Journal of
Marine Science and Technology

Volume 22 | Issue 3 Article 7

A WHEELED MOBILE ROBOT PATH-TRACKING SYSTEM BASED ON IMAGE PROCESSING AND ADAPTIVE CMAC

Jih-Gau Juang
Department of Communications, Navigation and Control Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C, jgjuang@mail.ntou.edu.tw

Ko-Jui Hsu
Department of Communications, Navigation and Control Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C

Chih-Min Lin
Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan, R.O.C

Recommended Citation

Juang, Jih-Gau; Hsu, Ko-Jui; and Lin, Chih-Min (2014) "A WHEELED MOBILE ROBOT PATH-TRACKING SYSTEM BASED ON IMAGE PROCESSING AND ADAPTIVE CMAC," *Journal of Marine Science and Technology*: Vol. 22: Iss. 3, Article 7.
DOI: 10.6119/JMST-012-1228-1
Available at: https://jmstt.ntou.edu.tw/journal/vol22/iss3/7

DOI: 10.6119/JMST-012-1228-1

A WHEELED MOBILE ROBOT PATH-TRACKING SYSTEM BASED ON IMAGE PROCESSING AND ADAPTIVE CMAC

Jih-Gau Juang[1], Ko-Jui Hsu[1], and Chih-Min Lin[2]

Key words: mobile robot, path following, CMAC, image processing.

ABSTRACT

This study entailed applying image-processing techniques and an intelligent control method to a wheeled mobile robot (WMR) for real-time trajectory recognition and tracking. The WMR was fitted with a charge-coupled device (CCD) camera to capture images of the environment, and the hue-saturation-intensity color space was used to classify the color features of the captured images. The WMR calculates the relative position of a target object by using image-processing and edge-detection algorithms. This paper proposes a cerebellar model articulation controller (CMAC) and adaptive CMAC to replace conventional proportional-integral-derivative (PID) controller for guiding the WMR to track a desired path. Moreover, using Lyapunov's stability theory to optimize the learning rate ensures that the proposed control system is stable. The tracking performance of several controllers used to track various path patterns were compared. Finally, according to the theoretical analysis, simulations, and experimental results of this study, the proposed control scheme is effective.

I. INTRODUCTION

Wheeled mobile robots (WMRs) are used in various applications. Current research on WMRs focuses on how to make WMRs function effectively in various environments. Many studies on intelligent robots have examined home services, health care, space exploration, military, and entertainment [10, 14, 18]. WMRs are intelligent systems that integrate various functions, such as environmental sensing, dynamic decision making and planning, and behavior control and execution. Recently, many studies have proposed intelligent tracking control systems for applications in robotics. Chen [5] used encoded information to develop path planning and parking control systems for a WMR. In that study, Lyapunov's theory was used to analyze the dynamic behavior of continuous and discrete controls. Shieh [17] incorporated a camera and fuzzy controller into the path planning system of a service robot. Wang [20] proposed a real-time image and fuzzy proportional-integral-derivative (PID) controller for the path tracking and parking control systems of a WMR. In [8, 12, 22], color, shape, and facial recognition systems were used to identify target objects. Subsequently, the distance between the target and sensors was measured to improve the path tracking control system of a vehicle. In this study, a network camera used to detect a desired path and a cerebellar model articulation controller (CMAC) were integrated to enable a WMR to track multiple paths.

Previously, most wheeled mobile robots were equipped with various sensors designed for specific operations. The WMR proposed in this study used a visual sensor for path recognition and tracking control. Fuzzy controllers are the most widely used intelligent controllers in WMR path tracking applications [6, 11, 18, 23]. Because fuzzy control relies on a designer's knowledge of the controlled system, mathematical control law is unnecessary. However, fuzzy control systems are limited by their incapacity to learn and inability to adapt to environmental changes. To track paths and analyze the environment, WMRs typically use a camera to capture information on their surroundings.

In this study, the WMR was equipped with a CCD camera to assist it in detecting a desired path. Captured images were converted from the red-green-blue (RGB) color space to the hue-saturation-intensity (HSI) color space to remove noise from the image [3, 9]. Subsequently, Hough transform [15] was applied to detect edges in the captured images and extract the location of a target segment. The deviation in relative distance and angle between the target trajectory and WMR were input into a CMAC network model to control the robot's movement. In addition, an adaptive learning rule was incorporated into the CMAC design [21]. In this study, a CMAC and adaptive CMAC were applied to control the speed and trajectory of the WMR. The CMAC and image processing

Paper submitted 08/31/12; revised 12/26/12; accepted 12/28/12. Author for correspondence: Jih-Gau Juang (e-mail: jgjuang@mail.ntou.edu.tw).

[1] Department of Communications, Navigation and Control Engineering, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

[2] Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan, R.O.C.

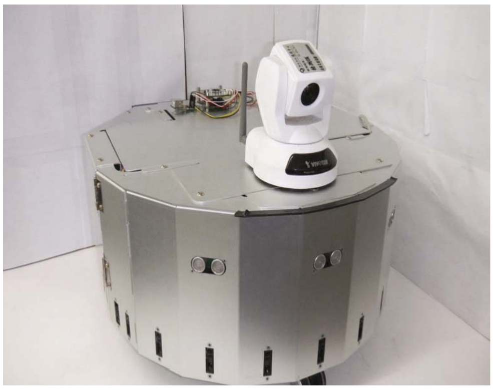


Fig. 1. Ihomer wheeled mobile robot (WMR) and camera.

were used to construct an intelligent control scheme that enabled the WMR to recognize and track a desired path automatically. The relative distance and position of the target object were used as CMAC inputs, whereas the speed control, angle control, and acceleration were the CMAC outputs. An Ihomer WMR was used to test the proposed system [16].

