Springer

KSME

Journal of Mechanical Science and Technology 24 (2010) 365~372
www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-009-1204-8

Sonar-based obstacle avoidance using region partition scheme†

T. T. Quyen Bui and Keum-Shik Hong*

School of Mechanical Engineering, Pusan National University, Busan, 609-735, Korea

(Manuscript Received May 25, 2009; Revised November 6, 2009; Accepted November 6, 2009)

Abstract

This paper addresses an obstacle avoidance problem for a mobile robot in indoor environment. The collision avoidance algorithm utilizes a region partition scheme to analyze the three forward sensory regions of the robot, then to calculate the minimum distances from the center of mass of the robot to the obstacles, respectively, and ultimately to find a navigable region. The supervisor inside the controller receives a series of obstacle avoidance behaviors and makes a decision that allows the robot to successfully navigate in a cluttered environment. In addition, experimental results to illustrate the proposed obstacle avoidance algorithm for the mobile robot in navigation are also presented.

Keywords: Mobile robot; Obstacle avoidance; Region partition; Sensor-based navigation

1. Introduction

In order for mobile robots to be truly versatile, they have to perform given tasks autonomously in unknown, unstructured, unpredicted and dynamic environments. Under these situations, an obstacle avoidance algorithm driven by sensory information must generate the robot motions. In closed-loop control, the obstacle avoidance algorithm should be able to regenerate a collision-free path towards a given target position. The robot moves recursively under the guidance of the perception-action loop (see Fig. 1). This paper focuses on a collision avoidance strategy using only sonar sensors in indoor environments (for the purpose of cost reduction).

The movement of a robot between locations while avoiding collision in real time has been a widely studied subject. The Bug algorithm [1] follows the contour of each obstacle in the robot's way and makes the robot circumnavigate it. The potential field methods (PFM) and their inherent limitation are introduced in [2]. A known problem of the PFM is that the robot gets easily trapped in a U-shape structure. The vector field histogram (VFH) [3] is an obstacle avoidance method that selects a motion direction from a pre-calculated set of solutions (valleys), switching among three different situations: wide-valley situation, narrow-valley situation, and the situation that the robot faces into the target direction. This set of solutions is obtained based on an empirical threshold. The turning of the robot should be different when the robot navigates from very dense environments to less dense ones. The VFH+ is also presented in [4], which is an improved version of VFH with additional considerations of threshold hysteresis, the robot's size, and masking blocked sectors. Another obstacle avoidance technique is the dynamic window approach [5]. The motion commands (translational and rotational velocities of the robot) are carried out directly in the velocity space. An objective function, which includes the heading to the target, the robot's forward velocities and obstacle clearance, is considered, and the motion commands are computed by optimizing the objective function. Other notable results include an interactive multiple model algorithm [6], a three-layer control architecture (deliberative, sequencing, reflexive) [7], a collision-free motion coordination for multiple heterogeneous robots [8], a predictive navigation approach with nonholonomic and minimum turning radius constraints [9], a time-varying feedback control via the chained form [10], and others [11-14].

In this paper, a strategy to avoid obstacles is carried out by using a region partition. First, a threshold is set up to ignore objects somewhat far away from the robot, and only objects within this threshold are considered; second, to ensure the safety of the robot, a security zone around the robot is set; and third, the measurement range is partitioned into three regions of 60 degrees (see Fig. 6). Analyzing the measured data, we can determine which region among the three would be the best navigable region for the planned motion of the robot. The supervisor module (see Fig. 1) calculates motion suitability and sends the motion commands (translational and rotational

† This paper was recommended for publication in revised form by Associate Editor Sungsoo Rhim

*Corresponding author. Tel.: +82 51 510 2454, Fax.: +82 51 514 0685
E-mail address: kshong @pusan.ac.kr

velocities) to the low-level driver modules through a serial port. This process is repeated under a perception-action loop. The outcome is a chain of online command motions that drive the robot to intermediate positions while avoiding collision.

Contributions of this paper are an efficient combination of geometric analysis and decision making, and this combination allows the robot to use an array of sonars to successfully navigate in indoor cluttered environment. The conditions that allow the robot to navigate forward safely are considered. The width of the robot is also taken into account while it is ignored in the VFH [3] and in the fuzzy logic methods [15, 16]. In addition, the accurate computation of the translational and rotational velocities (v, ω) based on circular motion and the direction solution received from the obstacle avoidance module is performed, and the algorithms presented in [2-5] use an assumption that the robot makes a circular path.

