Proceedings of 2017 4th International Conference on
Control, Decision and Information Technologies (CoDIT'17) / April 5-7, 2017, Barcelona, Spain

Tracking Control For Wheeled Mobile Robot Using RGBD Sensor

Raouf Fareh, Tamer Rabie and Mohammed Baziyad

***Abstract*—**This paper introduces velocity tracking control for differential wheeled mobile robot using vision information provided by a depth camera. This tracking control strategy is done into three phases. First, a depth camera mounted on the ceiling is used to collect texture and 3D shape information of the scene. Second, the piecewise 3D models generated in the previous phase and the Probabilistic Roadmap path planning algorithm (PRM) are used as an input for a robotic path planner to develop a free-obstacle path. Third, a kinematic controller is developed to track the desired path given from the path planner. Lyapunov theory is used to prove the stability of the closed loop system. Experimental results on a Pioneer P3-DX differential wheeled mobile robot and an Xtion PRO depth camera are illustrated at the end of the paper to prove the effectiveness and efficiency of the proposed tracking system.

***Keywords*—** Mobile robot, Path planning, Vision, kinematic control, Probabilistic Roadmap, obstacle avoidance, Lyapunov, trajectory tracking.

I. Introduction

CONTROL systems for mobile robots have been a huge research area of interest in recent years, especially the differential wheeled mobile robots. Nonlinearity modeling and the wide usage of differential mobile robots, have encouraged researchers to this research field. However, the traditional linear control algorithms are unable to solve the non-linearity property of real systems. This fact had led researchers around the world to examine other techniques and strategies. It has been turned out that controlling such systems is not a simple problem for many reasons [1-5].

To overcome nonlinear problems, the authors in [6] propose a predictive-fuzzy controller for trajectory tracking of autonomous mobile robots. A fuzzy controller is suitable for cases when the dynamic model is not accurate. On the other hand, a predictive controller can solve the problem of huge delays of sensor signals. To gain both advantages, the proposed controller is a system that combines fuzzy and predictive controller in one platform. A nonlinear differential-algebraic feedback control system has been presented in [6]. This control system was developed on a constrained robot system. The experimental results prove that the tracking error approaches to zero when applying different scenarios. Path tracking controllers for differential two-wheeled mobile robot was developed in [7] using its kinematics and dynamics model that can handle high speed and acceleration. In [9], the authors proposed a sliding mode control design to stabilize the mobile robot to a pre-defined trajectory. Many contributions were made using an adaptive control scheme, [8-10] are examples of such work. Another work in [14], a vision-based control system is developed to solve real-time experiences in [14].

The stability of several control schemes proposed for mobile robot were proved using Lyapunov approach. In [11], an exponential stabilization of a mobile robot via discontinuous Lyapunov –based Controller was introduced. In [16], the authors suggest to use a Lyapunov function to pre-define specific paths to move the robot to a particular pose. The idea is to take advantage of the invariance characteristic of sliding mode control. A number of quadratic Lyapunov functions were used to prove the stability of the system. For three wheeled unicycle-like robots, [12] designed his control using Lyapunov approach. The author proved that stability can be achieved by smooth feedback control law if the system state variables were elected carefully. Since that, the convergence of tracking errors and the time response of a Lyapunov controller have become critical characteristics that researchers have been working to improve.

The vision sensor is one of the versatile devices for robots to obtain the information about target objects and the environment. The visual feedback control technique contributes to achieve the object recognition, the object tracking, and so on [13]. When the target object is assumed to be fixed in the workspace, [14] proposes a non-delayed visual tracking. In [15], the target moves repetitively and the trajectory is estimated in advance.

This paper takes advantage of the vision system to design a tracking control strategy for the mobile robots. This control strategy is divided into three levels to achieve a smooth tracking movement while the robot moving forward on a predefined trajectory. In the first step, a depth camera mounted on the ceiling is used to generate a vision information about the environment. In the second step, a free-obstacle path is generated using a depth camera and the Probabilistic Roadmap path planning algorithm (PRM). Next, a kinematic controller is used to generate a velocity control law in order to adjust the right and left wheel velocities. This control law uses the desired and the real position/orientation of the platform to generate a desired velocity that will make the robot to follow the path

Dr. Fareh Raouf is with the University of Sharjah. Department of Electrical and Computer Eng., University of Sharjah, P.O.Box 27272, Sharjah, UAE. rfareh@sharjah.ac.ae.