II. IMAGE PROCESSING

The proposed control scheme was implemented using an Ihomer WMR (Fig. 1), which is a two-wheeled differentially driven car-like robot. Two small caster wheels fitted at the rear and front of the WMR were driven by two 18-V DC motors (7.2 mNm torque), each controlling one wheel. The dimensions, maximal speed, and weight of the WMR are 450 × 380 × 255 mm (length × width × height), 1.6 m/s, and 15.2 kg, respectively. As shown in Fig. 1, the WMR was fitted with a PZ6114/6124 network camera. Distances were calculated using a geometry method to calculate the distance between the first and final pixels of a detected edge [19].

1. Hue-Saturation-Intensity Color Space

The human eye can sense light at wavelengths ranging from approximately 400 to 700 nm (i.e., visible light). Color models are mathematical models describing how colors can be represented as tuples of numbers (typically comprising three or four color components). The primary colors of pigment can be combined to create various colors. The pigments are magenta, cyan, yellow, and black. Many possible color spaces can be constructed based on these **colors**. For example, the colors displayed on a computer monitor are typically defined in the RGB color space [3]. In this study, the HSI color space was used [9]. Colors can be created by adjusting the hue, saturation, and brightness values. Hue describes the shade of color and where it appears in the color spectrum. Red, yellow, and purple are words that describe hue. Saturation refers to the purity of a hue relative to a white reference. Saturation is expressed as percentage values; pure red that has no white is an example of 100% saturation. Like brightness, intensity is a subjective descriptor that is impossible to measure. Intensity embodies the achromatic notion of intensity and it is a key factor in describing color sensation. The HSI color scheme can be applied to visible light to improve the efficiency of color classification and image processing systems [9].

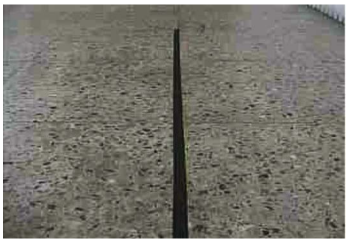


Fig. 2. RGB image.

Fig. 2 shows an RGB image of the test environment. The test environment was in a colorful space; therefore, removing environmental noise from target objects was difficult. The following equations can be used to transform RGB color space values into HSI values:

$$H = \cos^{-1}\frac{\frac{1}{2}(R-G)+(R-B)}{\sqrt{(R-B)^2+(R-B)(G-B)}}, \; B \le G \tag{1}$$

$$H = 2\pi - \cos^{-1}\frac{\frac{1}{2}(R-G)+(R-B)}{\sqrt{(R-B)^2+(R-B)(G-B)}}, \; B > G \tag{2}$$

$$S = 1 - \frac{3}{(R+G+B)}\left[\min\left(R, G, B\right)\right] \tag{3}$$

$$V = \frac{1}{3}(R+G+B) \tag{4}$$

Figs. 3 and 4 depict HSI models based on color triangles and circles. The triangles and circles are perpendicular to the vertical intensity axis. Fig. 5 illustrates an RGB image converted to the HSI color space. When images are processed in the HSI color space, the integrity of path lines in the color space enable more information to preserved; therefore, the I component was used as the basis of the image operator in this study.

A binary image was used to change image brightness values from 0 (black) to 255 (white). In the simplest case, an image is composed of one or several objects of relatively high intensity, which are viewed against a background of relatively low intensity. When transforming captured images into binary format, setting an appropriate threshold is a crucial because it determines how effectively an object can be tracked. A binary image is the image in pixels per point compared with a preset threshold value [1]. The image in Fig. 5(c) is shown in Fig. 6 as a binary image.

Performing median operations implies that high, low, or noisy values appear at the top or bottom of a sorted list. Thus,

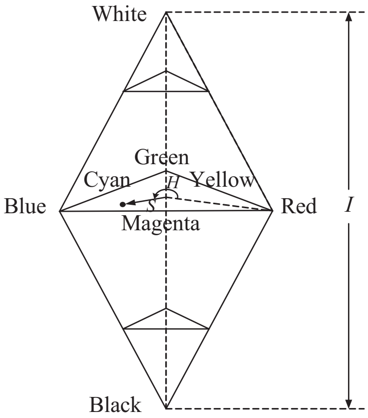


Fig. 3. The HSI color model based on triangular [9].

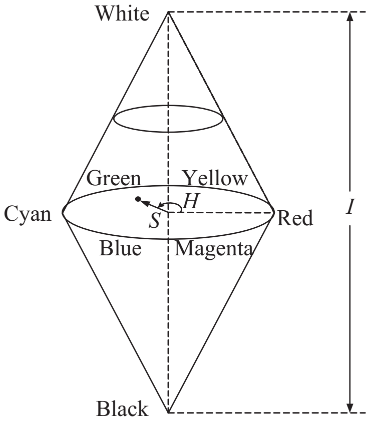
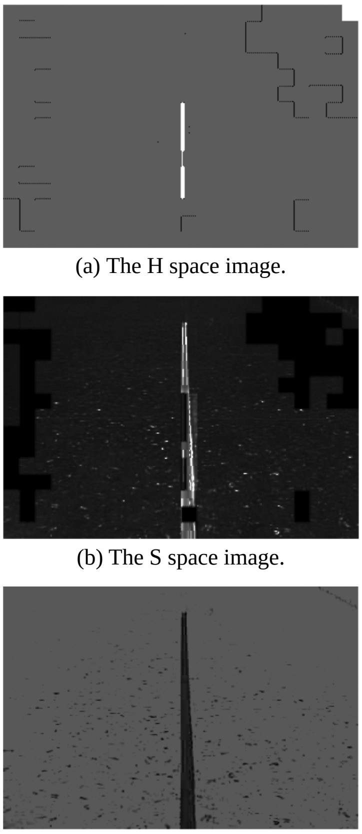


Fig. 4. The HSI color model based on circular [9].



(a) The H space image.

(b) The S space image.

(c) The I space image.

Fig. 5. From RGB image to HSI color space.

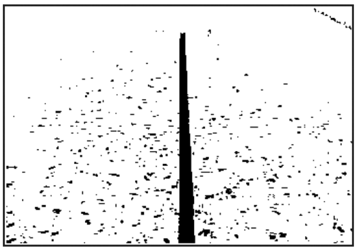


Fig. 6. The binary I space image.

