Advances in Animal Biosciences: Precision Agriculture (ECPA) 2017, (2017), **8:2**, pp 854–859 © The Animal Consortium 2017
doi:10.1017/S2040470017001078

advances in animal biosciences

Virtual Reality based Mobile Robot Navigation in Greenhouse Environment

M. Saiful Azimi†, Z. A. Shukri and M. Zaharuddin

Department of Control and Mechatronics, Faculty of Electrical Engineering, University of Technology Malaysia Skudai, 81310 Skudai, Johor, Malaysia

The difficulties of transporting heavy mobile robots limit robotic experiments in agriculture. Virtual reality however, offers an alternative to conduct experiments in agriculture. This paper presents an application of virtual reality in a robot navigational experiment using SolidWorks and simulated into MATLAB. Trajectories were initiated using Probabilistic Roadmap and compared based on travel time, distance and tracking error, and the efficiency was calculated. The simulation results showed that the proposed method was able to conduct the navigational experiment inside the virtual environment. U-turn trajectory was chosen as the best trajectory for crop inspection with 82.7% efficiency.

Keywords: greenhouse, mobile robot, navigation, virtual reality.

Introduction

Virtual reality is a combination of technologies offering a natural interaction with a simulated world created by a computer. The application of virtual reality has been implemented widely in mobile robot simulation (Dias *et al*., 2015; Mostefa *et al*., 2015). A robot system can be tested without having any hardware involvement by using a simulator and virtual robot models. Therefore, the development and the operation of a robot system can be more efficient, as proper tests can be conducted before there is any hardware implementation.

A virtual reality implementation was used in order to analyse the motion of a 6R robot manipulator (Li *et al*., 2016). It utilised software called EON Studio for the virtual reality and MATLAB was used for the inverse kinematics. This approach is not to be recommended as it uses different software and it is time consuming when learning different software. Nakaoka (2012) developed a software framework called Choreonoid that allowed a user to efficiently implement and integrate various GUI tools in order to handle virtual robot motions. It was developed based upon a C++ programming language and it could be applied by using various platforms such as Windows, Linux and Mac OS.

The movements of mobile robots called P2AT and Talon were modelled for search and rescue missions in a virtual environment (Chuan *et al*., 2007). They used USARSim, a high fidelity mobile robot teleoperation software that was employed together with the Unreal Game Engine. The implementation showed a lag in signal speed and it was also time consuming. (Lutovac *et al*., 2015) developed the use of a reconfigurable virtual robot laboratory for the programming, the simulation, and the control of a multi-robot operation. It was implemented in a Linux platform by using C++ and the OpenGL library was used for the 3D visualization.

In a virtual reality design using MATLAB software, (Menad *et al*., 2013) proposed a kinematic modelling of 18DOF humanoid robots in a virtual environment. (Le *et al*., 2013) developed an easy operating system for a hydraulic excavator in a virtual reality. The system was used and designed in order to ensure the intuitive monitoring for an excavating operator. Despite the implementation of virtual reality that has already been conducted when employing MATLAB, there are not any papers in the literature that show a detailed implementation of virtual reality when using the MATLAB software.

For agriculture, the application of a virtual environment can be a good alternative for a researcher in order to conduct experiments. A great deal of time and costs can be reduced by using this kind of an implementation in mobile robot development. This paper presents an application of a virtual greenhouse environmental design by using MATLAB software in a robot navigational experiment. The environment was first of all designed by using SolidWorks software and was then simulated in MATLAB. The navigational experiment was conducted by using the Probabilistic Roadmap Planner (PRM). The best path was chosen for crop inspection operations inside a greenhouse.

Materials and Methods

Navigation experiments for mobile robots in virtual environment can be conducted by combining a path planner and motion controller. In order to generate a path, a crop

† E-mail: shukri@fke.utm.my

https://doi.org/10.1017/S2040470017001078 Published online by Cambridge University Press

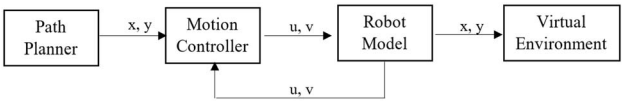


Figure 1 Navigation experiment process in virtual environment.

identification process need to be conducted to detect the crop position in the environment. Figure 1 shows the overall process needed to conduct the navigation experiment for mobile robot.

