setup, a real greenhouse environment has been developed in the University's orchard. It has a dimension of 11.5 m x 5.3 m and has a total of 4 rows and 72 crops. Figure 6 shows the overview of the environment that consists of a real greenhouse setup.



Figure 6 Greenhouse environment setup

For experimental test, a mobile robot system consisted of a modified Magellan Pro mobile robot's base has been used. A Pittman LO-COG DC motor equipped with an optical incremental encoder was used by the mobile robot's base in order to drive the system. Arduino Mega 2560 has been used as a microcontroller in this system. Figure 7 shows the mobile robot's system that was used.



Figure 7 The mobile robot's system that was used

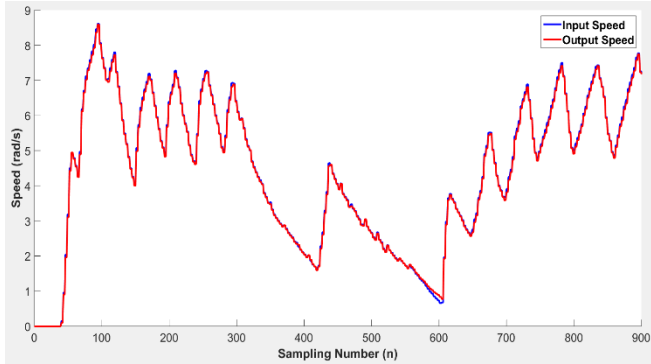
Based on the figure, the mobile robot's system was found to be compatible with the unicycle-like mobile robot model in Figure 2. The model was evaluated by using MATLAB software in order to find the feedback kinematic controller parameters. The parameters were then implemented on a real system. The Experimental test has been conducted by using MATLAB external mode.

3.0 RESULTS AND DISCUSSION

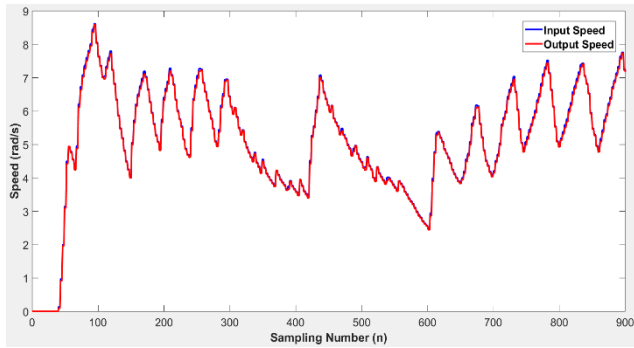
3.1 PI Speed Control Experiment

An experiment was conducted in order to test the performances of a Proportional Integral (PI) speed controller by comparing the input and output speeds from the DC motor for the left and the right wheel motors, respectively. The controller was used in order to control the output signal from the feedback kinematic

controller, so as to avoid a motion overshoot that would occur because of a speed overload. Hence, in order to avoid any unwanted motion of the mobile robot, the speed of the motor needed to be evaluated before the implementation of the motion controller itself. Figure 8(a) shows the velocity tracking result for the right wheel and Figure 8(b) for the left wheel.



(a) Right wheel velocity tracking



(b) Left wheel velocity tracking

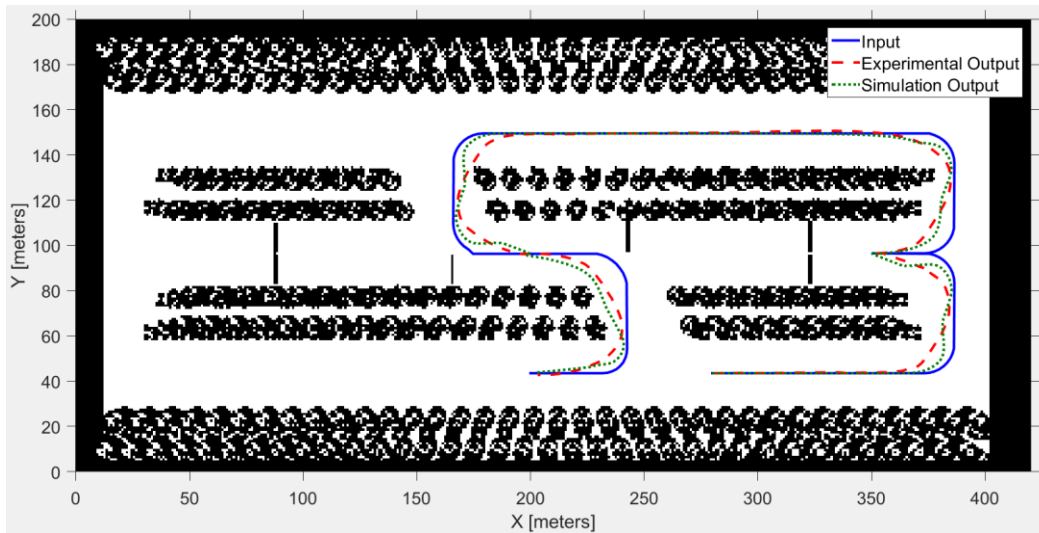
Figure 8 Proportional Integral (PI) speed control result