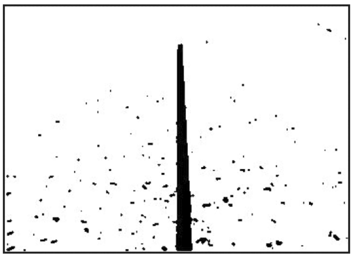


Fig. 7. The median filtering image.

a median filter generally replaces noisy values with values that are similar to the value of adjacent pixels [15]. Fig. 7 shows the image in Fig. 6 after a median filter was applied.

2. Erosion and Dilation

Given sets A and B, the erosion of A by B, denoted as $A \Theta B$, comprises all points $\omega = (x, y)$ at which B_w appears in A. To perform the aforementioned erosion, B is superimposed over A, and all places in which it fits are identified and marked in the corresponding (0, 0) point of B. A set of such points forms an erosion [15]. Fig. 8 shows an erosion of the image shown in Fig. 7.

Given pixel sets A and B, the dilation of A by B is expressed as $A \oplus B$. For each point where $x \in B$, the corresponding point in A is translated. Subsequently, the union of all translations is retained. Dilation is also known as Minkowski addition [15]. Fig. 9 shows an example of dilation, which was applied to the image shown in Fig. 8.

3. Canny Edge Detection

Edge detection procedures can reduce the number of data and filter redundant information while preserving the critical structural properties of an image. The Canny edge detection algorithm has been applied as an edge detector in numerous studies [4]. The Canny edge detector first smooths an image to eliminate noise. Subsequently, the image gradient is identified to accentuate regions with high spatial derivatives.

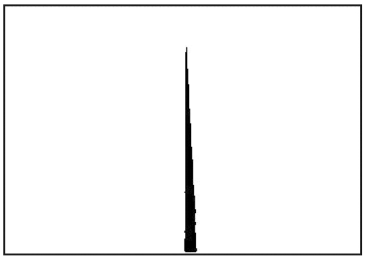


Fig. 8. The erosion image.

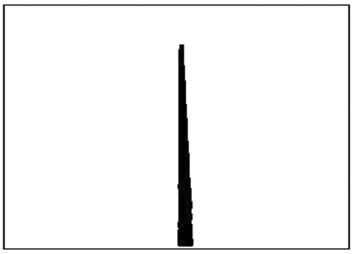


Fig. 9. The dilation image.

$\frac{1}{115}$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Fig. 10. Discrete approximation to gaussion function with σ = 1.4 [4].

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Fig. 11. The Sobel operator performs a gradient measurement [4].

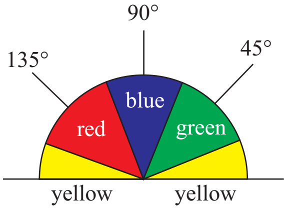


Fig. 12. This as taking a semicircle and dividing it into 5 regions [4].

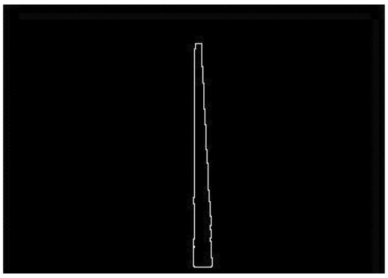


Fig. 13. The Canny edge detection image.

This method is suitable for detecting dynamic objects. The edge detection procedure comprises four steps. Because the Canny edge detector is sensitive to noise in raw image data, a filter is applied based on the first derivative of a Gaussian function. At the first step, the raw image is convolved using a Gaussian filter to remove noise from the original image before edge detection is performed. Fig. 10 shows the Gaussian mask used in this study. The prerequisite σ is the standard deviation.

At the second step, the edge strength is determined based on the image gradient. A Sobel operator performs a two-dimensional spatial gradient measurement of an image. Subsequently, the approximate absolute gradient magnitude (i.e., edge strength) at each point can be identified. The Sobel operator uses a pair of 3 × 3 convolution masks. As shown in Fig. 11, one mask estimates the gradient along the x-axis (i.e, columns), and the other mask estimates the gradient along the y-axis (i.e., rows). The magnitude or edge strength of the gradient is then approximated using (3).

$$|G| = |G_x| + |G_y| \tag{5}$$

At the third step, only four possible directions exist for describing the surrounding pixels; horizontal (0°), positive diagonal (45°), vertical (90°), or negative diagonal (135°). Nonmaximal suppression is applied to trace the edge in the edge direction and to suppress any redundant pixels (i.e., those that do not form an edge). Subsequently, a thin line is formed in the output image. They are edge points and only the local maximal values are used. Accordingly, any edge direction within the yellow range (0° to 22.5° or 157.5° to 180°), green range (22.5° to 67.5°), blue range (67.5° to 112.5°), or red range (112.5° to 157.5°) is set to 0°, 45°, 90°, or 135°, respectively. Fig. 12 shows a semicircle.

Finally, hysteresis can be applied to eliminate streaking. The process requires high and low threshold values as inputs. Any pixel value exceeding the high threshold is immediately identified as an edge, and any adjacent pixels with a value higher than the low threshold are also identified as an edge. Fig. 13 shows the image in Fig. 9 after it was transformed using the Canny edge dector.

4. Hough Transform

To establish a boundary between regions, it may be necessary to fit a line between points. However, this process is potentially time consuming and computationally inefficient, particularly where many edge points exist. One method for identifying boundary lines involves using the Hough

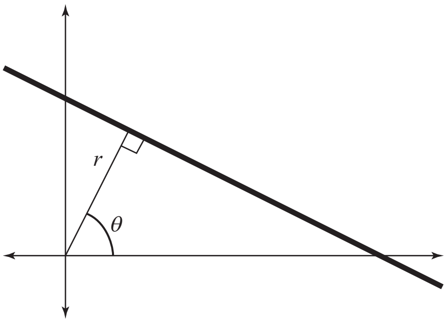


Fig. 14. The line and its parameters.

transform [15]. Although the Hough transform was originally designed to detect lines in images, it can be modified to detect shapes. The concept is simple. Suppose (x, y) is a point in a binary image. The equation $y = ax + b$ can be used to identify all pairs (a, b) that satisfy this equation. Subsequently, the identified pairs can be plotted in an accumulator array, which is a transform array. Each point in the image is mapped to a line in the transform, where the points corresponding to the highest number of intersections correspond to the strongest line in the image.

