SICE Annual Conference 2013
September 14-17, 2013, Nagoya University, Nagoya, Japan

A Practical Fuzzy Controller with Q-learning Approach for the Path Tracking of a Walking-aid Robot

Chun-Tse Lin[1], Hsin-Han Chiang[1], and Tsu-Tian Lee[2*]

[1]Department of Electrical Engineering, Fu Jen Catholic University, New Taipei City, Taiwan, ROC
(Tel : +886-2-2905-3799; E-mail: hsinhan@ee.fju.edu.tw)
[2*]Department of Electrical Engineering, Chun Yuan Christian University, Chung Li, Taiwan ROC
(Tel : +886-3-456-3171; E-mail: ttlee@cycu.edu.tw)

Abstract: This study tackles the path tracking problem of a prototype walking-aid (WAid) robot which features the human-robot interactive navigation. A practical fuzzy controller is proposed for the path tracking control under reinforcement learning ability. The inputs to the designed fuzzy controller are the error distance and the error angle between the current and the desired position and orientation, respectively. The controller outputs are the voltages applied to the left- and right-wheel motors. A heuristic fuzzy control with the Sugeno-type rules is then designed based on a model-free approach. The consequent part of each fuzzy control rule is designed with the aid of Q-learning approach. The design approach of the controller is presented in detail, and effectiveness of the controller is demonstrated by hardware implementation and experimental results under human-robot interaction environment. The results also show that the proposed path tracking control methods can be easily applied in various wheeled mobile robots.

Keywords: Fuzzy controller, path-tracking, fuzzy Q-learning, walking-aid robots.

1. INTRODUCTION

Wheeled mobile robots as a multidisciplinary field in engineering have been a kind of attractive platform, which integrates mechanical, electronic, control, computing, and communication systems to create a heterogeneous system for research and industrial applications [1-3]. This field so far has evolved quickly over the last decade because of the enormous increase in computing power and the considerable variety of sensors and actuators. The design problems of wheeled mobile robots can be classified into two categories that include the construction of mechanical structure as well as the development of an efficient control algorithm.

During the last decades, the motion control of robots, such as path tracking, parking, navigation, goal seeking, and wall following, has attracted the great amount of the attention of researchers [4-6]. To tackle the abovementioned motion control problems with regard to several varieties of robot configuration, some challenges have to be addressed. First, owing to the nonlinear dynamic and nonholonomic characteristics of the wheeled mobile robots, the controller design will be a challenge due to lack of systematic and simple controller design methodology for nonlinear systems. More details about the modeling of wheeled mobile robots can be found in [7]; however, the obtained model may be not very applicable because of the parameter uncertainties in practical robots. Moreover, even if a model-based nonlinear controller can be designed, this controller may be complicated and not easy for implementation in the source-constrained computing platform. In view of these difficulties mentioned above, a fuzzy controller is often adopted to control a mobile robot because of its capability of making inference even under uncertainty [8]. Although fuzzy logic approaches have been investigated for a great amount of robot motion control studies, the problem of finding a number of proper rules for a fuzzy controller still remains open. Besides, the selection of a set of linguistic rules of a fuzzy controller relies on a substantial amount of heuristic observation so that the expert knowledge can be incorporated into the controller design. Hence, it is complicated and time-consuming to implement the fuzzy controller from the collected data of a robot under various situations.

In some studies (e.g., [9, 10]), neurofuzzy-based approaches are proposed to create fuzzy if-then rules by using supervised learning algorithm. However, precise training data for supervised learning needs to be collected in advance. Recently, the reinforcement learning-based optimized fuzzy control design is proposed for robot wall-following control [11]. In this work, no preassignment of fuzzy rules is required and all fuzzy rules are generated online from the real input-output data of the robot. In that approach, the antecedent and consequent part use online aligned interval type-2 fuzzy clustering and Q-value-aid ant colony optimization, respectively. It is suitable for fewer system inputs of the robot and simple environment, since the more complex the environment, the more sensing inputs required that raises computation load to construct the fuzzy system.