Mohammed Baziyad is with the University of Sharjah. Research Institute of Sciences & Engineering, University of Sharjah P.O.Box 27272, Sharjah, UAE. mbaziyad@sharjah.ac.ae

Tamer Rabie is with the University of Sharjah. Department of Electrical and Computer Eng., University of Sharjah, P.O.Box 27272, Sharjah, UAE. trabie@sharjah.ac.ae

978-1-5090-6465-6/17/$31.00 ©2017 IEEE

calculated by the PRM algorithm. Experimental results show the effectiveness of the proposed control strategy for controlling the mobile robot.

The rest of this paper is organized as follows. In Section 2, the system description is presented. Section 3 describes the vision and the image processing algorithms. Section 4 illustrates the path planning technique. Section 5 presents the modeling and the control design. The experimental results are discussed in Section 6, and the conclusion is presented in Section 7.

II. SYSTEM DESCRIPTION

The system used in this paper is a differential wheeled mobile robot system. Two driving wheels are fixed in the front in the same axis while a small free castor rotational wheel is mounted at the back of the robot as shown in figure 1.

The generalized coordinate $q = [x, y, \emptyset]^T$ represents the position and orientation vector of the robot, and $q_d = [x_d, y_d, \emptyset_d]^T$ denotes the desired trajectory obtained from the path planning algorithm.

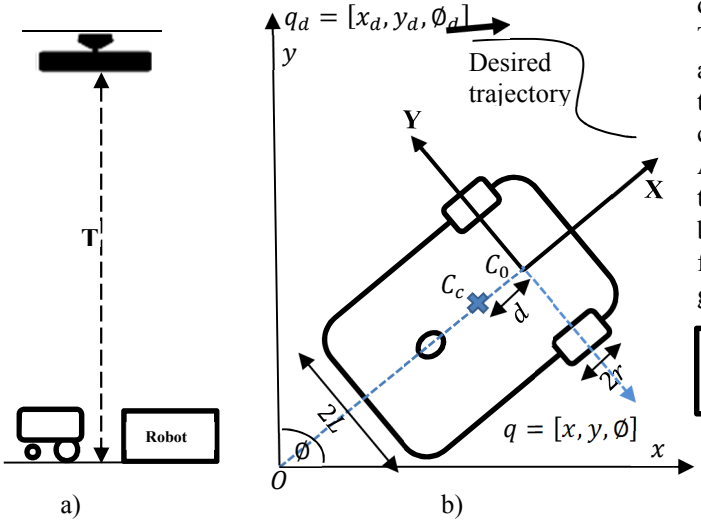

Fig. 1 a) A general description of the system. b) The Mobile robot as seen from the depth camera.

The goal is to generate a desired trajectory from the vision information and to develop a kinematic control that tracks this desired path. Figure 2 illustrates the block diagram of the control strategy. There are three main levels: vision system, path planner, and kinematic controller. First, a Kinect sensor is placed to collect data from the scene. Second, the path planning phase is done, where the obstacles are triggered, and a free-obstacle path is generated using Probabilistic Roadmap algorithm (PRM).

As shown in figure 2, the path generated from the path planning part is an input to the kinematic controller. Finally, the kinematic controller uses this path along with the readings of the current position and orientation of the robot to control the robot to follow the desired path. This is done by adjusting the forward and rotational velocity of the robot.

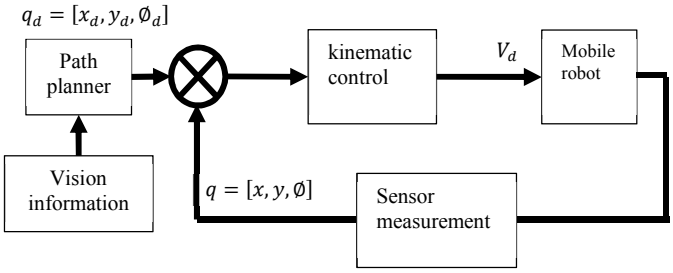

Fig. 2 control design

III. VISION AND IMAGE PROCESSING ALGORITHMS

The first essential part of the proposed system is generating a free-obstacle path from a starting point to the target. To be able to scan the area for obstacles, depth cameras are used. Depth cameras are those cameras that are able to estimate the distance of each pixel in the scene from the camera. Usually, a number of depth cameras are used in a collaboration mode to model a room or an area. To simplify this process, only one depth camera is used in our work, and mounted on the ceiling. The depth image of the area is taken and then processed using a computer. Since the distance from the depth camera (celling) to the floor is known, values that are less than that value is counted to be an obstacle. Figure 4 illustrates this technique. After that, a binary image is created where the black area are the obstacles that the robot should avoid. This binary image will be the input to the PRM planner to find a free-obstacle path from point A to point B. Figure 3 shows the path planning general procedure.