Based upon Figure 1, x and y indicate the mobile robot's position and u and v show the linear and angular velocity, respectively. The path planner is able to generate a trajectory for the planning of the robot's motion. The motion controller is able control the robot's motion and the motion can then be simulated in a virtual environment.

Path Planner

In order to generate a path for the mobile robot's navigation, the path planner needs to initialise a specific position as the goal point. A crop identification process is needed in order to detect the crop positions in the environment. The crop positions will then act as an input for the path planner in order to plan the path inside the environment. The crop identification system would then be executed based upon a colour filter when using Mahalanobis distance in order to detect the crops (Eddarouich *et al.*, 2014).

There are several path planners available for path planning, such as the Probabilistic Roadmap Planner (Kavraki *et al.*, 1996), the Artificial Potential Field Approach (Bence *et al.*, 2016) and the Voronoi Roadmap Method (Tian *et al.*, 2014). Among these planners, the Probabilistic Roadmap Planner was chosen for our work because of its simplicity. It also has a high success rate when solving motion planning for a mobile robot with many degrees of freedom.

Robot Model

A differential drive mobile robot was chosen, as it is the most common type of mobile robot that is used in various applications, such as in surveillance and in autonomous wheelchair procedures. It consists of two main wheels and one castor wheel. The model is shown in Figure 2 and it was introduced by Felipe *et al.* (2008). The robot kinematics model is given by:

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

Based upon Equation 1, this procedure was derived from the transformational matrix between the robot's frame and the inertial frame. The model equation was related to a non-holonomic robot, as the differential equation was not integrable into the final position. Therefore, the distance that each wheel travelled was not sufficient enough for the final position calculation.

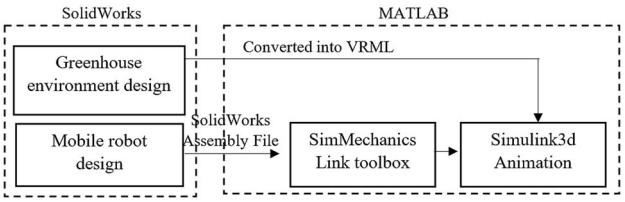


Figure 2 Virtual environment design process.

Motion Controller

The controller equation was derived from the inverse kinematics model that was obtained when deriving the kinematics equation of the model. The inverse kinematics model 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, by applying the kinematics control law, the kinematics based controller for the trajectory tracking was designed. Thus, the kinematics controller equation 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 $l_x \in \mathscr{R}$ and $l_y \in \mathscr{R}$ were the saturation constants.

Virtual Environment

In virtual environment design, the environment and robot model design has different approach in MATLAB implementation. The environment design process starts with designing the environment using SolidWorks and then converted into VRML file. The VRML file can be further edited using V-Realm application embedded in MATLAB. For mobile robot design, the process starts with SolidWorks design and then converted into MATLAB Simulink model using SimMechanics plugin installed inside SolidWorks application. The overall process in designing the virtual environment is shown in Figure 2.

Based upon Figure 2, the virtual environmental process was divided into different software environments: SolidWorks and MATLAB. The environment and the mobile robot were properly designed with real parameters. They were used later in the model's initialisation. For the environmental editing, this was modified in terms of rotation and translation and it needed to be properly inserted into the Simulink 3D Animation. Figure 3 shows the V-Realm software interface and the environmental translation. The rotation was modified in the Transform Group.

https://doi.org/10.1017/S2040470017001078 Published online by Cambridge University Press