Nowadays, motion control of assistive mobile robot platform attracts much attention in order to provide the mobility support with the navigation and collision-free assistance whenever necessary. One major application of wheeled mobile robot is the assisted-walking facility, known as the walking-aid (WAid) robot [12, 13]. The WAid robot can provide both support and navigation assistance while reducing the need for supervision so that the independence and well-being of walking/visual impaired and elderly can be increased while reducing the cost of care. To this end, this paper proposes an effective fuzzy control which tackles the path-tracking problem of the WAid robot. The robot dynamics is

PR0001/13/0000-0888 ¥400 © 2013

introduced in Section 2. Based on the real-time locations of the user and the robot, a desired path can be generated. The fuzzy tracking controller which combines the human knowledge and Q-learning-based on-line optimization is presented in Section 3. The control gains in the consequent part of the fuzzy controller can be automatically tuned without need of precise input-output training pairs. Our robot platform is introduced in Section 4, and the experimental results depict the effectiveness and efficiency of our system. Finally, the conclusion is given in Section 5.

2. DEVELOPMENT OF ROBOT DYNAMICS

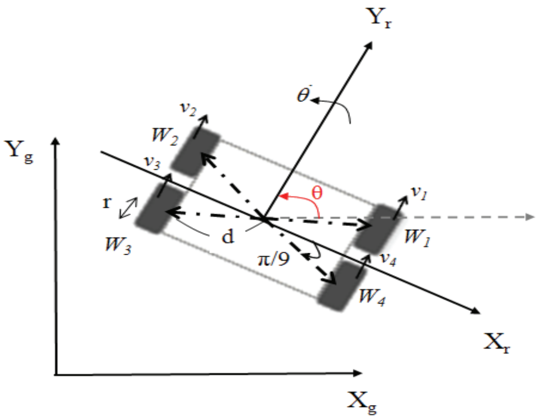

Fig.1 Kinematic model of the WAid robot.

The kinematic model with the wheeled configuration of the WAid is shown in Fig. 1. In this model, we define the horizontal of velocity as X-axis (X_r), front motion direction as Y-axis (Y_r), and rotation motion direction around the center of the robot. Thus, from the speed of each wheel, the velocity of the center of the robot in the world frame can be transformed as follows:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} \cos(\theta+\pi/9) & \sin(\theta+\pi/9) & d \\ -\cos(\theta-\pi/9) & -\sin(\theta-\pi/9) & d \\ -\cos(\theta+\pi/9) & -\sin(\theta+\pi/9) & d \\ \cos(\theta-\pi/9) & \sin(\theta-\pi/9) & d \end{bmatrix} \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = T_v(\theta) \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} \quad (1)$$

where v_i, i = 1, …, 4, are the velocities of each wheel, respectively; θ is the orientation angle of the robot; $\dot{\theta}$ is the rotation speed of the robot; d is distance between the wheel center and the center point of the robot; $\dot{x}_g$, $\dot{y}_g$ denote the robot velocity in the X and Y components of the world frame; $T_v(\theta)$ denotes the transformation matrix which varies with the orientation angle of the robot. The world coordinate frame can then be transformed to the robotic coordinate frame, and the robot dynamics can be obtained as follows:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \sin\theta & -\cos\theta & 0 \\ \cos\theta & \sin\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = T_r(\theta) \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} \quad (2)$$

where $\dot{x}_r$, $\dot{y}_r$ are the robot moving speed in the X and Y components of the robot frame. Hence, the inverse kinematic model that represents the dynamics between the robot center and each wheel can be shown as

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = T_v(\theta) \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = T_v(\theta) \cdot T_r^{-1}(\theta) \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} = M_R \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} \quad (3)$$

where M_R is the inverse kinematic transformation matrix given as follows:

$$M_R = T_v(\theta) \cdot T_r^{-1}(\theta) = \begin{bmatrix} -\sin(\pi/9) & \cos(\pi/9) & d \\ -\sin(\pi/9) & -\cos(\pi/9) & d \\ \sin(\pi/9) & -\cos(\pi/9) & d \\ \sin(\pi/9) & \cos(\pi/9) & d \end{bmatrix} \quad (4)$$

From (3), the speed of each wheel can be calculated by using the moving velocity of the center of the robot. Conversely, the robot motion dynamics can be yielded as

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \end{bmatrix} = rM_R^{+} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}, M_R^{+} = (M_R^{T} M_R)^{-1} M_R^{T} \quad (5)$$

where r is the radius of each wheel, and ω_i , i = 1, …, 4, denotes the rotation speed of each wheel. The current position and orientation of WAid can be estimated by using (5) and with the measurements of wheel encoders.