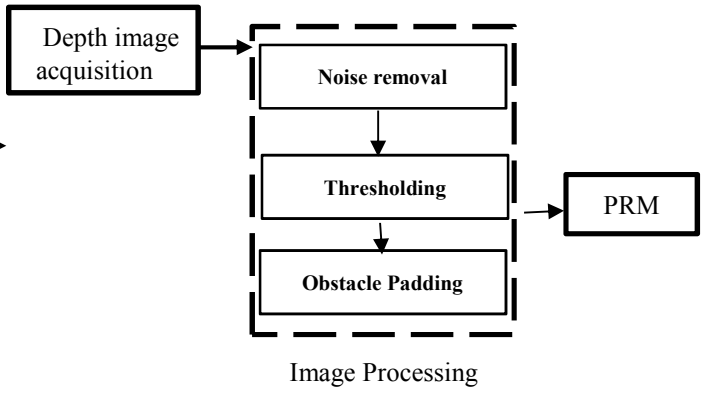

Fig. 3 general diagram of the path planning procedure

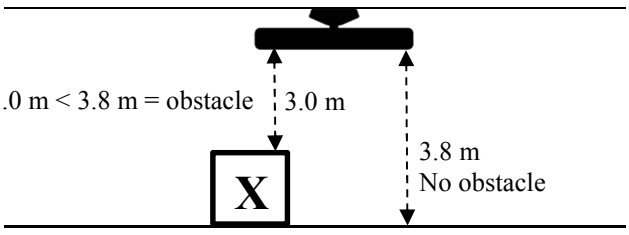

Fig. 4 Obstacle detection

Depth Image acquisition: Depth data received by the depth camera is represented by a 480 × 640 depth image where the value of each pixel represents the distance (in millimeters) from

the camera to the object. The image cannot be used as is as an input to the PRM planner; it should run first into a processing phase.

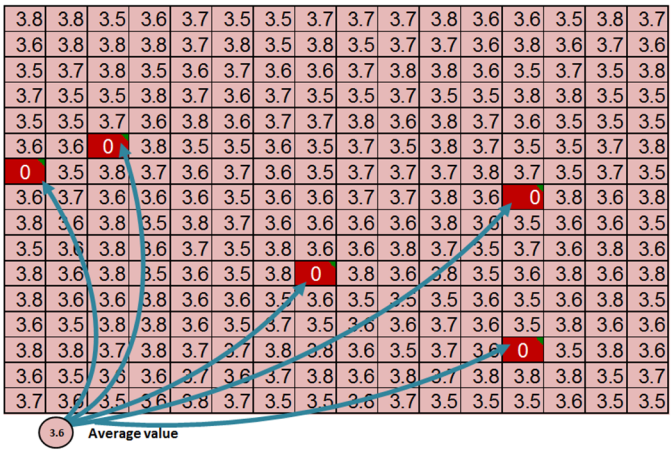

Fig. 5 A noise removal process on a 16 × 16 block

```
Algorithm: [f'] = PrePRM (im)
Input:
  im: depth image captured from the depth camera.
  T: Threshold value. (Usually the height of the
camera from the floor).
  Rad: The radius of the robot.
Output: f: binary image where black areas are
obstacles. To be fed as an input to the PRM
algorithm.
```

```
  // Noise removal
 1: blocks ← divide(im, 16) //divide the depth
image into [16 x 16] blocks
 2: for i := 1 to 1200 blocks
 3: block ← getBlock(i) // Get the current block
 3: index ← find(block==0)// find the indices of
zero pixels in block
 4: no_zero ← size(index)// find the total no.
of zeros in block
 5: if no_zero > 0
 6:    average←AVG(block)// find the average for
values in block
 7:    new_im←replace(0,average,index) // replace
0's with average in index location
 8: end if
 9: end for

        // Thresholding
11: for i := 1 to 480 x 640 pixels
12: if new_im(i) < T
13:    new_im(i) = 0 // change to black, and
consider as obstacle
14:    else new_im(i) = 1 // change to white, and
consider as free space
15: end for

        // Padding
16: index_obs ← find(new_im ==0) // find the
indices of obstacles
17: f ← margin(index_obs,Rad) // add a margin
size of Rad around obstacles.
```