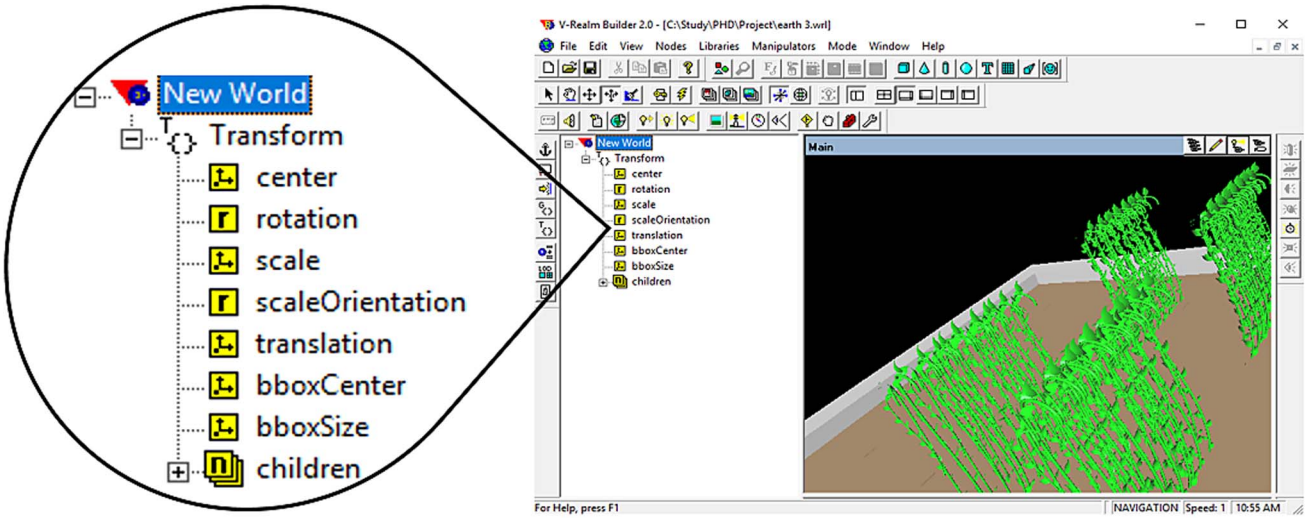


Figure 3 V-Realm software interface.

When designing the mobile robot model, the 'mate features' in the SolidWorks software for the assembly design needed to be identified and needed to be initialised carefully, as the Simulink conversion was computed based upon the joints that were initialised in the SolidWorks software. The joint directions were defined as a motion of follower's frame relative to the base frame. Therefore, it was important to verify the joint types in the generated model, in order to avoid any unwanted motion during the simulation. After the environment and the robot model were combined, the overall environment was able to be edited using the 3D World Editor that is embedded inside the MATLAB software.

Results and Discussion

In this section, the results of designing the virtual environment and the robot's navigational experiment are presented. The unicycle-like mobile robot model and the greenhouse environment with 4 different crop rows were designed by using the SolidWorks software. The navigational experiments were conducted in order to choose the best trajectories for the greenhouse navigation. The turning efficiency was calculated based upon the performances of each turn for each objective using the following formula:

$$U_{eff} = \left(\frac{\pi_{err} - U_{err}}{\pi_{err}} + \frac{\pi_{dist} - U_{dist}}{\pi_{dist}} + \frac{\pi_{time} - U_{time}}{\pi_{time}}\right) \times 100\,\% \tag{4}$$

$$\pi_{eff} = \left(\frac{U_{err} - \pi_{err}}{U_{err}} + \frac{U_{dist} - \pi_{dist}}{U_{dist}} + \frac{U_{time} - \pi_{time}}{U_{time}}\right) \times 100\,\% \tag{5}$$

This is where U_{eff} and π_{eff} were the turning efficiencies, π_{err} and U_{err} were the total mean square tracking errors, π_{dist} and U_{dist} were the distances travelled and π_{time} and U_{time} were the time taken for π turn and U turn respectively. The positive values of turning efficiency shows the better trajectory performance in all objectives, respectively.

Mobile robot model

The unicycle-like mobile robot model was designed based on the real mobile robot system. It consists of two wheels and one castor wheel. It was equipped with Rhino XR-4 robot arm with 5 degrees of freedom. The model was design using the exact dimension as the real model to avoid model conflict and controller parameter problems in the hardware implementation. Figure 4 shows the mobile robot model that has been designed in SolidWorks and MATLAB respectively.

Based upon Figure 4, the mobile robot that was designed in SolidWorks was compatible for a MATLAB implementation, as the SimMechanics™ Plugin was able to create a model that was based upon that design. The motion was then validated by simulating the SimMechanics™ model. Figure 5 shows the Simulink block that was generated by SimMechanics™ Plugin.