The robot plant model in (5) is the basis for designing the path tracking controller of the WAid. The control objective is to drive the WAid of output coordinates x_r and y_r , to follow a desired path in a flat space by feeding appropriate rotation speeds to the wheel motors. However, the model of (5) may not be very useful for controller designing due to the parameter uncertainties in practical robots, and also makes it difficult to realize the designed controller without the aid of reliable mathematical model. Besides, in view of the environmental disturbances, the readings of the wheel speeds subject to tolerances. As a result, errors of input signals are inevitable.

3. CONTROL DESIGN

3.1 Fuzzy control design

In this section, the fuzzy controller design is proposed. Initially, two inputs for the fuzzy controller are defined. The first input is the error distance between the current and the desired positions of the path and given as follows:

$$d_e = \sqrt{(x_t - x_r)^2 + (y_t - y_r)^2} \quad (6)$$

where x_t and y_t are the reference positions. The other input is the error angle between the current and the desired orientation. Based on the desired positions x_t and y_t, the desired heading angle can be calculated by

$$\theta_t = \tan^{-1}\frac{x_t - x_r}{y_t - y_r} \tag{7}$$

The error orientation can then be obtained as

$$\beta_e = \theta_t - \theta \tag{8}$$

According to the expert knowledge, the tracking controller needs to control the translational and rotational motions of the robot. If $\beta_e = 0$, the controller should only result in a translational motion; that is, $\omega_1 = \omega_2 = \omega_3 = \omega_4$. On the other hand, if $d_e = 0$, the controller should result in a rotational motion only; that is, $\omega_2 = \omega_3 = -\omega_1 = -\omega_4$.

To avoid the using of complicated plant model of robot, our system incorporates expert knowledge into the controller design process using linguistic rules. A fuzzy controller with the Sugeno-type fuzzy rules is developed to control the WAid:

Rule 1: IF d_e is small and $|\beta_e|$ is small, THEN

$$u_1 = \begin{bmatrix}\omega_{2,3}\\ \omega_{1,4}\end{bmatrix} = \begin{bmatrix}k_{d1}d_e + k_{\beta 1}\beta_e\\ k_{d1}d_e - k_{\beta 1}\beta_e\end{bmatrix}$$

Rule 2: IF d_e is small and $|\beta_e|$ is large, THEN

$$u_2 = \begin{bmatrix}\omega_{2,3}\\ \omega_{1,4}\end{bmatrix} = \begin{bmatrix}k_{d2}d_e + k_{\beta 2}\beta_e\\ k_{d2}d_e - k_{\beta 2}\beta_e\end{bmatrix}$$

Rule 3: IF d_e is large and $|\beta_e|$ is small, THEN

$$u_3 = \begin{bmatrix}\omega_{2,3}\\ \omega_{1,4}\end{bmatrix} = \begin{bmatrix}k_{d3}d_e + k_{\beta 3}\beta_e\\ k_{d3}d_e - k_{\beta 3}\beta_e\end{bmatrix}$$

Rule 4: IF d_e is large and $|\beta_e|$ is large, THEN

$$u_4 = \begin{bmatrix}\omega_{2,3}\\ \omega_{1,4}\end{bmatrix} = \begin{bmatrix}k_{d4}d_e + k_{\beta 4}\beta_e\\ k_{d4}d_e - k_{\beta 4}\beta_e\end{bmatrix} \tag{9}$$

where the control efforts $\omega_{2,3}$ and $\omega_{1,4}$ denote the rotation speeds applied to the left- and right-wheel motors, respectively; k_{di} and $k_{\beta i}$ are the control gains for the error distance and the error angle in i-th rule, respectively. Note that in the structure of (9), the velocity and acceleration data are not used as inputs of the controller; hence, the effect of error accumulation can be reduced during the path tracking.

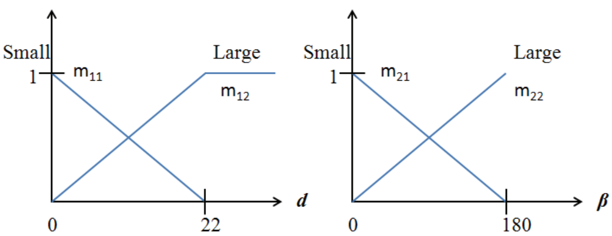

Fig. 2 Input membership functions.

