International Journal of Computational Intelligence Systems, Vol. 6, No. 3 (May, 2013), 503-517

An Algorithm Based on Neuro-Fuzzy Controller Implemented in A Smart Clothing System For Obstacle Avoidance

Senem Kursun Bahadir
ITU, Textile Engineering Department 34437 Istanbul, Turkey
Univ. Lille North of France, ENSAIT, GEMTEX F-59100, Roubaix, France
E-mail: kursuns@itu.edu.tr
www.itu.edu.tr

Sebastien Thomassey
Univ. Lille North of France, ENSAIT, GEMTEX F-59100, Roubaix, France
E-mail: sebastien.thomassey@ensait.fr
www.ensait.fr

Vladan Koncar
Univ. Lille North of France, ENSAIT, GEMTEX F-59100, Roubaix, France
E-mail: vladan.koncar@ensait.fr
www.ensait.fr

Fatma Kalaoglu
ITU, Textile Engineering Department 34437 Istanbul, Turkey
E-mail: kalaoglu@itu.edu.tr
www.itu.edu.tr

Received 7 February 2012

Accepted 6 January 2013

Abstract

In this study, to overcome navigation concerns of visually impaired people, an algorithm based on neuro-fuzzy controller composed of multi-layer fuzzy inference systems (FIS) for obstacle avoidance was developed and it was implemented in a smart clothing system. The success of the proposed algorithm was tested in real environment and it was compared with one layer FIS. Results showed that the proposed algorithm is capable of guiding user to a right orientation and it presented better results than the one layer FIS.

Keywords: Obstacle avoidance, Visually impaired, Navigation, Smart clothing, Neuro-fuzzy controller, Fuzzy Inference System.

1. Introduction

The ability to navigate visually impaired person through an environment cluttered with obstacles is a crucial issue. Navigation towards a target is a complex task and an important research field especially in robotic applications. The real-time obstacle avoidance algorithm is one of the key issues for mobile robots as well. The general theory for mobile robotic navigation is based on such principles:

First; the robot can perceive the surroundings by sensors mounted on it like cameras, sonars, laser range finders, GPS etc. Then, it is able to plan its operations based on the artificial intelligence model developed for navigation and obstacle avoidance task.

In the literature, a large number of algorithmic approaches were used in order to plan mobile robot motion such as grid method[1], vector field histogram method[2], potential field method[3], path planning[4], geometry based approach, pattern generation method[5], switching control approach[6-8], self localization[9], soft computing based approaches like fuzzy logic, neural network, genetic algorithm and their different combinations[10].

Fuzzy logic is easily used when a mathematical model of the process is difficult to be proved or implemented in a real-time operation[11]. In recent years, fuzzy logic, neural network and genetic algorithm based approaches have been successfully applied to control mobile robots.

Ragaruman et al.[12] proposed a fuzzy logic based model for navigation of mobile robots in indoor environment. Guo et al.[13] developed an algorithm by using fuzzy logic to control the lower limbs rehabilitation robot with the known environment information. Jincong et al.[10] introduced the design of an intelligent four-wheel obstacle avoidance robot based on fuzzy control. In another study, different fuzzy logic controllers with different membership functions in order to navigate mobile robots have been discussed[14]. Park and Zhang[15] used a dual fuzzy logic approach for navigation of mobile robot: the first controller was designed to control target steering while the second one to follow the edge of obstacles. Similarly, Chen and Juang[11] designed two model based on fuzzy logic controllers in order to control wheeled mobile robot The first model was set up to avoid short distance obstacles while the second one was for target seeking. Farooq et al.[16] designed a fuzzy logic based hurdle avoidance controller for mobile robot navigation in noisy and uncertain environments. Maaref and Barret [17] presented a study about the problem of navigating mobile robot either in an unknown environment or in a partially known one. A navigation method based on fuzzy inference was proposed for avoiding convex and concave obstacles. In most of the fuzzy logic controllers, the performance of the controller depends on the selection of membership functions and fuzzy if-then rules. Since the if-then rules were designed by human experts, it is hard to choose and implement correct rules in the controller[18-23.] Therefore, there were some attempts in order to extract rules automatically. Hui and Pratihar[24] used genetic algorithm to extract rules for fuzzy controller, thus they developed an algorithm based on combination of genetic and fuzzy approaches to avoid obstacles. Moreover, Liu and Liu et al.[25-26] adjusted the rules of fuzzy obstacle avoidance controller of autonomous mobile robot by using genetic algorithm. For the mentioned problem, some researchers have focused on using neural network approach to control the mobile robot. For instance, Szemes et al.[27] applied the observation of human walking behavior to train fuzzy neural networks (FNN). The trained FNNs were applied to approximate the obstacle avoidance behavior of human walking as well as to control the mobile robot in a human-robot shared environment, similarly Mahyuddin et al. and Nasuriddin[28-29] designed a neuro-fuzzy algorithm which is able to control the operations such as sense, map, plan and act. In their system, they used neuro-fuzzy approach in order to modify and extract new rules from a properly training. He et al.[30] used fuzzy neural network method based on the Takagi-Sugeno information fusion arithmetic to avoid obstacles. First, the information get by sensors was classified and fused. Then the fused results were considered as the inputs of fuzzy neural network. In another study, the neural network approach was combined with GPS. In that system, a radial basis function network (RBFN) based on neural network was used to map the GPS data into the robot coordinates and then, trained data was combined with sonar based navigation system of the mobile robot[31]. Hui and Pratihar[32] developed various algorithms based on genetic-fuzzy, genetic neural and potential field method (PFM) approaches and compared them as well. They found that soft computing based approaches (genetic fuzzy and genetic neural) were more adaptive and robust compared to the PFM.