Based upon Figure 5, the initialisation part of the generated model consisted of the world coordinates and they were then transformed into the robot's coordinate system. The first part of the rigid body was a base for the mobile robot. The base consisted of three different revolute joints: the left wheel, the right wheel and the robot arm base. The robot arm base had a revolute joint, as it rotated about the z-axis. The arm revolute joints then continued until they reached the gripper body part. The joints were then able to rotate by using either the velocity or the position of the rotation. The wheels had different revolute joints, as the wheel rotations did not link with each other.

Virtual greenhouse environment

The virtual greenhouse environment was designed based upon a real greenhouse environment with dimensions of approximately 4.3 m x 4.5 m. Each row had a 3 m length and

https://doi.org/10.1017/S2040470017001078 Published online by Cambridge University Press

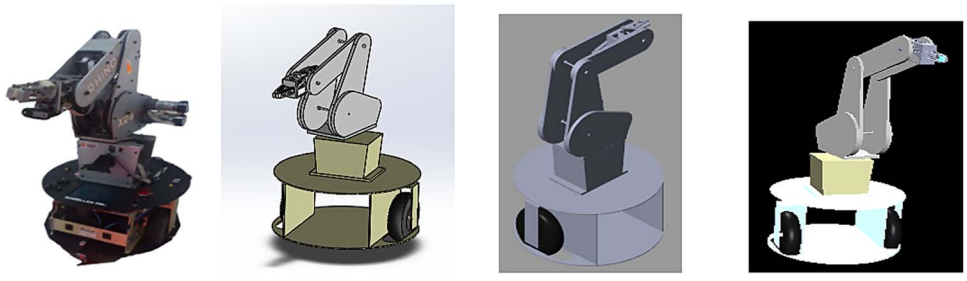


Figure 4 Designed mobile robot model.

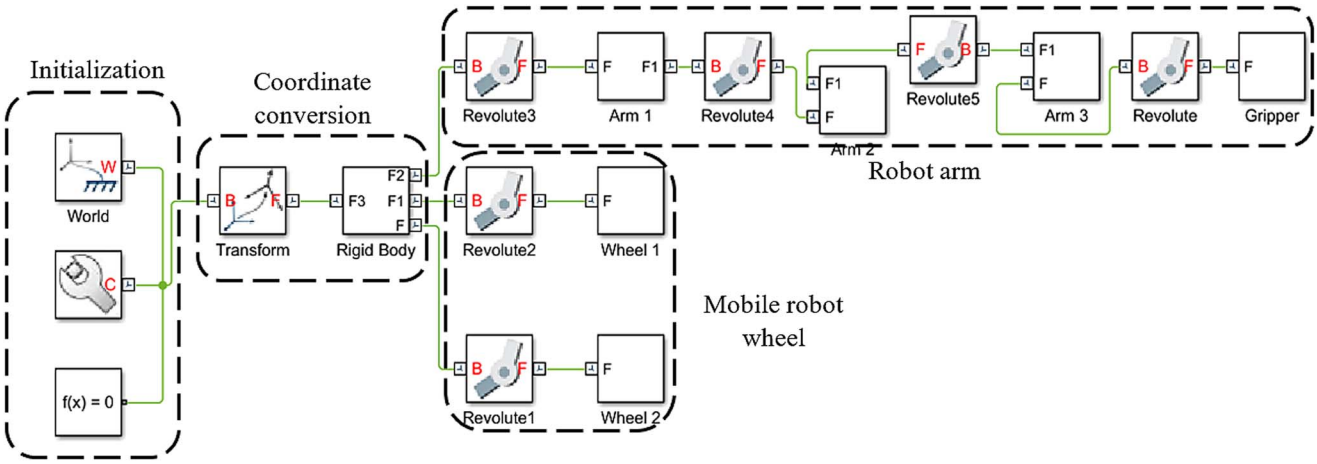


Figure 5 Generated mobile robot Simulink model.