One approach to autonomous navigation is the wall following method [17-19]. However, robot navigation by the wall following method is less versatile and the robot collides with walls [18] or sometimes the robot cannot reach to the target. In our work, the experimental results will demonstrate that the obstacle avoidance algorithm can be carried out completely for mobile robot navigation in a disordered environment. Moreover, the algorithm does not require an accurate analytical model of the environment, allows the robot to navigate in a narrow way easily, and takes less computation. This algorithm can be extended to 360-degree visibility and can be used for multiple mobile robots.

This paper is organized as follows. The kinematic model of a wheeled robot is overviewed briefly, and obstacle detection using a sonar array is considered in Section 2. In Section 3, the obstacle avoidance algorithm based on a region partition and sensory measurements is discussed in detail. Furthermore, velocity design strategy is also presented. Experimental results that illustrate the performance of the obstacle avoidance algorithm applied to a mobile robot are provided in Section 4. Finally, conclusions are drawn in Section 5.

Table 1. Configuration of sonar sensors in the local reference frame.

Order [i]	x_i [m]	y_i [m]	Height [m]	δ_i [rad]
0	0.069	0.136	0.215	1.571
1	0.114	0.119	0.215	0.873
2	0.148	0.078	0.215	0.524
3	0.166	0.027	0.215	0.175
4	0.166	-0.027	0.215	-0.175
5	0.148	-0.078	0.215	-0.524
6	0.114	-0.119	0.215	-0.873
7	0.069	-0.136	0.215	-1.571

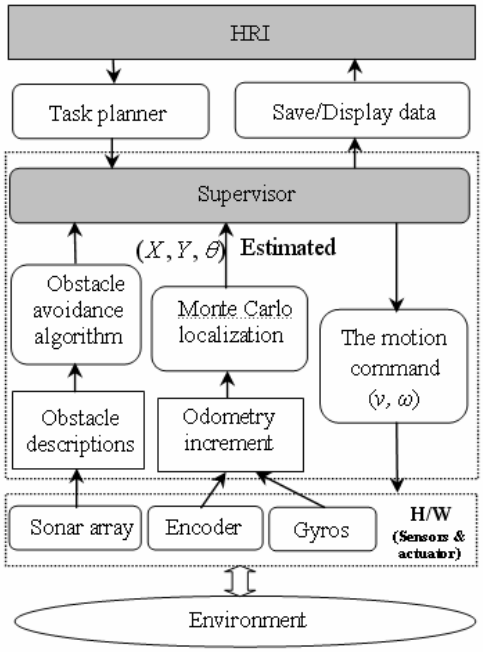


Fig. 1. Control architecture: perception-action loop.

2. Kinematics and obstacle detection

2.1 Kinematics

We consider a wheeled mobile robot moving on a flat surface under the classical hypothesis "rolling without slipping" between the wheels and the ground. A schematic of such a robot of rectangular shape is shown in Fig. 3: l_1 is the length, l_2 is the width, and l_w is the distance between the two rear wheels. Let $O-X-Y$ and $o-x-y$ be the global coordinate frame and the body coordinate frame attached to the robot, respectively, and (*X*, *Y*) denote the coordinate of the centre of the robot in the global frame. Let θ denote the rotational angle of the *x*-axis with respect to the *X*-axis. The body coordinate frame $o-x-y$ is established in such a way that the origin of the frame is located at the mid point of the rear axle, and the *x*-axis coincides with the main axis of the robot. The pose (configuration) of the robot is given by (*X*, *Y*, θ).

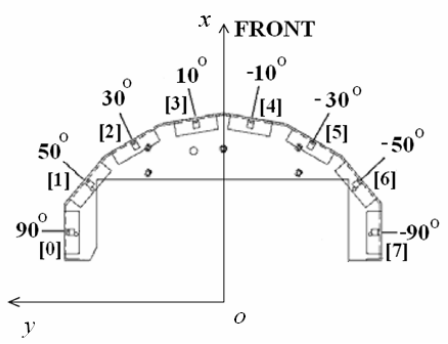
Due to the no-slippage assumption, a nonholonomic constraint holds as follows:

$$\dot{X}\sin\theta = \dot{Y}\cos\theta\,. \tag{1}$$

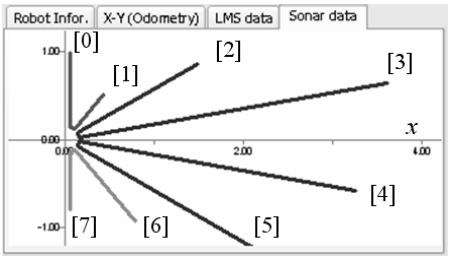
Under this constraint, the robot motion can be described by only two motion commands (v, ω) as follows:

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

where *v* is the translational velocity of the robot along the *x*-axis and ω is the rotational velocity of the robot. Also, $|v| \le v_{\max}$ and $|\omega| \le \omega_{\max}$ are assumed, where $v_{\max}$ and $\omega_{\max}$ are their maximum values, respectively.