The membership functions for the fuzzy controller are designed as

$$m_{11}(d_e) = \min(\,\max(\,0\,, \frac{d_e - 22}{-22}\,)\,, 1\,) \tag{10}$$

$$m_{12}(d_e) = 1 - m_{11}(d_e) \tag{11}$$

$$m_{21}(|\beta_e|) = \frac{-|\beta_e| + 180}{180} \tag{12}$$

$$m_{22}(|\beta_e|) = 1 - m_{21}(|\beta_e|) \tag{13}$$

The operation max(.) and min(.) in (10) are used to restrict the strength value of m11 within the range [0, 22] even when the value of d_e is larger than 22. From (12), the strength value of m21 will lie between zero and one as the maximum value of $|\beta_e|$ is 180. Accordingly, the input membership functions are shown in Fig. 2. The membership values of rules 1 to 4, respectively, are computed as follows:

$$w_1 = m_{11}(d_e) \times m_{21}(|\beta_e|) \tag{14}$$

$$w_2 = m_{11}(d_e) \times m_{22}(|\beta_e|) \tag{15}$$

$$w_3 = m_{12}(d_e) \times m_{21}(|\beta_e|) \tag{16}$$

$$w_4 = m_{12}(d_e) \times m_{22}(|\beta_e|) \tag{17}$$

The output of the fuzzy controller is given by

$$u = \sum_{i=1}^{4} w_i u_i \Big/ \sum_{i=1}^{4} w_i \tag{18}$$

The developed controller shows the adaptive capability with respect to different operational regions of the robot. Obviously, this algorithm with only 4 useful rules has benefits over rule bases with hundreds or even thousands of fuzzy rules.

3.2 Q-learning for the consequent part of the fuzzy controller

In the previous section, the control gains k_{di} and $k_{\beta i}$ are required in the consequent action for each rule. Although their values can be determined based on trial and error and manually tuned according to a tradeoff between the speed and the stability of robot, the advantages of the fuzzy controller with an adaptive capability may not be retained in a desired level for path tracking response. Besides, the tuning effort for a total of 2^4 combinations of consequent actions heavily increases the control design burden. This paper proposes the use of Q-learning to automatically determine the one combination which satisfies the quick response in tracking performance. This section briefly introduces the basic concept of Q-learning as follows.

As being a reinforcement learning method, the learning agent of Q-learning can on-line observe the present system state $x(t)$ and execute an action $a(t)$ according to the returned evaluation in the current stage. The evaluation function, known as the Q-function, can be used to estimate the discounted accumulative reinforcement for executing actions from given states [14]. The Q-function can provide the mapping from the trained data among state-action pairs, and calculate the output Q-value, denoted as $Q(x, a)$, with the adaptation law as

$$\begin{aligned} &Q(x(t), a(t)) \leftarrow Q(x(t), a(t)) \\ &\quad + \alpha[r(t+1) + \gamma Q^*(x(t+1)) - Q(x(t), a(t))] \\ &\qquad 0 < \alpha \le 1, 0 \le \gamma \le 1 \end{aligned} \tag{19}$$

where $x(t+1)$ is the next state of $x(t)$ after the execution of $a(t)$; $r(t)$ is the reinforcement from the environment; α is a constant step-size learning rate and determines the updating speed of the Q-function; γ is a discount factor which determines the present value of future rewards; $Q^*(x(t+1))$ represents the best estimated Q-value by the agent at state $x(t+1)$ and is defined by

$$Q^*(x(t+1)) = \max_{b \in U(x(t+1))} Q(x(t+1), b) \tag{20}$$

where $U(x(t+1))$ is the set of possible actions in state $x(t+1)$. In the algorithm of Q-learning, the objective will take future rewards into account more strongly while γ approaches one. In our design of consequent-part learning, each candidate control action u_h in i-th rule is assigned with a Q-value, denoted as q_{ih}, which dominates the selection of the control gains in the consequent part of the fuzzy controller (9).

Based on the fuzzy rule-structure of (9), the control gains in the consequent part are generated on-line using Q-learning algorithm. At each control time step, Q-values are adapted. For each rule i, let $\hat{h}$ = {1, 2} be the subscript of the consequent action chosen by Q-learning. The fuzzy control output is given by

$$a(x(t)) = \sum_{i=1}^{4}\left(w_i u_{\hat{h}}\right) / \sum_{i=1}^{4} w_i \tag{21}$$

The actual Q-value for controller output a(x(t)) is calculated by

$$Q(x(t), a(x(t))) = \sum_{i=1}^{4}\left(w_i q_{i\hat{h}(t)}\right) / \sum_{i=1}^{4} w_i \tag{22}$$