Fig. 6 Pseudocode of the image processing phase

Image Processing: The depth camera is not a fully accurate device where it usually fails to find the corresponding distance for some pixels. Those pixels will have a zero value which means that there is an error computing the distance. This can be viewed as a noise that should be removed before any further work. The suitable way to remove this kind of noise is to use a technique similar to an averaging filter. The depth image is divided into 16 × 16 non-overlapping blocks, and the average value of pixels is computed in all blocks. Each zero value is replaced by the average value in the corresponding block. Figure 5 shows an example of a 16 × 16 block where zeros are being replaced by the average value of the block.

Once the noise has been removed, a thresholding process begins. The main purpose of this process is to extract the obstacles and to produce a binary image where black areas are considered to be obstacles, and white areas are the free space. The main concept in this process is that any pixel that has a depth value detected less than the distance to the floor will be considered to be an obstacle, and turned to black in the new binary image.

The thresholding process can be expressed by:

$$A(x,y) = \begin{cases} 0 & if\ B(x,y) < T \\ 1 & otherwise \end{cases} \quad (1)$$

where A is the new binary image; B is the original depth matrix; T is the threshold value. (Usually the height of the camera from the floor).

The last process to be done to the image is padding. A margin of the robot size must be added around obstacles in the image. The reason of this process is that path planning algorithms does not take in considerations the size of the robot, and simply considers it as a point (pixel) which may cause collisions since it might choose a path that is very close to an obstacle.

IV. PATH PLANNING

The probabilistic roadmap algorithm (PRM) is a path planning technique which solves the problem of determining a path between two points while avoiding collisions [21]. The idea is to take randomly some points in the free space, and then connect them together. This creates a bunch of paths from the starting point to the target. The shortest path among these paths is selected using Dijkstra's shortest path algorithm. The binary image produced from the image processing phase will be the main map that the PRM planner will work on, along with the starting point and the ending points. Moreover, two primary attributes must be selected carefully to ensure having a satisfying result, the first attribute is the Number of Nodes (N) which simply means the number of samples to be used in the calculations. The other attribute is the distance attribute (D). PRM connects all points separated by this distance or less on the map. Increasing these two parameters makes finding a path easier. However, a very huge number can increase the computation time, and might cause the process to fail with real-time applications.

Before exporting the path to the kinematic controller, a conversion from pixels to meters must be done. For any point, the position (x, y) in meters can be computed by:

$$x = \frac{x_{pix} \times x_{total}}{res_x} \quad , \quad y = \frac{y_{pix} \times y_{total}}{res_y} \quad (2)$$

where:
x_{pix} is the x-distance in pixels,
y_{pix} is the y-distance in pixels,
x_{total} is the length of the captured area in meters,
y_{total} is the width of the captured area in meters,
res_x is the length of the captured depth image in pixels ,
res_y is the width of the captured depth image in pixels,

V. MODELING AND CONTROL DESIGN

The mobile robot considered in this paper consists of a wheeled mobile robot shown in Figure 1. The general dynamic equation is described in [16] as follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)\tau - A^T(q)\lambda \quad (3)$$

where $\tau = [\tau_r, \tau_l]^T \in R^r$ is the input vector and consists of motors' torques τ_r and τ_l which act on the right and left wheels, $\lambda \in R^m$ is the vector of constraint forces, $M(q) \in R^{n\times n}$ is a symmetric and positive-definite inertia matrix, $C(q,\dot{q}) \in R^{n\times n}$ is the centripetal and Coriolis vector, $G(q) \in R^n$ is the gravitational vector, $B(q) \in R^{n\times r}$; is the input transformation matrix, and $A(q) \in R^{m\times n}$ is the matrix associated with the constraints. We consider that the robot is moving on a flat terrain and thus $G(q) = 0$. M, C, G, q in Eq. (1) can be expressed as:

$$M(q) = \begin{bmatrix} m & 0 & -2m_w d sin\emptyset \\ 0 & m & 2m_w d cos\emptyset \\ -2m_w d sin\emptyset & 2m_w d cos\emptyset & I \end{bmatrix}$$

$$C(q,\dot{q})\dot{q} = \begin{bmatrix} -2m_w\dot{\emptyset}^2 d cos\emptyset \\ -2m_w\dot{\emptyset}^2 d sin\emptyset \\ 0 \end{bmatrix}; G = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix};$$

$$A^T = \begin{bmatrix} sin\emptyset \\ -cos\emptyset \\ -d \end{bmatrix}; B(q) = \frac{1}{r}\begin{bmatrix} cos\emptyset & cos\emptyset \\ sin\emptyset & sin\emptyset \\ L & -L \end{bmatrix}; q = \begin{bmatrix} x \\ y \\ \emptyset \end{bmatrix}$$

$m = m_c + 2m_w$; $I = I_c + 2m_w(d^2 + L^2) + 2I_m$

where m_c is the mass of the robot without the driving wheels, m_w is the mass of each driving wheel plus the motor rotor, I_c is the moment of inertia of the platform without the driving wheels and I_m is the moment of inertia of each wheel and the motor rotor about a wheel diameter; the kinematic constraints can be denoted as:

$$A(q)\dot{q} = 0 \quad (4)$$

$$\dot{x}sin\emptyset - \dot{y}cos\emptyset = 0 \quad (5)$$

When selecting a full rank matrix $S(q)$ to be a basis of null space A(q), the constraint equation will be:

$$A(q)S(q) = 0 \quad (6)$$

where

$$S = \begin{bmatrix} \frac{r}{2L}(Lcos\emptyset + dsin\emptyset) & \frac{r}{2L}(Lcos\emptyset - dsin\emptyset) \\ \frac{r}{2L}(Lsin\emptyset - dcos\emptyset) & \frac{r}{2L}(Lsin\emptyset + dcos\emptyset) \\ \frac{r}{2L} & -\frac{r}{2L} \end{bmatrix}$$

Therefore, we have:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\emptyset} \end{bmatrix} = S(q)\begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} = S(q)W \quad (7)$$

where $\omega_r; \omega_l$ represent the angular velocities of the right and left wheels and $W = [\omega_l \quad \omega_r]^T$. If we consider $v; \omega$ as the linear and angular velocities of the mobile robot, the relation between $v; \omega$ and $\omega_r; \omega_l$ can be explained as:

$$\begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} = \begin{bmatrix} 1/r & L/r \\ 1/r & -L/r \end{bmatrix}\begin{bmatrix} v \\ \omega \end{bmatrix} \rightarrow W = HV \quad (8)$$

where $V = [v \quad \omega]^T$. From (5) and (6) it is clear that:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\emptyset} \end{bmatrix} = \begin{bmatrix} cos\emptyset & dsin\emptyset \\ sin\emptyset & -dcos\emptyset \\ 0 & 1 \end{bmatrix}\begin{bmatrix} v \\ \omega \end{bmatrix} = SV \quad (9)$$

The derivative of equation (9) gives:

$$\dot{q} = SV \Rightarrow \ddot{q} = \dot{S}V + S\dot{V} \quad (10)$$

In this part, the velocity control based on kinematic model is designed to develop the desired forward and rotational velocities.

Note that the objective is to track a reference trajectory by the mobile platform. Then, the desired position is $q_d = [x_d \quad y_d \quad \phi_d]$ and the desired velocity is $V_d = [v_d \quad \omega_d]^T$. Therefore, the tracking errors are obtained using the Kanayama transformation [17] as follows:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{\phi} \end{bmatrix} = \begin{bmatrix} cos\phi & sin\phi & 0 \\ -sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_d - x \\ y_d - y \\ \phi_d - \phi \end{bmatrix} \quad (11)$$

Using (4), (5) and (11), the error dynamics can be given as follows:

$$\dot{\tilde{x}} = \omega\tilde{y} - v + v_d cos\emptyset \quad (12)$$
$$\dot{\tilde{y}} = -\omega\tilde{x} + v_d sin\emptyset$$
$$\dot{\tilde{\phi}} = \omega_d - \omega$$

The error dynamics (12) are asymptotically stable when using the following velocity control law:

$$V(t) = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1\tilde{x} + v_d cos\tilde{\phi} \\ \omega_d + k_2 v_d\tilde{y} + k_3 v_d sin\tilde{\phi} \end{bmatrix} \quad (13)$$

where $k_1 > 0; k_2 > 0; ; k_3 > 0$ are the controller gains.