(a) Distribution of eight sonar sensors.



(b) Example of sonar data: the robot's origin is (0, 0).

(c) Pioneer P3-DX.

Fig. 2. Sonar array and an example of sonar data.

2.2 Obstacle detection using sonar sensors

Eight sonar sensors were attached around the robot at the height of 0.215 m, namely, one in right- and left-side of the robot, respectively, and six in the forward-side at 20-degree intervals, see Fig. 2(a). Their locations are fixed. Fig. 2(b) shows captured sonar data, where [*i*] indicates the *i*-th sonar in the array, *i* = 0,..,7. Table 1 shows the location data of the sensors in the robot coordinate frame.

In Fig. 4, let (X_i, Y_i, α_i) and (x_i, y_i, δ_i) be the arrangement of the *i*-th sonar in the global and local coordinate frames, respectively, where (X_i, Y_i) and (x_i, y_i) are the coordinates of the *i*-th sonar sensor in the inertia and body coordinate frames, respectively, and α_i and δ_i are the angles that the sonar ray makes with the *X*-axis and the *x*-axis, respectively. Let d_i be the measured distance from the *i*-th sonar sensor to the obstacle, in which P_i is assumed to be the contact point of the sonar ray on the obstacle. Let D_i be the distance from the center of the robot to P_i. Since there are eight sonar sensors, eight object coordinates are obtained: They are, (X_i^p, Y_i^p) and (x_i^p, y_i^p), $i = 0, \cdots, 7$ in the global and local coordinate frames, respectively.

The coordinate of an *i*-th object point P_i in the local coordinate is calculated as follows:

$$x_i^p = x_i + d_i \cos\delta_i,$$
$$y_i^p = y_i + d_i \sin\delta_i. \tag{3}$$

Thus, D_i can be calculated as

$$D_i^2 = (x_i^p)^2 + (y_i^p)^2. \tag{4}$$

Substituting (3) into (4) yields

$$D_i = \sqrt{d_i^2 + x_i^2 + y_i^2 + 2d_i(x_i\cos\delta_i + y_i\sin\delta_i)}. \tag{5}$$

On the other hand, the global sensor position is calculated by a matrix equation of the form

$$\begin{bmatrix} X_i \\ Y_i \\ \alpha_i \end{bmatrix} = \begin{bmatrix} X \\ Y \\ \theta \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ \delta_i \end{bmatrix}. \tag{6}$$

Finally, using the robot pose (X, Y, θ) and the sensor data (x_i^p, y_i^p), the coordinate of the *i*-th obstacle P_i in the global reference frame is given as follows:

$$X_i^p = X + x_i^p\cos\theta - y_i^p\sin\theta,$$
$$Y_i^p = Y + x_i^p\sin\theta + y_i^p\cos\theta. \tag{7}$$

3. Obstacle avoidance algorithm

In this section, a methodology of avoiding obstacles using sonar sensors is discussed. A moving direction ϕ_d is determined based on a region partition and geometrical analysis. Then, motion commands (v, ω) are archived based on the hypothesis that the robot movement is a circular motion.

3.1 Determining moving direction

Consider a situation that the robot is approaching to a point F (Fig. 5). Let the distance from the robot to point F be ρ and the angle between the X-axis and the line that joins the center of the robot and point F be γ, respectively. Then, they are obtained as follows.

$$\rho = \sqrt{(X_F - X)^2 + (Y_F - Y)^2},$$
$$\gamma = \text{atan2}(Y_F - Y, X_F - X), \tag{8}$$

where (X_F, Y_F) is the position of point F in the global coordinate. The distance ρ is reduced and becomes to zero when

the robot approaches to F. Our goal is to control the robot in the presence of obstacles while approaching to a goal point. The moving direction equals the difference angle $\xi = \gamma - \theta$ when the robot navigates in a free obstacle environment. The determination of moving direction is discussed as follows.

Unfortunately, sonar sensors cannot detect the shape of an obstacle(s) exactly because only finite number of sensors is used and discrete data are provided. Also, shape detection becomes more difficult in a cluttered environment. Therefore, the nearness diagram algorithm [20] and its extended method [21] will not work efficiently in this case. In this paper, we propose a scheme of splitting the forward range (180 degrees) into three regions of 60 degrees (see Fig. 6). In Fig. 6, R denotes the radius of the minimum circle that contains the robot (0.22 m in this paper), and D_s is the security distance (0.1 m has been set). Upon the calculation of D_i, the obstacle avoidance algorithm scrutinizes the three forward regions to pick the best one with respect to passableness for the robot's forward movement. Let R_{th} be the threshold distance for detecting obstacles (4 m was set); that is, objects beyond R_{th} are ignored (in programming, $D_i = R_{th}$ if $D_i \geq R_{th}$). Let $D^L_{\min}$, $D^C_{\min}$, and $D^R_{\min}$ denote the minimum distances to the obstacles in the left, center, and right regions, respectively, and $D_{\min}$ be the minimum of the three, as follows:

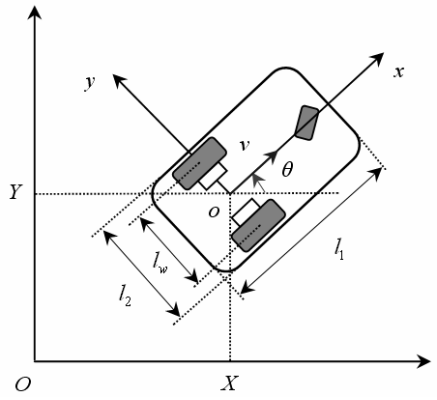


Fig. 3. Nonholonomic robot differentially driven by two rear wheels.

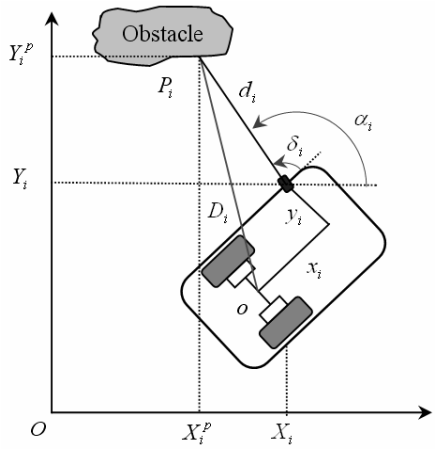


Fig. 4. Global and local coordinates of the i-th sonar sensor.

$$
\begin{aligned}
D^L_{\min} &= \min_i \{D_i : \ i = 0, 1, 2\}, \\
D^C_{\min} &= \min_i \{D_i : \ i = 2, 3, 4, 5\}, \\
D^R_{\min} &= \min_i \{D_i : \ i = 5, 6, 7\}, \text{ and} \\
D_{\min} &= \min_i \{D_i : \ \ i = 0, \cdots, 7\}.
\end{aligned} \tag{9}
$$

In Fig. 6, let $\bar{P}_i$ be the intersection point of the i-th sonar ray with the line(s) that is parallel to the x-axis, which is tangent with the circle of radius $(R + D_s)$ in the left and right side, respectively. Let $\bar{d}_1$ and $\bar{d}_2$ be the measured distance from the first and the second sonar to $\bar{P}_1$ and $\bar{P}_2$, respectively. Then, due to the symmetry of the sensor arrangement, we have

$$
\begin{aligned}
\bar{d}_1 &= \frac{R + D_s - y_1}{\sin \delta_1} \ \left(= \frac{R + D_s + y_6}{-\sin \delta_6} \right), \\
\bar{d}_2 &= \frac{R + D_s - y_2}{\sin \delta_2} \ \left(= \frac{R + D_s + y_5}{-\sin \delta_5} \right).
\end{aligned} \tag{10}
$$

From (9), the distances $o\bar{P}_1$ and $o\bar{P}_2$ are obtained as follows:

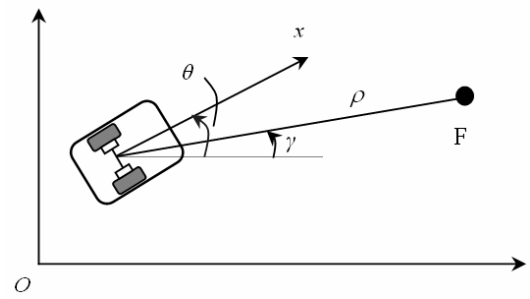


Fig. 5. A situation that the robot is reaching to a point F.

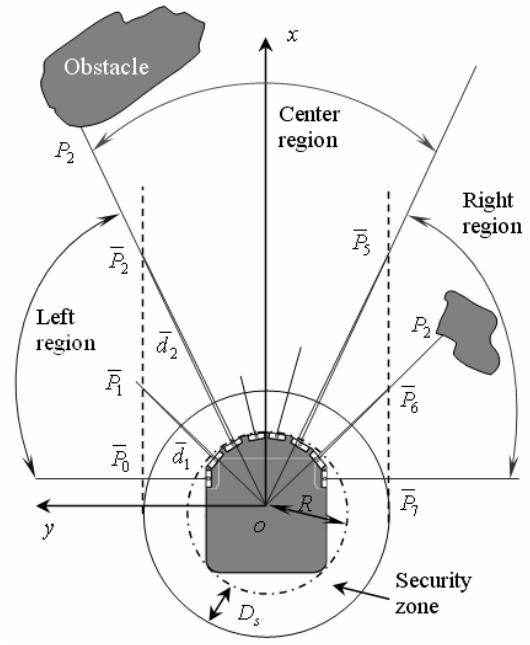


Fig. 6. Partition into three regions and the security zone for safety.