Based on Figure 8, the input velocity shown by the blue line was generated from the Proportional Integral feedback kinematic controller for the turning motion for path 1. The tuned parameters were identified as follows: For the right wheel, 3 for K_p and 0.001 for K_i and for the left wheel, 0.81 for K_p and 0.001 for K_i . This figure shows that the implemented controller was able to control the speed of the DC motor without using any other complex control system. The controller was also able to generate an output velocity that was almost similar to the input velocity signal. Therefore, it can be concluded that the PI controller was robust and able to control the speed of the DC motor in this experiment.

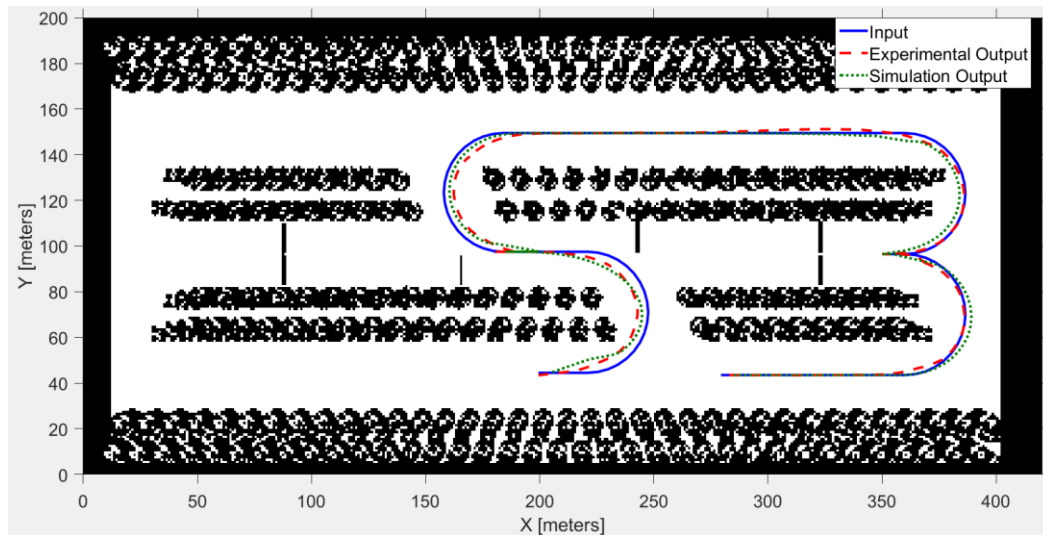
3.2 Trajectories Evaluation

A turning comparison was conducted in order to evaluate the turning performances of the mobile robot during lane changing. It was important for the mobile robot when conducting the greenhouse navigation to optimize its path, by using the appropriate turning scheme for the lane changing. Figure 9(a) shows the trajectory tracking results for π turn and Figure 9(b) for U turn.

Based on Figure 9, the blue line show the input trajectory, the dotted red line shows the experimental output and the dotted green line shows the simulation output. Based on the figure, it can be seen clearly that the π -Turn trajectory shown the worst results in terms of path tracking. In addition, the maximum trajectory tracking error was shown to be occurred during the turning motion. Therefore, it was important to investigate the turning method that offered a better performance for the mobile robot. The details regarding the tracking errors, the distance traveled, and the time taken, is presented in Table 1.