To prove proposition 2, we consider the following positive Lyapunov function:

$$W = \frac{1}{2}\tilde{x}^2 + \frac{1}{2}\tilde{y}^2 + \frac{1 - cos\tilde{\phi}}{k_1} \quad (14)$$

It has been proven that the kinematic system of mobile robot, which consisted of Equations (11) and (13), is closed-loop stable [18].

VI. EXPERIMENTAL RESULTS

The experiment was done in a lab where an Asus Xion pro depth camera was mounted on the ceiling. The mobile robot which is shown in Fig. 7 is utilized to demonstrate the effectiveness of the proposed control strategy.

Fig. 7. Pioneer P3-DX

The dynamic parameters for this mobile robot are illustrated in Table 1.
The experimental setup is given in Fig.8

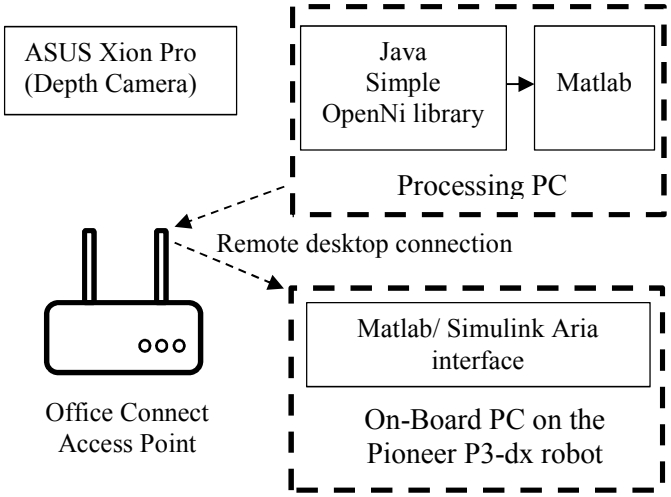

Fig. 8. Experimental steup

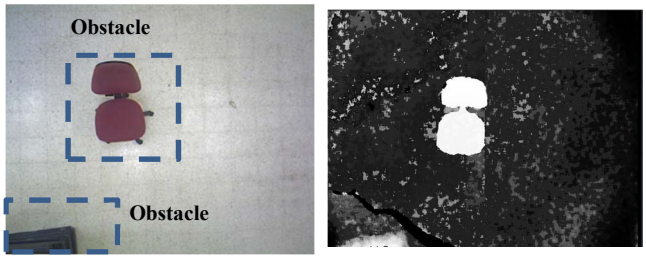

Fig. 9. Vision information. (a)RGB image captured by the ASUS Xtion Pro. (b) Raw depth image captured by the ASUS Xtion Pro.

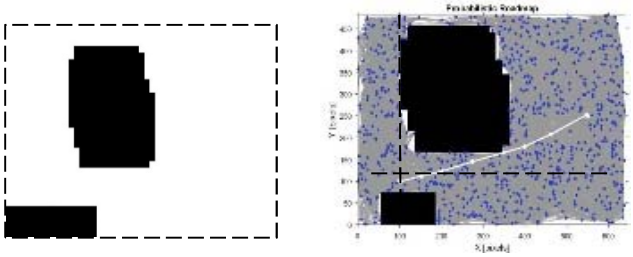

Fig. 10. Path planning. The scene converted to a binary image. (b) PRM path

The number of Samples used for the PRM algorithm is 800 samples, with a connection distance of 100 pixels. The reference trajectory is chosen as $q_{rd} = [x_r \quad y_r \quad \emptyset_r]^T$ then $\dot{x}_r = V_r cos\emptyset_r;\ \dot{y}_r = V_r sin\emptyset_r,\ \dot{\emptyset}_r = \omega_r$. where the linear velocity and the reference angular velocity are chosen as $V_r = 0.2\ m/s$ and $\omega_r = 0.1\ rad/s$, respectively. The initial position is $q(0) = [0 \quad 0 \quad 0^o]^T$. Note that this pose is converted from the [100,100] pixel space to the robot's space by shifting by [-100,-100] in both directions. Then another conversion must be done to translate the pixel-path to a meter-path by using equation (2). The kinematic controller is defined in Eq. (14), where gains are set to be $k_1 = 2;\ k_y = 10;\ and\ k_3 = 6$. The experimental results obtained are shown in figures 11-13.