However, this implementation of the Hough transform is problematic because vertical lines cannot be identified; specifically, a vertical line cannot be expressed as $y = ax + b$ because a represents a gradient and a vertical line has an infinite gradient. Therefore, another line parameterization is required. Consider a general line, as shown in Fig. 14. Any line can be described using the two parameters r and θ, where r is the perpendicular distance from the line to the origin, and θ is the angle of the line perpendicular to the x axis. This parameterization can be applied to identify vertical lines as those where $\theta = 0$. If negative values are allowed for r, then the value of θ can be restricted to $-90 < \theta \leq 90$.

Given this parameterization, the equation of the line can be determined. First, the point (p, q) where the perpendicular line meets the original line is $(p, q) = (r\cos\theta, r\sin\theta)$. In addition, the gradient of the perpendicular line is $\tan\theta = \sin\theta/\cos\theta$. Let (x, y) be any point on the line, the gradient of which can be expressed as

$$\frac{\Delta x}{\Delta y} = \frac{y-q}{x-p} = \frac{y-r\sin\theta}{x-r\sin\theta} \tag{6}$$

However, because the gradient of the perpendicular line is $\tan\theta$, the gradient of the original line is

$$-\frac{1}{\tan\theta} = -\frac{\cos\theta}{\sin\theta} \tag{7}$$

According to (5) and (6),

$$\frac{y-r\sin\theta}{x-r\sin\theta} = -\frac{\cos\theta}{\sin\theta} \tag{8}$$

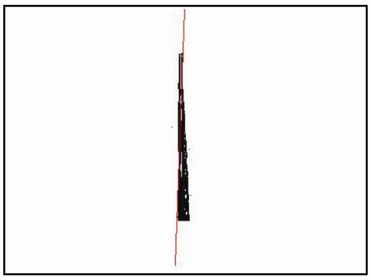


Fig. 15. Red line found by the Hough transform.

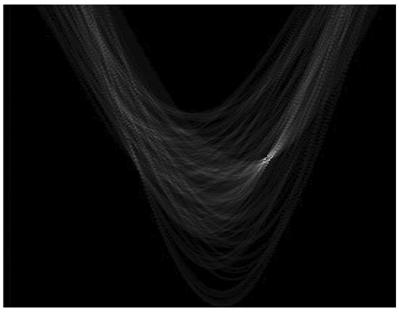


Fig. 16. Corresponding line in the transformed space.

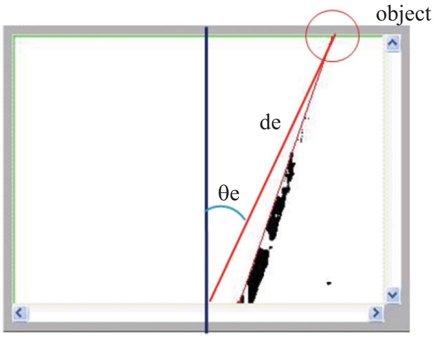


Fig. 17. From Hough Transform information.

Subsequently, (8) can be reorganized as follows [15]:

$$x\cos\theta + y\sin\theta = r \tag{9}$$

For each pixel (x, y) in the image, r is computed for each value of θ, and the result is assigned to the appropriate position in the (r, θ) array. Consequently, the values of (r, θ) that are the highest in the array correspond to the strongest lines in the image [22]. A transform line is superimposed over the image in Fig. 15, and this line appears in the transformed image shown in Fig. 16.

Because the WMR should remain on the desired path, the Hough transform line should be adjusted when necessary to ensure that it remains positioned at the middle of a captured image. Fig. 17 shows the output of the Hough transform, which is then input into the controller. Let θ_e denote the relative angle of the WMR trajectory and anticipated position,

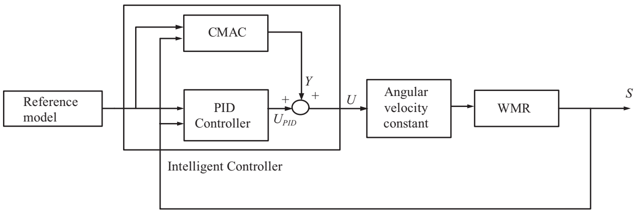


Fig. 18. The CMAC control scheme.

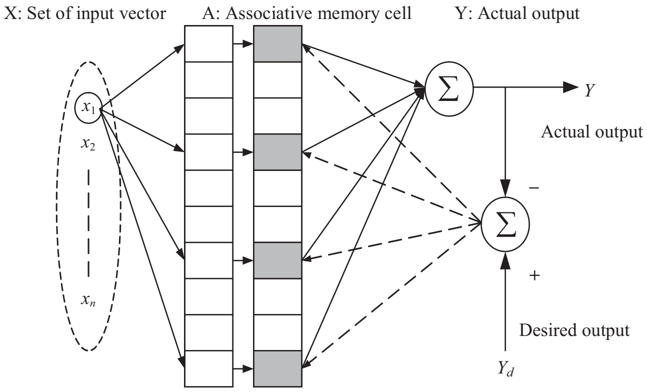


Fig. 19. Conventional CMAC structure.

and let d_e represent the relative distance between the CCD camera and anticipated position.

III. CONTROL SCHEME

The CMAC is a type of associative memory network (AMN), which is a class of artificial neural network structures. It is an associative memory learning algorithm based on the human cerebellum. The function of the CMAC is similar to a lookup table technique, which represents a complex nonlinear system. In this study, a PID controller was used to construct an adaptive hybrid CMAC for the proposed WMR. In addition, Lyapunov's stability theory [2] was used to identify an adaptive learning rule for the hybrid CMAC. Fig. 18 shows an overview of the proposed control scheme, where the control signal U is the sum of the PID controller output and CMAC output. The CMAC output compensates for the control signal.