(a) π -Turn trajectory based tracking



(b) U-Turn trajectory based tracking

Figure 9 Trajectory control results

Table 1 Simulation and experimental result

Experimental Mode	Turning Type	Total Mean Square Tracking Error (m)	Distance Traveled (m)	Time Taken (s)
Simulation	π turn	$x=0.1567$, $y=0.0369$	16.5919	299
	U turn	$x=0.1074$, $y=0.0191$	12.3907	266
Experimental	π turn	$x=0.2953$, $y=0.1298$	16.4233	384
	U turn	$x=0.1653$, $y=0.0512$	12.9632	358

The experimental implementation has been conducted to validate the simulation result using a cascade control method. This control method track position and speed at the same time thus enable the robot position to be tracked. The speed control has been conducted using a PI controller by observing the motor speed using encoder.

By comparing the turning type, the U turn offered a better turning performance. It has a lower trajectory error, shorter distance traveled and shorter time taken. In terms of trajectory error, the U turn has a softer turning compared to π turn and thus it is easier to track a softer motion. The U turn type trajectory has a lower distance travelled as the area covered by this turning method is lower than π turn and thus produce a lower time taken. However, the U turn would not be recommended to be implemented if the distance between the headland is wider as the turning radius of U turn will increase drastically. As a result, the distance travelled and time taken will be drastically increase.

In terms π turn based trajectory, it offers a higher distance travelled, controller error and time taken. In term of distance, the π turn covers a larger area than the U turn thus contribute to the additional distance travelled and time taken. In term of controller error, it is mainly because of the small turning radius. It will be hard for a controller to track such a small change in position. In order to reduce the error, the alternative of reducing the robot speed or using more complex controller can be used. However, those alternatives will increase the travel time as the controller complexity is increase and thus making the turning more inefficient.

The turning efficiency was calculated based on the performance of each turning in each objectives using formula:

$$U_{eff} = \left(\frac{\pi_{err} - U_{err}}{\pi_{err}} + \frac{\pi_{dist} - U_{dist}}{\pi_{dist}} + \frac{\pi_{time} - U_{time}}{\pi_{time}} \right) \times 100\% \quad (7)$$

$$\pi_{eff} = \left(\frac{U_{err} - \pi_{err}}{U_{err}} + \frac{U_{dist} - \pi_{dist}}{U_{dist}} + \frac{U_{time} - \pi_{time}}{U_{time}} \right) \times 100\% \quad (8)$$

Where U_{eff} and π_{eff} is the turning efficiency, π_{err} and U_{err} is the total mean square tracking error, π_{dist} and U_{dist} is the distance travelled and π_{time} and U_{time} is the time taken for π turn and U turn respectively. Based on the result, the U turn show a 69.1% better efficiency than π turn in experimental mode. Therefore, it is recommended to be applied in agricultural field as it provides a better optimal path in agriculture.

3.2 System Evaluation

In this paper, simulation and experimental test was conducted to evaluate the performance of robot motion in agricultural environment. For motion control, a feedback kinematics controller has been used and PI based controller was used to control the robot velocity. Both controllers are the simplest and does not need

much computational cost. Despite of using a simple controller, both controller was able to track the robot motion in a great accuracy. However, as the mobile robot was implemented on a soil surface, the accuracy was decreased as the total mean square tracking error increase. It is mainly because of the uneven surface of the soil and thus it is difficult to maintain the mobile robot in a specific position while traversing. Therefore, in order to minimize the tracking error, a more complex controller is needed but it will be expected that a computational cost will be increased.

In terms of the localization, it has been conducted by using dead reckoning method. The encoder readings have been taken and calculated to deduce the robot position as it travels. This experiment was a success, as the simulation mode and the experimental mode resulted in showing a similar pattern. However, in the experimental mode, the completion time was increased drastically. It is mainly because of the delay of the encoder readings as it need to read and calculate the speed and position of the robot. In addition, as the simulation experiment was conducted using a computer and the experimental implementation conducted using an Arduino, the processing speed of both of the microprocessor also contribute to the delay.

The experimental test was conducted to verify the result of the motion control test in simulation. Based on Table 1, the experimental result shows a similar pattern with the simulation result. Therefore, the performance of motion controller has been validated and tested successfully.

4.0 CONCLUSION

In this paper, the implementation of a motion tracking controller and localization for an agricultural mobile robot has been presented. The simulation result for robot motion control has been validated by conducting an experimental test in a greenhouse environment. The implemented motion controller was a success in simulation and experimental, as it was able to control the robot position and guided the robot into the right path despite of having a simple structure. In trajectory assesment, the U turn based trajectory is to be recommended, as it offered a lower characteristic in terms of distance traveled, tracking error and time taken with a 69.1% better efficiency.