$$\overline{oP_1} = \sqrt{\overline{d}_1^2 + x_1^2 + y_1^2 + 2\overline{d}_1(x_1\cos\delta_1 + y_1\sin\delta_1)}\ ,$$
$$\overline{oP_2} = \sqrt{\overline{d}_2^2 + x_2^2 + y_2^2 + 2\overline{d}_2(x_2\cos\delta_2 + y_2\sin\delta_2)}\ , \quad (11)$$

From the geometry in Fig. 6, it is clear that the robot can move forward if there is no obstacle inside the rectangle made by $\overline{P}_0\overline{P}_2\overline{P}_5\overline{P}_7$. The conditions that allow the robot to move forward are $D_{\min}^C \geq o\overline{P}_2$, $D_1 \geq o\overline{P}_1$, and $D_6 \geq o\overline{P}_1$. If there is any obstacle inside the security zone, an emergency is called, and the *Pause* state is sent immediately to the robot. At the *Pause* state, the motion commands are $(v, \omega) = (0, 0)$. Otherwise, a navigable region is specified.

A series of obstacle avoidance motions are executed using a set of *if-then* rules as follows:

```
If( (D_min < R + D_s) || ( ρ ≈≈ 0 )) state = Pause;
else{
  If  min(D_1, D_6) ≥ oP_1 {
    If   D_min^C  ≥  oP_2  {// Option 1
      If  min(D_min^L, D_min^R) > oP_2 {
        If  ((D_min^L < D_min^R) && (D_min^C < D_min^R − D))
            state = TurnRight;
        else if  ((D_min^L > D_min^R) && (D_min^C < D_min^L − D))
            state = TurnLeft;
        else state = Forward;
      }
      else state = Forward;
    }
    else{//Option 2
        if  (D_min^C < D_min^R − D)  state = TurnRight;
        else state = TurnLeft;
    }
  else{//Option 3
    if  ((D_min^L < oP_1) && (D_min^R > oP_1))
        state = TurnRight;
    else if  ((D_min^L > oP_1) && (D_min^R < oP_1))
        state = TurnLeft;
    else state = Backward;
  }
}
```

In *Option 1*, if $\min(D_{\min}^L, D_{\min}^R) \leq o\overline{P}_2$, the controller puts a top priority to the forward movement. If $\min(D_{\min}^L, D_{\min}^R) > o\overline{P}_2$, there are three possibilities: first, if $D_{\min}^L > D_{\min}^R$ and $D_{\min}^C < D_{\min}^L - \overline{D}$ (where $\overline{D}$ is a threshold value, $\overline{D} < R_{th}$), the left region is selected as a navigable region, and thus the robot will turn left; second, if $D_{\min}^L < D_{\min}^R$ and $D_{\min}^C < D_{\min}^R - \overline{D}$, the robot will turn right; otherwise, the center region will be the last accessible region for the robot to move forward.

In *Option 2*, the robot cannot move forward because $D_{\min}^C < o\overline{P}_2$. Thus, the algorithm should consider either to a turn to the left or a turn to the right. If $D_{\min}^L < D_{\min}^R - \overline{D}$, the right region becomes a passable region; thus, the robot will turn right. If $D_{\min}^R < D_{\min}^L - \overline{D}$, the robot will turn left.

In *Option 3*, the given conditions do not allow the robot to move forward, so turning left or right needs to be considered. If $D_{\min}^R < o\overline{P}_1$ and $D_{\min}^L > o\overline{P}_1$ (or $D_{\min}^R > o\overline{P}_1$ and $D_{\min}^L < o\overline{P}_1$), the robot has to turn left (or right). If both conditions are not satisfied, then the robot has to move backward until it escapes from the deadlock.

3.2 Designing velocities

This subsection discusses a scheme for determining the robot's velocities according to the solution to motion direction obtained from the obstacle avoidance algorithm. During the execution of motions, the velocities remain constant in each loop-time. At constant velocities, the robot moves in a circular path or a straight line. Note that a straight motion and a pure rotation are both circular paths with infinity and zero radii, respectively.