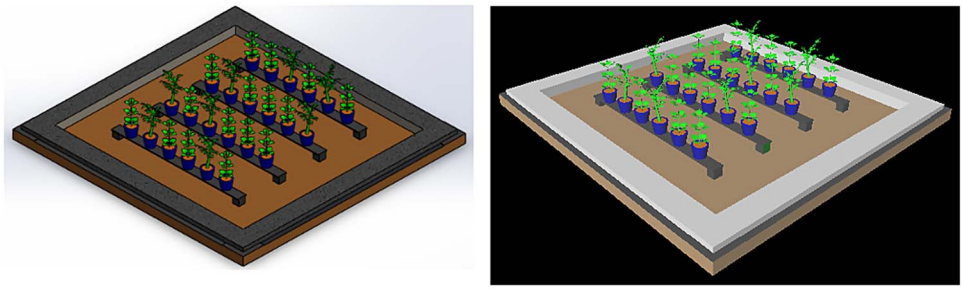


Figure 6 Designed virtual environment.

the row width varied according to the crop size. The designed environment consisted of 2 different crop types and they were arranged randomly in 4 different rows. Figure 6 shows the designed virtual environment.

Based upon Figure 6, the designed SolidWorks model was then converted into a VRML file. In order to verify the conversion when comparing the VRML file with the SolidWorks model, V-Realm software was used. The environmental VRML file and the SimMechanics™ mobile robot model were then combined inside the Simulink 3D environment. Figure 7 shows the Simulink 3D environment using different virtual camera views and the crop tagging results can be seen.

Figure 7 shows the results from a Simulink 3D animation view. In Simulink 3D, the camera views can be designed according to different positions with dissimilar orientations. For the navigational experiments, two camera views were used: an overhead view and a first-person view. Based upon the Figure, the crop tagging results show that the crops were identified successfully in the environment and they were named based upon the rows and the columns. The coordinates of the tagging processes were then transferred into the motion planner inputs in order to plan the motions according to the crop coordinate sequences. Figure 8 shows the motions that were computed by the Probabilistic Roadmap Planner.

Based upon Figure 8, the path computed by the Probabilistic Roadmap Planner is shown by the dashed line and the simulated path is shown by the dotted line. The simulated path

https://doi.org/10.1017/S2040470017001078 Published online by Cambridge University Press

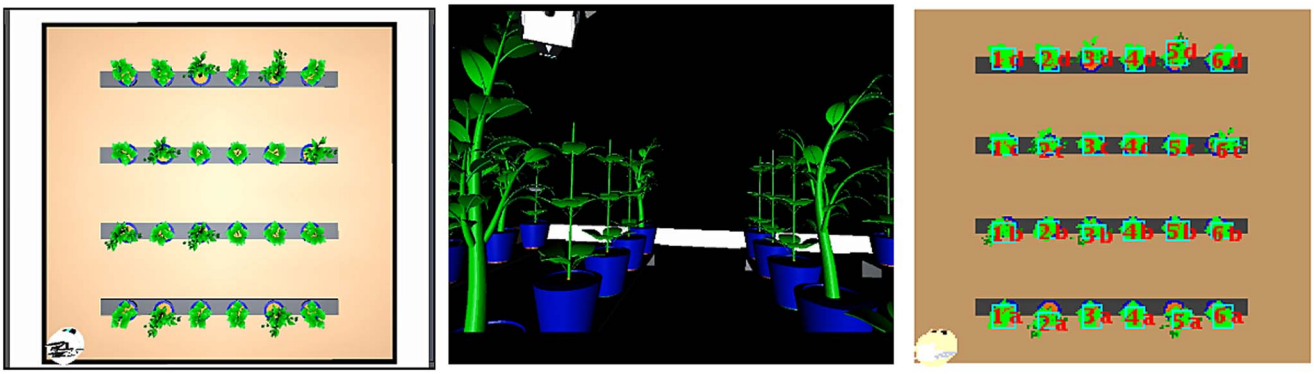


Figure 7 Simulink3d virtual environment result.