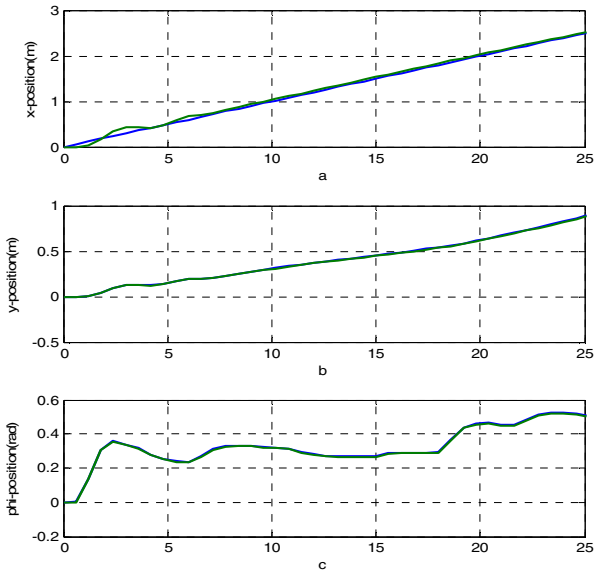

Fig. 11. Tracking control for mobile robot. (a) Tracking trajectory of x-position, (b) Tracking trajectory of y-position (c) Tracking trajectory of ϕ-direction.

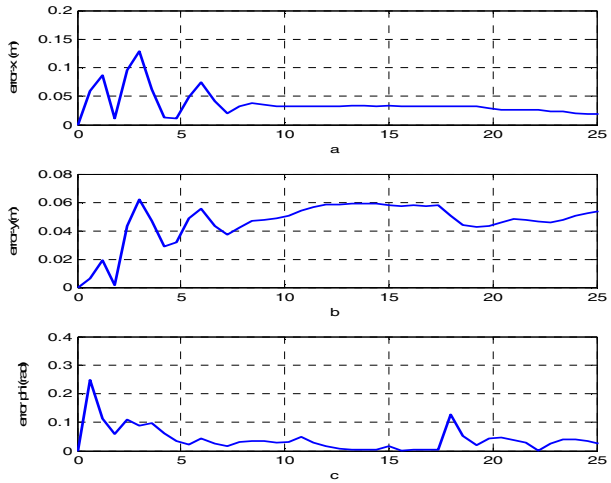

Fig. 12. Tracking error for mobile robot. (a) Tracking error of x-position, (b) Tracking error of y-position (c) Tracking error of ϕ-direction

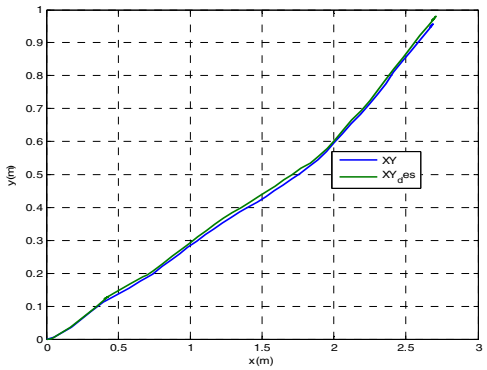

Fig. 13. Tracking trajectory in (X, Y) plan.