However, the experimental delay between the encoder readings and the processing led to the time taken to increase drastically for the path tracking in experimental test. In addition, the odometry based localization also may lead to incremental localization error as the mobile robot does not have any information on the current environment. Therefore, for future improvement, the odometry-based localization data can be improved by combining with the Inertial Measurement Unit (IMU) data and laser data from low cost LIDAR sensor. The combination of those sensors will provide an improved position estimation accuracy for robot localization. Therefore, path tracking will be

improved and application of precision agriculture can be enabled at a very low cost.

Acknowledgement

The authors are grateful to the University of Technology Malaysia, the Ainuddin Wahid scholarship and the Ministry of Higher Education (MOHE), for their partial financial support through their research funds, Vote No. R.J130000.7823.4F759.

References

- [1] Bechar, A., Vigneault, A. 2016. Agricultural Robots for Field Operations: Concept and Components. *Biosystems Engineering*. 149(2016): 94-111.
- [2] Mousazadeh, H. 2013. A Technical Review on Navigation Systems of Agricultural Autonomous off-road Vehicles. *Journal of Terramechanics*. 50(2013): 211-232.
- [3] Belforte, G., R. Deboli, P. Gay, P. Piccarolo, and D. Riccauda imonino. 2006. Robot Design and Testing for Greenhouse Applications. *Biosystems Engineering*. 95(3): 309-321.
- [4] Shalal, N., T. Low, C. McCarthy, and N. Hancock. 2015. Orchard Mapping and Mobile Robot Localization using On-Board Camera and Laser Scanner Data Fusion – Part B: Mapping and localization. *Computers and Electronics in Agriculture*. 119 (2015): 267-278.
- [5] Shivaprasad B. S., Ravishankara, M. N., Shoba, B. N. 2014. Design and Impementation of Seeding and Fertilizing Agriculture Robot. *International Journal of Application or Innovation in Engineering & Management*. 3(6): 251-255.
- [6] Gallakota A. 2011. Agribot- A Multipurpose Agricultural Robot. *2011 Annual IEEE India Conference*. Hyderabad, India. 16-18 December 2011. 1-4.
- [7] Zaidner, G. and A. Shapiro. 2016. A Novel Data Fusion Algorithm for Low-Cost Localization and Navigation of Autonomous Vineyard Sprayer Robots: *Biosystems Engineering (Advances in Robotic Agriculture for Crops)*. 146 (2016): 133-148.
- [8] Yi-Chich, C., S. Chen, and L. Jia-Feng. 2013. Study of an Autonomous Fruit Picking Robot System in Greenhouses. *Engineering in Agriculture, Environment and Food*. 6(3): 92-98.
- [9] Bayar, G., M. Bergerman, A. Bugra Koku, and E. Ilhan Konukseven. 2015. Localization and Control of an Autonomous Orchard Vehicles. *Computers and Electronics in Agriculture*. 115 (2015): 118-128.
- [10] Bechar, A., Vigneault, A. 2017. Agricultural Robots for Field Operations. Part 2: Operations and Systems. *Biosystems Engineering*. 153(2017): 110-128.
- [11] Bac, C. W., Hemming, J., van henten, E., J. 2013. Robust Pixel-based Classification of Obstacles for Robotic Harvesting of Sweet-Pepper. *Computers and Electronics in Agriculture*. 96(2013): 148-162.
- [12] Sivaraman, B. and Burks , T. F. 2006. Geometric Performance Indices for Analysis and Synthesis of Manipulators for Robotics Harvesting. *Transactions of the ASABE*. 49(5): 1589-1597.
- [13] Guo, L.S., Zhang Q., Han S. 2002. Position Estimate of Off-Road Vehicles using a Low Cost GPS and IMU. *ASAE Annual International meeting/CIGR XVth World Congress*. Chicago, Illinois, USA. 28-31 July 2002. 1-8.
- [14] Bakker, T., van Asselt, K., Bontsema, J., Muller, J., van Straten, G. 2011. Autonomous Navigation using a Robot Platform in a Sugar Beet Field. *Biosystems Engineering*. 109(2011): 357-368.
- [15] Earl, R., Thomas, G., Blackmore, B.S. 2000. The Potential Role of GIS in Autonomous Field Operations. *Computers and Electronics in Agriculture*. 25(2000): 107-120.