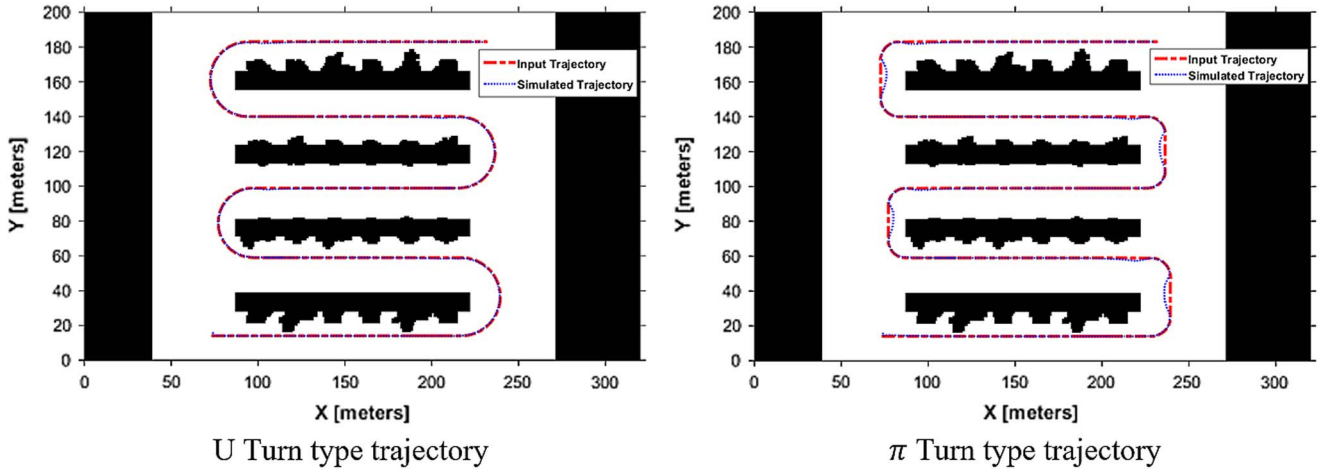


Figure 8 Trajectory experiment result.

Table 1 *Robot navigation simulation result.*

Trajectory Type	Total Mean Square Tracking Error (m)	Distance Traveled (m)	Travel Time (s)	Efficiency (%)
π turn	x = 0.0806, y = 0.0048	11.4136	126	17.3
U turn	x = 0.0789, y = 0.0008	11.3145	130	82.7

was computed by the mobile robot's motion by using the feedback kinematic controller. The control parameters used were 2 for k_x and k_y and 0.5 for l_x and l_y, respectively. Figure 8 also proved that the controller was able to track the motion that was computed with a high degree of accuracy.

Table 1 shows the comparison that was computed between the U turn based trajectory and the π turn based trajectory. The comparisons were conducted based upon the travel time, the distance travelled and the controller mean square errors. The efficiency was calculated by using Equation 4 and 5 and by combining all of the criteria as stated. Based upon the table, the U turn based trajectory offered a better turning performance with 87.2% efficiency. It had a shorter distance that was travelled and lower trajectory errors. However, the time taken for the U turn was higher than it was for the π turn. The additional time was a result of the mobile robot's speed depletion during the turning. As the U turn had a higher turning radius than did the π turn, the turning travel time was, therefore, higher. In addition, the controller also needed time to track the positions during the turning, as both of the x and y coordinates changed. However, the U turn would not be recommended to be implemented, if the time taken was the main criteria to be evaluated, as the π turn offered less time for the robot's navigation.

When considering the π turn based trajectory assessment, it offered a higher distance travelled and more prominent controller errors. In terms of the distance travelled, the π turn covered a larger area than did the U turn, and thus, this contributed to the additional distance that was travelled.

https://doi.org/10.1017/S2040470017001078 Published online by Cambridge University Press

In terms of the controller errors, this was mainly because of the small turning radius. It was hard for the controller to track such a small change in positions. In order to reduce the errors, the alternatives of reducing the robot's speed, or using a more complex controller, could be implemented and used. However, these alternatives would increase the travelling time, as the controller implementation would become more complex, and thus, it would make the turning more inefficient.

In terms of an overall comparison, the U turn based trajectory is recommended to be applied in an agricultural field with similar and minimum headland distance, as it provided for a better optimal path in agriculture. However, if the distance between headlands is high, π turn is recommended as it covers lower area and it was proved by Bochtis *et al*. (2009) where the π turn is executed when the turning radius is twice the machine minimum turning radius. Therefore, both turnings can be executed depending on the headlands distance and for minimum headland distance, U turn was recommended.