1. Structure of CMAC

Fig. 19 shows the structure of the CMAC, which is a type of AMN. CMACs yield faster self-learning rates than do normal neural networks by the use of less adjustments of memory weights and feature adequate local generalizability. CMACs function similarly to lookup tables, and the CMAC output is obtained from a linear combination of weights stored in the system memory. The CMAC stores data (knowledge) into overlapping storage hypercubes (recall region) in such a manner that the stored data can be recalled easily through a process of association. Two types of operation are included in the CMAC; one type involves calculating the output result, and the other type entails learning and adjusting the weight. The CMAC output can be obtained using the mapping process $X \rightarrow A \rightarrow Y$ when the input is $x \in X \subset R^N$ with a corresponding function $y \in Y \subset R^M$, where A denotes an M-dimensional storage space, and $\mathbf{a} \in A \subset R^M$ is the associated binary vector. Let the input x address N ($N < M$) storage hypercubes. The mapping process $A \rightarrow Y$ is used to compute the output as follows:

$$y(x) = a^T w(x) = \sum_j^N a_j(x) w_j,\ j = 1, 2, ..., N \tag{10}$$

where w_j is the weight of the j-th storage hypercube, and $a_j(x)$ is a binary factor indicating whether the j-th storage hypercube is addressed by the input x. At the network learning stage, the CMAC modifies the weights of the storage hypercubes according to the error between the desired and real outputs. The weight-updating rule is expressed as

$$w_j^{(i)} = w_j^{(i-1)} + \Delta w_j^{(i-1)} = w_j^{(i-1)} + \frac{\alpha}{m} a_j \left(y_d - \sum_{j=1}^{N} w_j^{(i-1)} a_j \right) \tag{11}$$

where i is the i-th iteration, y_{d} denotes the desired output, m represents the number of addressed hypercubes, and α is the learning rate. Initially, the input vector with the blocks in associative memory cell is a crisp relationship. The relationship between the input condition and association intensity is "activated" or "not activated."

2. Stability Analysis of Adaptive CMAC

An adaptive learning rate is adopted to address the limitation of the conventional CMAC on the crisp relation. Here, the discrete-time Lyapunov function is applied to define α. Let the tracking error $e(t)$ be expressed as follows [7, 13, 21]:

$$e(t) = y_d - y(t) \tag{12}$$

where t is the time index. A discrete-type Lyapunov function can be expressed as

$$V = \frac{1}{2} e^2(t) \tag{13}$$

Thus, the change in Lyapunov's function can be obtained by computing

$$\Delta V = V(t+1) - V(t) = \frac{1}{2}[e^2(t+1) - e^2(t)] \tag{14}$$

The error difference can be expressed as

$$\Delta e(t) \approx \left[\frac{\partial e(t)}{\partial W} \right] \Delta W(t) \tag{15}$$

Using a gradient descent method yields

$$\Delta w_j(t) = -\frac{\alpha}{m}\frac{\partial e(t)}{\partial W} \tag{16}$$

Because

$$\frac{\partial V(t)}{\partial w_j(t)} = e(t)\frac{\partial e(t)}{\partial w_j(t)} = (y_d - y(t))(\alpha_j) \tag{17}$$

Thus

$$\Delta w_j(t) = \frac{\alpha}{m}\alpha_j(y_d - y(t)),\ j = 1, 2, \ldots\ldots, N \tag{18}$$

$$\Delta W(t) = \begin{bmatrix} \Delta w_1(t) \\ \Delta w_2(t) \\ \vdots \\ \Delta w_N(t) \end{bmatrix} = \frac{\alpha}{m}\alpha_j(y_d - y(t)),\ j = 1, 2, \ldots\ldots, N$$

$$= \frac{\alpha}{m}\mathbf{a}(y_d - y(t)) \tag{19}$$

According to (10) and (12),

$$\frac{\partial e(t)}{\partial w_j(t)} = -a_j, \frac{\partial e(t)}{\partial W(t)} = -\mathbf{a}^T \tag{20}$$

According to (11) and (18),

$$\Delta V = \Delta e(t)\left[e(t) + \frac{1}{2}\Delta e(t)\right] \tag{21}$$

According to (15) and (21),

$$\Delta V = \left[\frac{\partial e(t)}{\partial W(t)}\frac{\alpha}{m}\mathbf{a}e(t)\right]\left\{e(t) + \frac{1}{2}\left[\frac{\partial e(t)}{\partial W(t)}\frac{\alpha}{m}\mathbf{a}e(t)\right]\right\} \tag{22}$$

Because $\frac{\partial e(t)}{\partial W(t)} = -\mathbf{a}^T$,

$$\Delta V = \left[-\frac{1}{2}\frac{\alpha}{m}e^2(t)\right]\left[2 - \frac{\alpha}{m}\|a\|^2\right] \tag{23}$$

If $\left[2 - \frac{\alpha}{m}\|a\|^2\right] > 0$, then $\Delta V < 0$; in other words,

$$\frac{2\alpha}{m} > \alpha > 0 \tag{24}$$

where ΔV becomes negative definite, implying that $e(t) \to 0$ for $t \to \infty$. Thus, convergence of the adaptive CMAC learning process is guaranteed. The control system is thus locally and asymptotically stable.

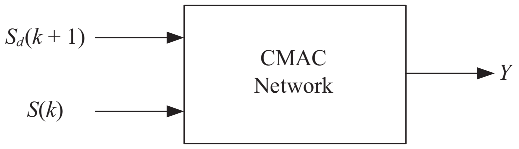


Fig. 20. The control process of CMAC.