Now the best Q-value at this state corresponds to the quality of the best action inferred from the fuzzy controller. According to (22), the maximum estimated greedy policy can be employed finding $q_{i\hat{h}(t+1)} = q_{ih^*(t)}$, and the expected Q-value Q^* at state $x(t+1)$ is obtained as

$$Q^*(x(t+1)) = \sum_{i=1}^{4}\left(w_i q_{ih^*(t)}\right) / \sum_{i=1}^{4} w_i \tag{23}$$

From (19), the temporal difference error of the output $Q(x(t), a(x(t)))$ is then given as

$$\Delta Q = r(t+1) + \gamma Q^*(x(t+1)) - Q(x(t), a(x(t))) \tag{24}$$

where the reinforcement signal is set as:

$$r(t+1) = \begin{cases} 1, \text{if } d_e < 1 \text{ and } |\beta_e| < 5 \\ \text{-1}, \text{if } d_e > 5 \text{ or } |\beta_e| > 15 \\ 0, \quad \text{otherwise} \end{cases} \tag{25}$$

At each time, a reinforcement signal $r(t+1)=0$ is received. The robot tracks a desired path until it moves outside the constraint of error distance or error orientation, the control has to stop and a failure reinforcement signal $r(t+1)=-1$ is received. A reward reinforcement signal $r(t+1)=1$ is given while the tracking errors are reduced to a desired range.

At each step time t, the Q-value in each rule is updated as follows:

$$q_{i\hat{h}}(t+1) = q_{i\hat{h}}(t) + \rho \Delta q_{i\hat{h}}(t) \tag{26}$$

where i = 1, …, 4, ρ is the learning rate. By and ordinary gradient descent and the error signal ΔQ, we obtain

$$\Delta q_{i\hat{h}}(t) = \Delta Q \cdot w_i / \sum_{i=1}^{4} w_i \tag{27}$$

In this study, the eligibility trace technique [15] is used to accelerate the learning speed so that (26) and (27) can be modified as follows:

$$q_{ih}(t+1) = q_{ih}(t) + \rho \Delta q_{ih}(t) \tag{28}$$

where h = 1, 2. By defining the eligibility $e_{ih}(t)$ of an action as

$$e_{ih}(t) = \begin{cases} \lambda e_{ih}(t-1) + w_i, \text{if } h = \hat{h} \\ \lambda e_{ih}(t-1), \quad \text{otherwise} \end{cases}, \quad 0 \le \lambda \le 1. \tag{29}$$

As a result, the updating law of (27) becomes

$$\Delta q_{ih}(t) = e_{ih}(t) \Delta Q \cdot w_i / \sum_{i=1}^{4} w_i \tag{30}$$

Since the eligibility trace considers the states and actions of the conclusion pair, more pairs near the receipt of a reward or a punishment cause more affects to the corresponding quality. In this paper, all of the coefficients in Q-learning algorithm are set the same to the study [14].

4. EXPERIMENT

4.1 Robot platform setup

Figure 3 shows the photo of the prototype walking-aid robot. The robot system has a suitable size and weight for a home environment. The on-board embedded system integrates a 400 MHz real-time processor running the VxWroks real-time operating system (RTOS), a reconfigurable FPGA, and 128 MB of DRAM. The laser scanner is mounted in front at 45 cm from the floor pointing horizontally. The laser is connected via serial (RS232) interface, and measures environment information ahead. The ultrasonic sensor is installed at the back of the robot, connected via digital I/O port, and measures the relative distance to a user. In this study, the laser scanner and the ultrasonic sensor are used to monitor the operation status in the path-tracking process. The robot wheels are driven by DC motors, which are connected via digital I/O port. The motor encoder information is available for dead reckoning. A wireless module with Ethernet port is available for communicating with the home server. Two batteries provide a 12-V power supply to the robot.

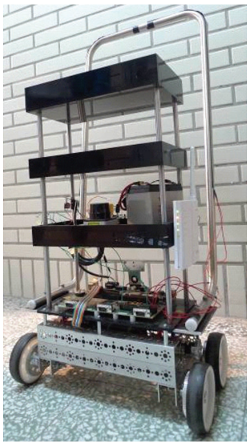

Fig.3 Photo of the robot platform (WAid).