By using virtual reality as the programming environment, researchers especially in agriculture can visualize and validate their experiment findings and methodology with a clearer perspective. In this paper, the experiment was executed successfully as the robot motion can be observed clearly in the virtual environment. The controller parameter can also be tuned without using any hardware. The designed virtual environment that visualize the real environment will give benefits to the researcher to conduct experiment by using the same environment but executed in a very comfortable way.

Conclusions

In summary, the greenhouse virtual environment that employed 2 crop types was designed and simulated by MATLAB software. The mobile robot's navigational experiments were conducted by using the Probabilistic Roadmap Planner and the trajectory performances were evaluated. The simulation results have shown that the U turn based trajectory is to be recommended, as it offered a lower distance travelled with lower tracking errors and with 82.7% efficiency. The method of using a virtual environment as an alternative method of conducting navigational experiments was a success, as the mobile robot's motion was able to be simulated in the environment. For future works, a hardware implementation will be conducted and the trajectory performances will be evaluated and compared, in order to validate the best of operations.

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

Bence K, Geza S, Ferenc T, Mauricio B and Peter K 2016. A novel potential field method for path planning of mobile robots by adapting animal motion attributes. Robotics and Automation Systems 82, 24–34.

Bochtis DD and Sorensen CG 2009. The vehicle routing problem in field logistics part I. Biosystems Engineering 104, 447–457.

Chuan TK, Shin CN, Foo D, Chao WY and Wenhao Y 2007. Modelling robot movement in a virtual environment. In: 10th IFAC, IFIP, IFORS, IEA Symposium on Analysis, Design, and Evaluation of Human-Machine Systems, Elsevier, pp. 177-182.

Kavraki EL, Svetska P, Latombe J-C and Overmars HM 1996. Probabilistic roadmap for path planning in high dimensional configuration spaces. IEEE Transactions on Robotic and Automation 12, 566–579.

Dias T, Miraldo P, Goncalves N and U.Lima P 2015. Augmented reality on robot navigation using non-central catadioptric cameras. In: 2015 IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS), IEEE, pp. 4999-5004.

Eddarouich S, Hammouch A, Meriem T, Touahni R and Sbihi A 2014. Unsupervised neural-morphological color image segmentation using mahalanobis criteria of resemblance. In: Multimedia Computing and Systems (ICMCS) International Conference, IEEE, pp. 314-320.

Felipe Martins N, Wanderly Caleste C, Carelli M, Sarcinelli-Filho M and Teodiano Bastos-Filho F 2008. An adaptive dynamic controller for autonomous mobile robot trajectory tracking. Control Engineering Practice 16, 1354–1363.

Le QH, Park K-T and Yang S-Y 2013. Study on development of easy operating system for field robot in virtual reality environment. In: 2013 3rd International Conference on Control, Automation and Systems (ICCAS 2013), ICROS, pp. 1152-1155.

Li WJ, Song ZH, Zhu ZX and Mao ER 2016. Analysis and simulation of a 6R robot in virtual reality. In: 5th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2016, Elsevier, pp. 426-430.

Lutovac MM, Dimic Z, Mitrovic S and Stepanovic A 2015. Reconfigurable multi-robot virtual environment. In: 23rd Telecommunication forum TELFOR 2015, IEEE, pp. 954-957.

Menad M, Derouiche Z, Ahmed Foitih Z and Nouibat W 2013. Kinematic modelling of a humanoid robot with 18 DOF in a virtual environment. In: 2013 3rd International Conference on Innovative Computing Technology (INTECH), IEEE, pp. 424-429.

Mostefa M, Kaddour El Boudadi L, Loukil A, Mohamed K and Amine D 2015. Design of mobile robot teleoperation system based on virtual reality. In: 2015 3rd International Conference on Control, Engineering & Information Technology (CEIT), IEEE, pp. 1-6.

Nakaoka S 2012. Choreonoid: Extensible virtual robot environment built on an integrated GUI framework. In: 2012 IEEE/SICE International Symposium on System Integration (SII), IEEE, pp. 79-85.

Tian S, Cui X and Gong Y 2014. A general vector-based algorithm to generate weighted voronoi diagrams based on ArcGIS engine. International Journal of Computer and Information Technology 3, 737–742.

https://doi.org/10.1017/S2040470017001078 Published online by Cambridge University Press