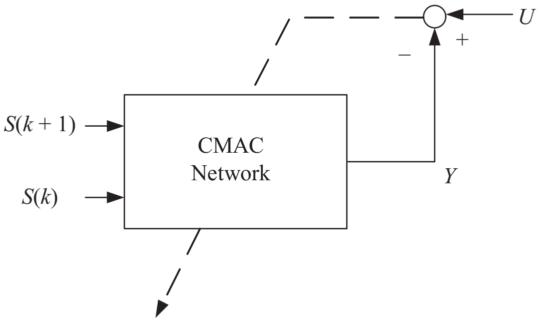


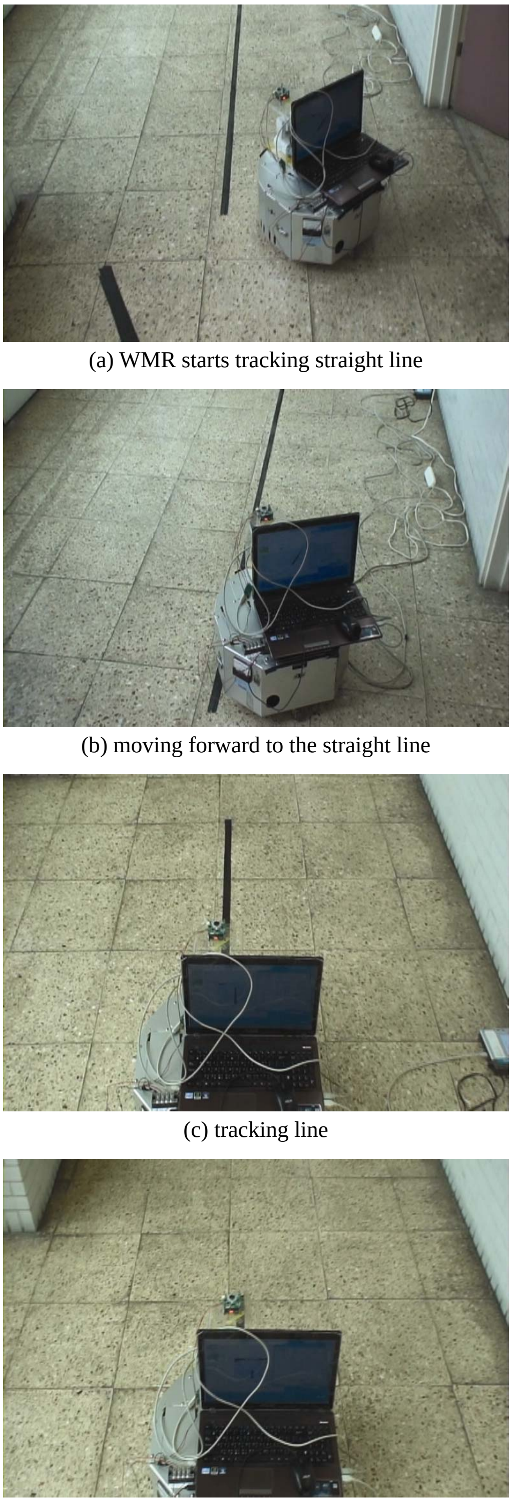
Fig. 21. The learning process of CMAC.

3. Learning Process

Fig. 18 shows the results of applying the CMAC to track the trajectory of the WMR. The positions of the desired path are used as inputs for the PID controller and CMAC. The input of the angle controller U is the summation of the PID controller output U_{PID} and CMAC output Y. The output of the angle controller is the control signal input for the WMR model. The conventional PID controller stabilizes the WMR tracking control and facilitates the CMAC learning process, thereby improving the performance of the intelligent controller. At each sampling instant k, the function of the CMAC comprises two phases; recall and learning. First, the CMAC applies $S_d(k+1)$ and $S(k)$ to address the corresponding memory weights to generate an output Y in the recall phase, where $S(k)$ is the output of the dynamic WMR model at k, and $S_d(k+1)$ represents the desired dynamic WMR model output at the subsequent time step (Fig. 20). During the recall phase, Y is used to calculate the demanded control signal U. Subsequently, Y is integrated with the PID controller output U_{PID} to form the demanded control signal U. During the learning phase (Fig. 21), the value of U (obtained during the recall phase) is considered the desired output. The error obtained from $U - Y$ is then used to update the corresponding memory weights stored at locations $S(k)$ and $S_d(k+1)$. After several iterations, the error converges, and the CMAC network can compensate for the PID controller [21].

IV. EXPERIMENT RESULTS

This section presents a comparison of the PID controller,



(a) WMR starts tracking straight line

(b) moving forward to the straight line

(c) tracking line

(d) stopping location

Fig. 22. Tracking straight line.

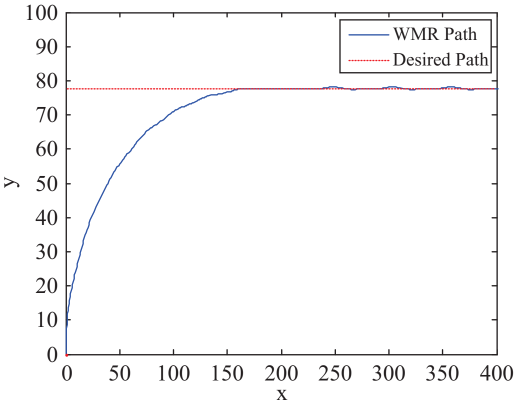


Fig. 23. Tracking straight line by PID controller.

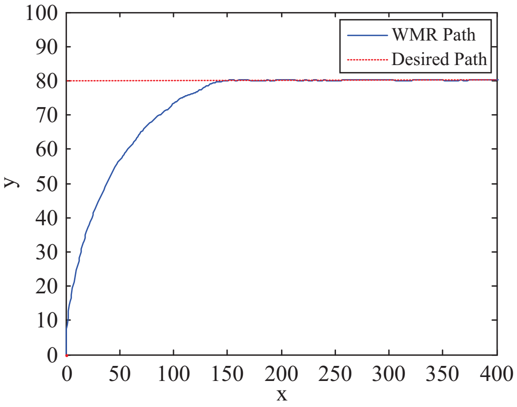


Fig. 24. Tracking straight line by CMAC controller.