4.2 Experimental results

The experiment of the WAid robot has two parts. The first part is curved path tracking control, and we compare the performance of the proposed fuzzy controller, which includes Q-learning-aid and manually tuned, with the conventional PID controller. The second part is path-tracking control under the ramp sidewalk with right angle corner.

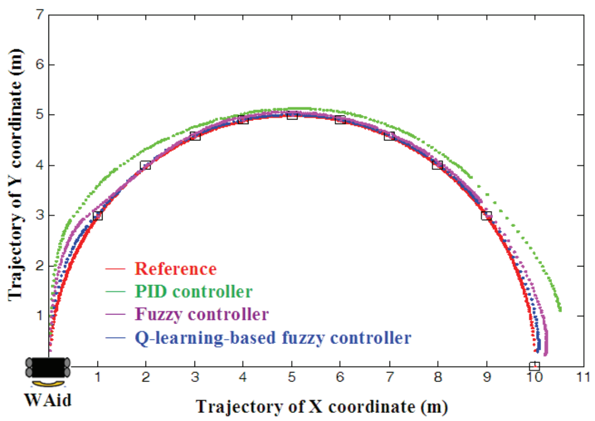

Fig. 4 Trajectory plot of the WAid robot for the curved line tracking.

In the first part, a reference curved trajectory starting from the position (0, 0) to the target position (10, 0) is specified, as illustrated in Fig. 4. The robot starts at the position (0, 0) with the initial heading angle about 90^o vertical to X-axis. In practice, the reference path is generated as discrete points depending on the sampling interval. That is, every generated point is a new destination of the WAid robot after the sampling period. Figure 4 shows the plot of the reference and the actual trajectories. For comparison purposes, a PID-controller and the fuzzy controller without Q-learning aid are employed to perform the same task. The control gains of the PID and the fuzzy controller are obtained by trial and error. If the gains are set too large, the tracking stability of robot is easily deteriorated. From the results of Fig. 4, the tracking performance of the Q-learning-based fuzzy controller is superior to other approaches. Besides, its action time of the WAid robot from starting to send commands to arriving at the destination is about 20 s which is almost twice as fast as the fuzzy controller without Q-learning aid, and thrice as fast as the PID controller.

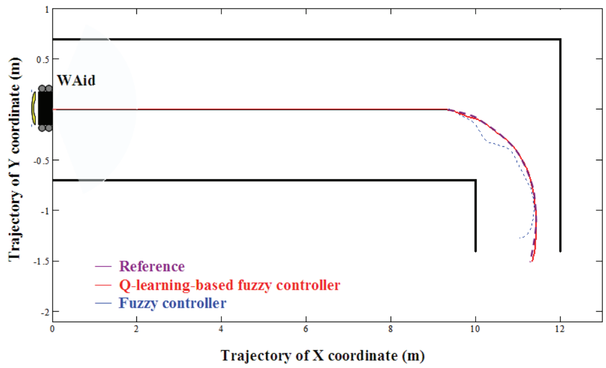

Fig. 5 Path tracking performance in a ramp sidewalk.

In the second part, the path-tracking performance is evaluated under more complicated environment. The starting position of the robot is set to (0, 0). For reference path generation, the robot is initially controlled by human operation via the use of joystick to navigate a designated route five times with the aim of obtaining the consistent trajectory profile. Next, we test path tracking performance for the proposed fuzzy controller. Similarly, several sampled points are one-by-one generated from the reference path. The comparison result of the path tracking is shown in Fig. 5. The total tracking period of the Q-learning-based fuzzy controller is about 40 s, whereas the manually tuned fuzzy controller needs about 50 s. Besides, the better tracking accuracy can be achieved by the fuzzy controller with the aid of Q-learning algorithm.

Some images of the real-time environment and robot testing are shown in Fig. 6. The subject's eyes are blanked to imitate the blind situation. With the assistance of the WAid robot, the user can successfully complete the goal tracking on the ramp sidewalk without collision. From this achievement, the proposed fuzzy controller can also be extended to the user-friendly assistance such as call-to-come service [6]. Once the user's location is accessed, the WAid robot can pursuit the planned path and come to the user autonomously.