The relationship between the translational and rotational velocities is given by $v = r\omega$, where r is the radius of circular motion. Since the velocities must satisfy the conditions $|v| \leq v_{\max}$ and $|\omega| \leq \omega_{\max}$, respectively, the following angle is introduced.

$$\phi_0 = \operatorname{atan}\frac{\omega_{\max}}{v_{\max}}\ . \quad (12)$$

Assume that, at a given time instant k, the controller gives motion commands (v, ω) that makes the robot pass a point (x_k, y_k) by circular motion. Then, the radius of a circle r and the moving direction $\phi_d \in [-\pi, \pi]$ are given by

$$r = \frac{x_k^2 + y_k^2}{2y_k}\ , \quad (13)$$
$$\phi_d = \begin{cases} \operatorname{atan2}(x_k, r - y_k) & \text{if } y_k \geq 0 \\ -\operatorname{atan2}(x_k, y_k - r) & \text{otherwise} \end{cases} \quad (14)$$

respectively, since the robot has to satisfy the circular motion of $x_k^2 + (y_k - r)^2 = r^2$ (see [20]).

The robot moves at its maximum velocities if obstacles are far away from the robot. However, if obstacles appear within a predetermined range, the velocities will become proportional to the remaining distance between the robot and the obstacle.

Let μ be a coefficient (a weighting factor) defined by

$$\mu = \begin{cases} 1, & \overline{d}_{obs} \geq \overline{d}_{\min} \\ \dfrac{\overline{d}_{obs}}{\overline{d}_{\min}}, & \text{otherwise} \end{cases} \quad (15)$$

where $\overline{d}_{\min}$ is the prearranged threshold distance used to check whether the maximum velocity $v_{\max}$ should be used or not (in this paper, $\overline{d}_{\min}$ = 4 m) and $\overline{d}_{obs}$ is the distance from the robot to the closest obstacle. Finally, the following control laws are proposed in (16). The sign preserves the "forward"

$$(v,\omega)=\begin{cases}(\mu v_{\max},\ \mu v_{\max}\tan\phi_d\ ), & \phi_d\in[-\phi_0,\phi_0]\\ (\mu\omega_{\max}|\cot\phi_{\mathrm{d}}\ |,\ \mu\omega_{\max}\mathrm{sgn}(\phi_d)), & \phi_d\in\{[-\pi/2,-\phi_0)\cap(\phi_0,\pi/2]\}\\ (-\mu v_{\max},\ \mu v_{\max}\tan(\mathrm{sgn}(\phi_d)\pi-\phi_d)\ ), & \phi_d\in\{[-\pi,-\pi+\phi_0]\cap[\pi-\phi_0,\pi]\}\\ (-\mu\omega_{\max}|\cot(\mathrm{sgn}(\phi_{\mathrm{d}})\pi-\phi_d)\ |,\ \mu\omega_{\max}\mathrm{sgn}(\phi_d)), & \text{otherwise.}\end{cases} \tag{16}$$

and "backward" motion. If $\phi_d=0$ or π, $\tan\phi_d=0$, then the robot motion is a straight forward or backward motion, respectively. Similarly, if $\phi_d=\pi/2$ or $-\pi/2$, the robot motion is pure rotation to the left or right, respectively.

4. Experimental results

4.1 Mobile robot platform and settings

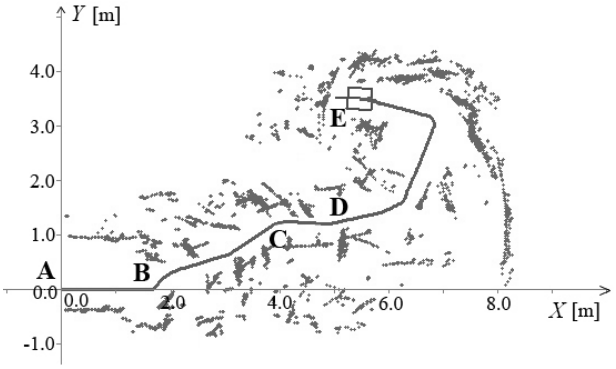
The collision avoidance algorithm was tested on a Pioneer P3-DX mobile base, a nonholonomic vehicle equipped with a sonar array (8 sonar sensors in the front) with dimensions of l_1 = 0.44 m and l_2 = 0.38 m. Because the robot moves in an indoor environment with highly dense obstacles, the safety of the humans around the robot must be preserved. The maximum translational and rotational velocities were set to $v_{\max}$ = 0.5 m/s and $\omega_{\max}$ = 0.523 rad/s, respectively. Other parameter values were set as follows: R = 0.22 m, D_s = 0.1 m, R_{th} = 4.0 m, and $\bar{D}$ = 0.3 m. All the calculations of the algorithm were implemented in the C++ program language on a Windows operating system. The control loop time was 100 ms.