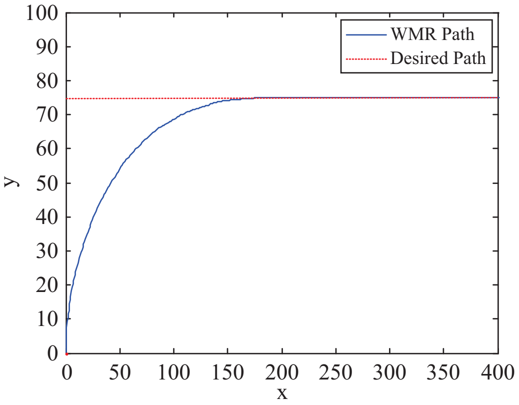


Fig. 25. Tracking straight line by adaptive CMAC controller.

CMAC, and adaptive CMAC in tracking various paths. Fig. 22 shows the WMR tracking a straight line. Fig. 22(a) shows the initial location and Fig. 22(d) shows the final location.

Figs. 23-25 show the desired straight line and real path taken using the PID controller, CMAC, and adaptive CMAC, where the dotted (solid) line represents the desired (recorded real) path. Figs. 26-28 show the distance error between the desired straight line and the real path of the WMR. For comparison, Table 1 shows the distance error values obtained using the PID controller, PID-CMAC, and adaptive PID-CMAC. Fig. 29 shows the WMR tracking an irregular path, and Fig. 30 shows the recorded and desired paths taken using

Table 1. Distance error of following a straight line.

Method	PID	CMAC	Adaptive CMAC
Mean Distance Error (cm)	1.604	1.201	1.087

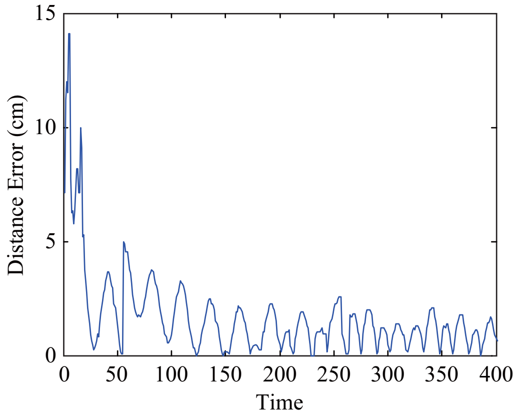


Fig. 26. Distance error of PID controller.

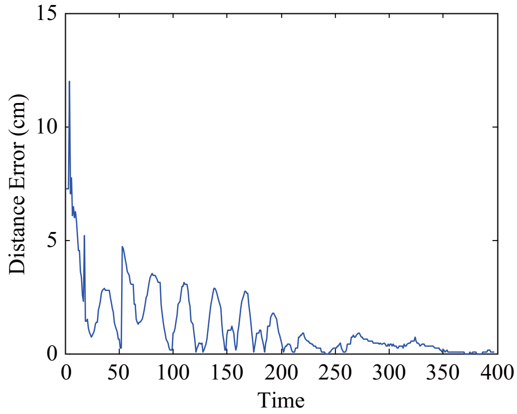


Fig. 27. Distance error of CMAC controller.

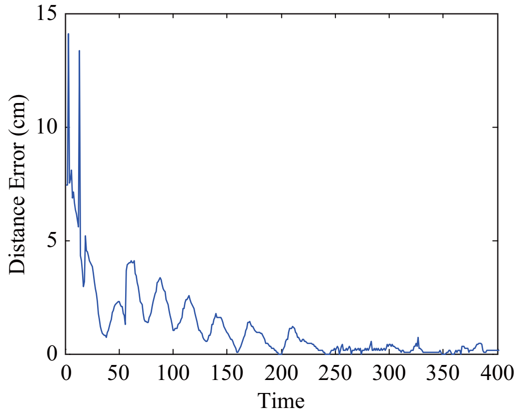
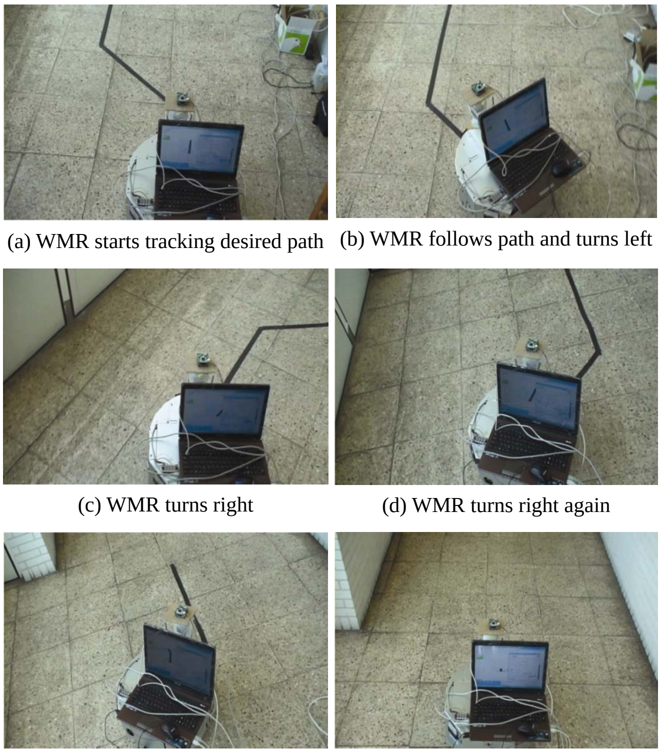


Fig. 28. Distance error of adaptive CMAC controller.

the various controllers. For further comparison, Table 2 shows the distance error values obtained using the various controllers. The table shows that the PID-CMAC controller outperformed

Table 2. Distance error of the irregular path.

Method	PID	CMAC	Adaptive CMAC
Mean Distance Error (cm)	1.906	1.490	1.312



(a) WMR starts tracking desired path (b) WMR follows path and turns left

(c) WMR turns right (d) WMR turns right again

(e) WMR turns left again (f) WMR reaches ending position

Fig. 29. Tracking irregular path.