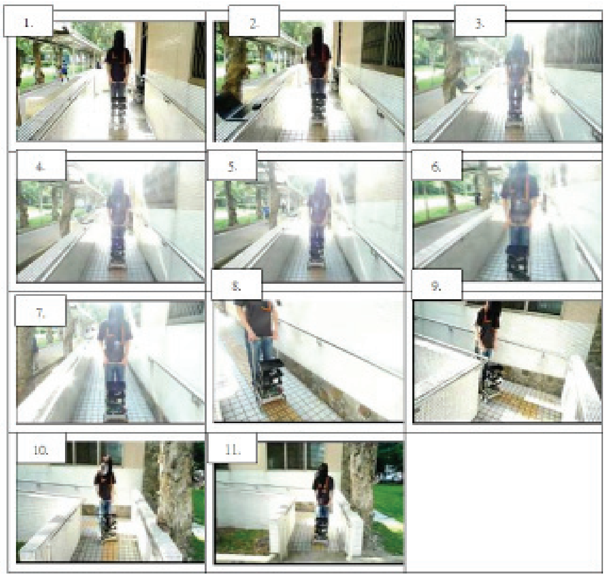

Fig. 6 Robot tracking control for walking-aid field trials.

5. CONCLUSION

In this paper, a practical fuzzy controller is proposed to control the WAid robot in the task of path tracking. For the improvement of tracking accuracy and efficiency, Q-learning algorithm is applied to automatically generate the control gains of the fuzzy controller, and the design effort based on a model-free approach can be mitigated. With the Q-value-aid, the consequent part learning of the fuzzy controller is shown to be effective and efficient in comparison with a fine-tuned PID controller and the fuzzy controller based only on human knowledge. From the hardware experimental results, the response time of the proposed fuzzy controller is faster while remaining the good tracking control performance.

REFERENCES

[1] G. Song, H. Wang, J. Zhang, and T. Meng, "Automatic docking system for recharging home surveillance robots", *IEEE Trans. Consumer Electronics*, vol. 57, no. 2, pp. 428-435, May, 2011.
[2] D. Nakhaeinia, S. H. Tang, S. B. N. Mohd, and O. Motlagh, "A review of control architecture for autonomous navigation of mobile robots," *Int. J. Physical Sciences*, vol. 6, no. 2, pp. 169-174, 2011.
[3] O. Chuy, Y. Hirata, Z. Wang, and K. Kosuge, "A control approach based on passive behavior to enhance user interaction," *IEEE Trans. on Robotics,* vol. 23, no. 5, pp. 899-908, 2007.
[4] S. S. Ge and Y. J. Cui, "New potential function for mobile robot path planning," *IEEE Trans. Robot. Autom*., vol. 16, no. 5, pp. 615-620, Oct. 2000.
[5] H. Yu, M. Spenko and S. Dubowsky, "An adaptive shared control system for an intelligent mobility aid for the elderly," *Auton. Robots*, vol. 15, no. 1, pp. 53-66, 2003.
[6] Y. Oh, J. Yoon, J. Park, M. Kim and H. Kim, "A name recognition based call-and-come service for home robots," *IEEE Trans. on Consumer Electronics*, vol. 54, no. 2, pp. 247-253, 2008.
[7] J. M. Yang and J. H. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots", *IEEE Trans. Robot. Automat.*, vol. 15, pp. 578-587, 1999.
[8] M. A. Olivares and J. A. F. Madrigal, "Fuzzy logic user adaptive navigation control system for mobile robots in unknown environments," in *Proc. IEEE Int. Symp. on Intell. Signal Processing*, pp. 1-6, 2007.
[9] A. Zhu and S. X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments", *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and reviews*, vol. 37, no. 4, pp. 610-621, 2007.
[10] H. Erdem, "Application of neuro-fuzzy controller for sumo robot control", *Expert Systems with Application*, vol. 38, pp. 9752-9760, 2011.
[11] C. F. Juang and C. H. Hsu, "Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control", *IEEE Trans. Industrial Electronics*, vol. 56, no. 10, pp. 3931-3940, 2009.
[12] O. Chuy, Y. Hirata and K. Kosuge, "A new control approach for a robotic walking support system in adapting user characteristics," *IEEE Trans. on Systems, Man, and Cybernetics, Part-C,* vol. 36, no. 6, pp. 725-733, 2006.
[13] Y. Hirta, A. Hara, and K. Kosuge, "Motion control of passive intelligent walker using servo brakes", *IEEE Trans. Robotics*, vol. 23, no. 5, pp. 981-990, 2007.
[14] P. Y. Glorennec and L. Jouffe, "Fuzzy Q-learning", in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 1997, pp. 659–662.
[15] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. Cambridge,MA: MIT Press, 1998.