4.2 Experiment

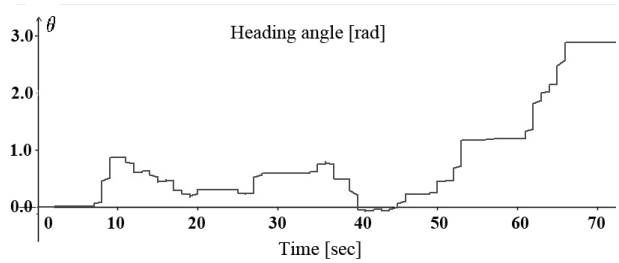
A challenge was that the robot must search a target automatically during its navigation in an unknown and dynamic scenarios (the robot knows only the position of the target). Fig. 7 shows experimental data, in which the feasibility of the collision avoidance algorithm was demonstrated. The target position was (5.5, 3.5). The robot's pose (position and orientation) at time t = 0 was (0, 0, 0). The initial and goal locations of the robot and their intermediate points were denoted as A-E. Four different zones were tested as follows:

- A to B: a sparse zone (less dense),
- B to C: a densely cluttered zone,
- C to D: a narrow passage,
- D to E: a zone in which the target is surrounded by obstacles.

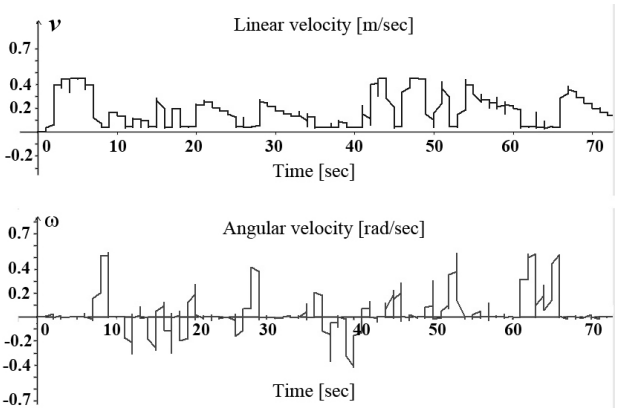
The target position with respect to the robot position has been updated in the control loop. By using the distance ρ, the angle ξ, and sonar data as inputs, the controller directed the robot's motion toward the target. The trajectory of the robot and the sonar points collected in the experiment are shown in Fig. 7(a). The heading angle, the linear velocity, and the angular velocity of the robot are depicted in Fig. 7(b) and Fig. 7(c), respectively. From A to B, the robot was able to move on a straight line because there was no obstacle. After that, the robot had to search for a movable direction in the presence of scattered obstacles (B to D). Fig. 7(d) shows the overall experimental environment (left) and a snapshot of the



(a) The path executed by the robot (dots are sonar points).



(b) Heading angle of the robot.



(c) Linear and angular velocities.

(d) The overall experimental environment (left) and the robot passing through a narrow passage (right).

Fig. 7. Experiment: navigation from A to E.

robot passing through a narrow passage at D. At the goal point, the robot's pose was (5.466, 3.496, 3.066). This experiment took 72 s. In our experiment, only the forward zone of the mobile robot was considered, hence, the moving direction is $\varphi_d \in [-\pi/2, \pi/2]$. For a narrow passage, the robot could move through the passageway by reducing D_s to a small value.

5. Conclusions

An obstacle avoidance methodology using ultrasonic sensors for the mobile robot was presented. The velocity generation strategy is based on the elementary circular path to the direction solution of motion. Moreover, the obstacle avoidance method does not require a precise analytical model of the environment and gives another insight on the effective use of a sonar array in obstacle avoidance for the mobile robot. The experiment results obtained also showed that the robot navigated in an unknown, complex environment as well as in narrow passage easily. The algorithm can be extended to 360-degree visibility and applied to many types of mobile robot platforms as an approach well-suited for robot navigation in unknown, cluttered environments.

Acknowledgment

This work was supported by the Regional Research Universities Program (Research Center for Logistics Information Technology, LIT) granted by the Ministry of Education & Human Resources Development, Korea.

References

[1] V. Lumelsky and A. Stepanov, Incorporating range sensing in the robot navigation function, *IEEE Trans. on Systems, Man, and Cybernetics*, 20 (5) (1990) 1058-1069.

[2] Y. Koren and J. Borenstein, Potential field methods and their inherent limitation for mobile robot navigation, *Proc. of IEEE Int. Conf. on Robotics and Automation*, Sacramento, California, USA (1991) 1398-1404.

[3] J. Borenstein and Y. Koren, The vector field histogram - fast obstacle avoidance for mobile robots, *IEEE Trans. on Robotics and Automation*, 7 (3) (1991) 278-288.

[4] I. Ulrich and J. Borenstein, VFH+: Realiable obstacle avoidance for fast mobile robot, *Proc. of IEEE Int. Conf. on Robotics and Automation*, Leuven, Belgium (1998) 1572-1577.