Regarding the experimental results, Figure 9(a) shows the raw depth image captured by the ASUS Xtion Pro depth camera where closer objects are brighter. This Image ran into 3 operations: Noise removal, Thresholding, and Obstacle padding. The result of the 3 operations is obtained in figure 9(b). The result shows a successful translation from a raw depth image that has lots of noise and artifacts to a clean binary image. Obstacles are represented by black pixels, and a margin is added around obstacles. The PRM path generator was able to obtain a free-obstacle path as shown in figure 10. The new coordinate system that will be used for the robot is defined. Figure 11 (a) shows a perfect tracking in the x-position. As a result, the related tracking error was very small which confirms such a good tracking. In the y-position, figure 11(b) illustrates an excellent tracking of the desired trajectory. Based on the related tracking error demonstrated in figure 12(b), a good tracking in y-position is obtained. For the ∅-direction, the tracking of the desired trajectory is presented in figure 11(c). It is clear that our controller is able to follow a non-smooth path with a satisfactory tracking. Figure 12(c) shows that the related tracking error converges to zero, which again confirms a good tracking. In the XY-plan, figure 13 shows a good tracking. Finally, from these experimental results, the steady state errors are very small and converge to zero, which demonstrate an effective control performance on differential wheeled mobile robot. Note that for all trajectories, the path was successfully translated from the pixel space to the meter space by using equation (2).

VII. Conclusion

This paper presents a control strategy for differential wheeled mobile robot based on a vision system and was proved by Lyapunov approach. First, a depth camera was used to provide a vision information of the scene. Second, a free-obstacle path was generated by a vision system, then a kinematic controller was designed to generate the forward and rotational velocities that make the robot to follow the desired path efficiently. Experimental results obtained have demonstrated the efficacy of the proposed control method in controlling mobile robots. As a future work, the proposed control strategy will be extended to include a dynamic controller since motors generate control torques, and cannot directly generate the velocity controls.

References

[1] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in cartesian space," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, 1991, pp. 1136-1141.

[2] J.-B. Pomet, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift," *Systems & control letters,* vol. 18, pp. 147-158, 1992.

[3] R. Murray, G. Walsh, and S. Sastry, "Stabilization and tracking for nonholonomic control systems using time-varying state feedback," *IFAC Nonlinear control systems design,* pp. 109-114, 2014.

[4] A. Bloch and S. Drakunov, "Stabilization of a nonholonomic system via sliding modes," in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, 1994, pp. 2961-2963.

[5] C. C. De Wit, H. Berghuis, and H. Nijmeijer, "Practical stabilization of nonlinear systems in chained form," in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, 1994, pp. 3475-3480.

[6] J.-M. Yang and J.-H. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots," *Robotics and Automation, IEEE Transactions on,* vol. 15, pp. 578-587, 1999.

[7] J.-C. Ryu and S. K. Agrawal, "Differential flatness-based robust control of a two-wheeled mobile robot in the presence of slip," in *ASME 2008 Dynamic Systems and Control Conference*, 2008, pp. 915-921.

[8] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *Neural Networks, IEEE Transactions on,* vol. 9, pp. 589-600, 1998.

[9] O. Mohareri, R. Dhaouadi, and A. B. Rad, "Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks," *Neurocomputing,* vol. 88, pp. 54-66, 2012.

[10] B. S. Park, S. J. Yoo, J. B. Park, and Y. H. Choi, "Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty," *Control Systems Technology, IEEE Transactions on,* vol. 17, pp. 207-214, 2009.

[11] L. Huang and L. Tang, "Dynamic Target Tracking Control for a Wheeled Mobile Robots Constrained by Limited Inputs," in *Proceedings of 17th World Congress, IFAC, Seoul, Korea*, 2008, pp. 3087-3091.

[12] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle like vehicles via Lyapunov techniques," *Robotics & Automation Magazine, IEEE,* vol. 2, pp. 27-35, 1995.

[13] G. Eason, B. Noble, and I. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences,* vol. 247, pp. 529-551, 1955.

[14] M. Ito and M. Shibata, "Visual tracking of hand-eye robot for moving target object with multiple feature points," in *2010 11th IEEE International Workshop on Advanced Motion Control (AMC)*, 2010, pp. 240-245.

[15] M. Shibata, H. Eto, and M. Ito, "Visual tracking control for stereo vision robot with high gain controller and high speed cameras," in *Access Spaces (ISAS), 2011 1st International Symposium on*, 2011, pp. 288-293.

[16] T. Fukao, H. Nakagawa, and N. Adachi, "Adaptive tracking control of a nonholonomic mobile robot," *Robotics and Automation, IEEE Transactions on,* vol. 16, pp. 609-615, 2000.

[17] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, 1990, pp. 384-389.

[18] K. Liu and F. L. Lewis, "Decentralized continuous robust controller for mobile robots," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, 1990, pp. 1822-1827.