- [16] Xue, J., Zhang, L., Griff, TE. 2012. Variable Field-of-View Machine Vision based Row Guidance of an Agricultural Robot. *Computers and Electronics in Agriculture*. 84(2012): 85-91.
- [17] Weiss, U., Biber, P. 2011. Plant Detection and Mapping for Agricultural Robots using a 3D LIDAR Sensor. *Robot Autonomous Systems*. 59(2011): 265-273.
- [18] Torii, T. 2000. Research in Autonomous Agriculture Vehicles in Japan. *Computers and Electronics in Agriculture*. 25(2000): 133-153.
- [19] Yarza, A., V. Santibanez, and J. Moreno-Vanezuela. 2013. An Adaptive Output Feedback Motion Tracking Controller for Robot Manipulators: Uniform Global Asymptotic Stability and Experimentation. *International Journal of Applied Mathematics and Computer Sciences*. 23 (3): 599-611.
- [20] Santibanez, V., R. Kelly, and M. Llama. 2002. Asymptotic Stable Tracking for Robot Manipulators via Sectorial Fuzzy Control. *15th World Congress of the International Federation of Automatic Control (IFAC)*. Barcelona, Spain. 21-26 July 2002. 359-364.
- [21] Felipe Martins, N., C. Wanderly Caleste, R. Carelli, M. Sarcinelli-Filho, and F. Teodiano Bastos-Filho. 2008. An Adaptive Dynamic Controller for Autonomous Mobile Robot Trajectory Tracking. *Control Engineering Practice*. 16(2008): 1354-1363.
- [22] Schnelle, F. and P. Eberhard. 2015. Constraint Mapping in a Feedback Linearization/MPC Scheme for Trajectory Tracking of Under Actuated Multibody Systems. *5th IFAC Conference on Nonlinear Model Predictive Control (NMPC'15)*. Seville, Spain. 17-20 September 2015. 446-451.
- [23] De La Cruz, C. and R. Carelli. 2006. Dynamic Modeling and Centralized Formation Control of Mobile Robot. *32nd Annual Conference on IEEE Industrial Electronics (IECON)*. Paris, France. 7-10 November 2006. 3880-3885.
- [24] Min Hyuc, K., Beom-Sahng, R., Kyoung Chul, K., Suprem, A, Nitaigour Mahalik, P. 2015. Autonomous Greenhouse Mobile Robot Driving Strategies From System Integration Perspective: Review and Application. *IEEE/ASME Transaction on Mechatronics*. 20(4): 1705-1716.
- [25] Urrea, C. and Munoz, J. 2015. Path Tracking of Mobile Robot in Crops. *Journal of Intelligent & Robotics System*. 80(2): 193-205.
- [26] Changlong, Y., J. Chen, M. Chen, and L. Liu. 2015. A Control Approach of an Omnidirectional Mobile Robot with Differential Wheels. *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*. Beijing, China. 2-5 August 2015. 1211-1216.
- [27] Kumar Malu, S. and J. Majumdar. 2014. Kinematics, Localization and Control of Differential Drive Mobile Robot. *Global Journal of Researches in Engineering: Robotics & Nano-Tech*. 14(1): 1-7.
- [28] Hameed, I.A, A La Cour-Harbo, and K. D. Hansen. 2014. Task and Motion Planning for Selective Weed Control using a Team of Autonomous Vehicles. *13th International Conference on Control, Automation, Robotics & Vision*. Marina Bay, Singapore. 10-12 December 2014. 1-5.
- [29] Jin J. Optimal Field Coverage Path Planning on 2D and 3D Surfaces. 2011. *Phd Thesis*, Iowa State University, United States.
- [30] Jesus, C., M, Jose Maria, B., G, Andujar, D., Roberio, A. 2016. Route Planning for Agricultural Tasks: A General Approach for Fleets of Autonomous Vehicles in Site-Specific Herbicide Applications. *Computers and Electronics in Agriculture*. 127(1): 204-220.
- [31] Eddaurich S., A. Hammouch, T. Meriem, R. Touahni, and A. Shibi. 2014. Unsupervised Neutral-Morphological Color Image Segmentation using Mahalanobis Criteria of Resemblance. *Multimedia Computing and Systems (ICMCS) International Conference*. Marrakesh, Morocco. 10-12 May 2014. 314-320.
- [32] Claude Overmars, L., E. Kavradi Lydia, P. Svetska, and M. H. Jean. 1996. Probabilistic Roadmap for Path Planning in High-Dimensional Configuration Spaces. *IEEE Transactions on Robotic and Automation*. 12(1): 566-579.