[5] D. Fox, W. Burgard and S. Thrun, The dynamic window approach to collision avoidance, *IEEE Robotics and Automation Magazine*, 4 (1) (1997) 23-33.

[6] Y.-S. Kim and K.-S. Hong, An IMM algorithm with federated information mode-matched filters for AGV, *International Journal of Adaptive Control and Signal Processing*, 21 (7) (2005) 533-555.

[7] K.-S. Hong, T. A. Tamba and J. B. Song, Mobile robot control architecture for reflexive avoidance of moving obstacle, *Advanced Robotics*, 22 (13-14) (2008), 1397-1420.

[8] N. Y. Ko, D. J. Seo and R. G. Simmons, Collision free motion coordination of heterogeneous robots, *Journal of Mechanical Science and Technology*, 22 (11) (2008) 2090-2098.

[9] A. Widyotriatmo, B. Hong and K.-S. Hong, Predictive navigation of an autonomous vehicle with nonholonomic and minimum turning radius constraints, *Journal of Mechanical Science and Technology,* 23 (2) (2009) 381-388.

[10] T. A. Tamba, B. Hong and K.-S. Hong, A path following control of unmanned autonomous forklift, *International Journal of Control, Automation, and Systems,* 7 (1) (2009) 113-122.

[11] Q. H. Ngo, K.-S. Hong and I. H. Jung, Adaptive control of axially moving systems, *Journal of Mechanical Science and Technology*, 23 (11) (2009) 3071-3078.

[12] Q. H. Ngo and K.-S. Hong, Skew control of a quay container crane, *Journal of Mechanical Science and Technology*, 23 (12) (2009).

[13] J. Borenstein and Y. Koren, Obstacle avoidance with ultrasonic sensors, *IEEE Journal of Robotics and Automation*, 4 (2) (1998) 213-218.

[14] G. D. Castillo, S. Skaar, A. Cardenas and L. Fehr, A sonar approach to obstacle detection for a vision-based autonomous wheelchair, *Robotics and Autonomous System*, 54 (2006) 967-981.

[15] F. Abdessemed, K. Benmahammed and E. Monacelli, A fuzzy-based reactive controller for a nonholonomic mobile robot, *Robotics and Autonomous System*, 47 (2004) 31-46.

[16] A. Zhu and S. X. Yang, A fuzzy logic approach to reactive navigation of behavior-based mobile robots, *Proc. IEEE Int. Conf. on Robotics and Automation*, New Orleans, USA, 5 (2004) 5054-5050.

[17] A. G. Lamperski, O. Y. Loh, B. L. Kutscher and N. J. Cowan, Dynamical wall following for a wheeled robot using a passive tactile sensor, *Proc. of Int. Conf. on Robotics and Automation*, Barcelona, Spain (2005) 3838-3843.

[18] W. A. Lewinger, M. S. Watson and R. D. Quinn, Obstacle avoidance behavior for a biologically-inspired mobile robot using binaural ultrasonic sensors, *Proc. of Int. Conf. on Intelligent Robots and Systems*, China (2006) 5769-5774.

[19] A. Bemporad, M. Di Marco and A. Tesi, Sonar-based wall following control of mobile robots, *Journal of Dynamic Systems, Measurement, and Control*, 122 (1) (2000) 227-230.

[20] J. Minguez and L. Montano, Nearness diagram (ND) navigation: collision avoidance in trouble scenarios, *Trans. on Robotics and Automation*, 20 (1) (2004) 45-59.

[21] J. Minguez and L. Montano, Abstracting vehicle shape and kinematics constraints from obstacle avoidance methods, *Autonomous Robots*, 20 (1) (2006) 43-59.

T. T. Quyen Bui received her B.S. and M.S. degrees in Hanoi University of Technology, Vietnam, in 2001 and 2006, respectively. She is currently a Ph.D. program student in the School of Mechanical Engineering, Pusan National University, Korea. Her research interests include robotics, vision systems, and navigation of autonomous vehicles.

Keum-Shik Hong received the B.S. degree in mechanical design and production engineering from Seoul National University in 1979, the M.S. degree in ME from Columbia University in 1987, and both the M.S. degree in applied mathematics and the Ph.D. degree in ME from the University of Illinois at Urbana-Champaign in 1991. Dr. Hong serves as Editor-in-Chief of the Journal of Mechanical Science and Technology. He served as an Associate Editor for Automatica (2000-2006) and as an Editor for the International Journal of Control, Automation, and Systems (2003-2005). Dr. Hong received Fumio Harashima Mechatronics Award in 2003 and the Korean Government Presidential Award in 2007. His research interests include nonlinear systems theory, adaptive control, distributed parameter system control, robotics, and vehicle controls.