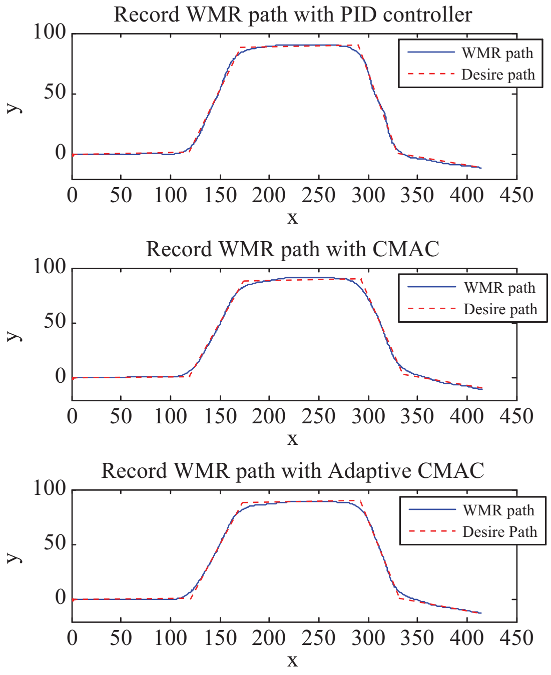


Fig. 30. Recorded WMR path.

than PID controller, and the adaptive PID-CMAC was superior to the conventional CMAC.

V. CONCLUSION

Based on image processing and CMAC, this study proposes an intelligent control scheme for application in WMR path tracking control. The proposed scheme involves applying the HSI color space and Hough transform. An image converted using the Hough transform can be used to locate a desired path, calculate the movement speed, and determine angle of a WMR with a CMAC. Under test conditions, the WMR successfully tracked an irregular path. This study confirmed that the proposed CMAC performs effectively and accurately in path tracking control applications.

REFERENCES

1. Ando, Y. and Yuta, S., “Following a wall by an autonomous mobile robot with a sonar-ring,” *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 4, pp. 2599-2606 (1995).
2. Åström, K. J. and Wittenmark, B., *Adaptive Control*, Pearson Education Taiwan Ltd. (2006).
3. Bunks, C., *The RGB Color Space*, http://gimp-savvy.com/BOOK/index.html?node50.html.
4. Canny, J. F., “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, pp. 679-698 (1986).
5. Chen, H. S. and Juang, J. G., “Path planning and parking control of a wheeled mobile robot,” *Proceedings of National Symposium on System Science and Engineering*, P0473 (2008).
6. Chen, Y. S. and Juang, J. G., “Intelligent obstacle avoidance strategy for wheeled mobile robot,” *Proceedings of ICROS-SICE International Joint Conference*, pp. 3199-3204 (2009).
7. Chiang, C. T. and Chiang, T. S., “A converged recurrent structure for CMAC_GBF and S_CMAC_GBF,” *IEEE International Symposium on Industrial Electronics*, pp. 1876-1881 (2007).
8. Chung, Y., Park, C., and Harashima, F., “A position control differential drive wheeled mobile robot,” *IEEE Transactions on Industrial Electronics*, Vol. 48, No. 4, pp. 853-863 (2001).
9. Gonzalez, R. C., Woods, R. E., and Eddins, S. L., *Digital Image Processing using Matlab*, Pearson Prentice Hall, Upper Saddle River, NJ (2004).
10. Hsu, F. Y. and Fu, L. C., “Intelligent robot deburring using adaptive fuzzy hybrid position/force control,” *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 4, pp. 325-335 (2000).
11. Juang, J. G., Yu, C. L., Lin, C. M., Yeh, R. G., and Rudas, I. J., “Real-time image recognition and path tracking to wheeled mobile robot for taking an elevator,” *Acta Polytechnica Hungarica*, Vol. 10, No. 6, pp. 5-23 (2013).
12. Lee, T. H., Lam, H. K., Leung, F. H. F., and Tam, P. K. S., “A practical fuzzy logic controller for the path tracking of wheeled mobile robots,” *IEEE Control Systems Magazine*, Vol. 23, pp. 60-65 (2003).
13. Lin, C. M., Peng, Y. F., and Hsu, C. F., “Robust cerebellar model articulation controller design for unknown nonlinear systems,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 51, No. 7, pp. 354-358 (2004).
14. Liu, J. N. K., Wang, M., and Feng, B., “iBotGuard: an internet-based intelligent robot security system using invariant face recognition against intruder,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 35, No. 1, pp. 97-105 (2005).
15. McAndrew, A., Wang, J. H., and Tseng, C. S., *Introduction To Digital Image Processing with MATLAB Asia Edition*, Cengage Learning Asia Pte Ltd., Taipei (2010).
16. MSI Idrobot, http://tw.msi.com/index.php.
17. Shieh, M. Y., Chan, Y. H., Cheng, C. P., and Hsieh, J. C., “Two-stage fuzzy navigation control of a vision-based intelligent shopping service robot,” *30th Annual Conference of IEEE on Industrial Electronics Society*, Vol. 1, pp. 430-435 (2004).
18. Sim, K. B., Byun, K. S., and Harashima, F., “Internet-based teleoperation of an intelligent robot with optimal two-layer fuzzy controller,” *IEEE Transaction on Industrial Electronics*, Vol. 53, No. 4, pp. 1362-1372 (2006).
19. VIVOTEK, *User’s Manual, Pan/Tilt/Zoom Network Camera PZ6114/6124* (2008).
20. Wang, W. H. and Juang, J. G., “Application of localization system to WMR path planning and parking control,” *IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 1677-1682 (2009).
21. Yang, T. C. and Juang, J. G., “Aircraft landing control based on adaptive CMAC,” *IEEE International Conference on Advanced Intelligent Mechatronics*, pp. 791-796 (2009).
22. Yu, M. S., Wu, H., and Lin, H. Y., “A visual surveillance system for mobile robot using omnidirectional and PTZ cameras,” *Proceedings of SICE Annual Conference*, pp. 37-42 (2010).
23. Zhan, J. J., Wu, C. H., and Juang, J. G., “Application of image process and distance computation to WMR obstacle avoidance and parking control,” *IEEE Conference on Industrial Electronics and Application*, pp. 1264-1269 (2010).