According to articles published until now, a great number of different obstacle avoidance algorithms for mobile robots have been developed for indoor and outdoor environments. However, there is no obstacle avoidance algorithm developed for visually impaired people through an integrated system that consists of sonar sensors mounted on a garment. In the present study, in order to guide visually impaired person through an environment cluttered with obstacles, neuro-fuzzy logic based obstacle avoidance control algorithm was developed for smart clothing system. Generally, algorithms for obstacle avoidance are developed for mobile robots. Sensors are located on the robot body close to the ground and their support is rigid and stable. In the case of our algorithm, sensors (sonar) are located on the clothing that may be loose and in the chest and abdominal region of the body. Therefore, signals generated by sensors are disturbed, noisy and not often reliable. This is the reason, why our algorithm should be more robust and able to integrate disturbances in order to generate an optimal control. Moreover, possible obstacles are very different implying a large variety of sensor signals from four sensors functioning simultaneously. The reliability and robustness of proposed algorithm have been achieved by a network of fuzzy inference systems based on the rules. The original contribution of our method is based on those rules and on the global control system configuration making an expert system dedicated to obstacles avoidance in harsh environmental conditions.

For people without vision problems avoiding obstacles seems very simple, because they have a high definition image of the environment in a real time and large experience coming from childhood in avoiding various kinds of objects. For visually impaired people, this “simple” problem is very complex even if they are equipped with four ultrasonic sensors located on the garment and moving in all directions during the walking process. Signals generated from those sensors are very poor and also strongly disturbed (unreliable) comparing with the high definition image generated in the brain by eyes. This is the reason, why the architecture of the control system has to be complex (with three fuzzy controllers) and organized in multi layers in order to generate apparently simple decision: turn left, turn right, turn left or right or just go ahead. This novel architecture is explained in details in the following sections.

In Section II, the design of smart clothing system is introduced. In section III, navigation problem of visually impaired people as well as kinematic analysis of walking person is described. In Section IV, the structure of neuro-fuzzy control algorithm is explained. In Section V, the experimentation procedure of the proposed system is given. In Section VI, the success of developed algorithm and its comparison with one layer fuzzy controller is presented. Conclusions are included in Section VII.

2. Smart clothing system for visually impaired people

Smart clothing system including ultrasonic sensors, vibration motors, power supplies, and a microcontroller, is shown in Fig. 1. The working principle of the system is based on two main functions: sensing the surrounding environment as well as detection of obstacles via sensors and guiding user by actuators through a feedback process (control algorithm) interpreted in signal processing unit. In this system, four ultrasonic sensors were used to detect obstacles, eight vibration motors (four on the left and right) were used in order to guide user by recommending him/her turning direction and angle. The function of its circuitry is to digitize as well as transform analog signals acquired by sensors into vibration signals. It modulates analog signals into different levels of vibrations by identifying correlation between position of obstacle and required turning action (direction and angle) for user. The design of system aims at analysing signals acquired by the ultrasonic sensors and transforming them into different vibration intervals in the case of obstacles for guiding person with recommended turning action to avoid obstacle[33]. Indeed, in the literature the transformation of surrounding information into vibrating instructions is commonly chosen considering people especially with hearing and visual disabilities[34].

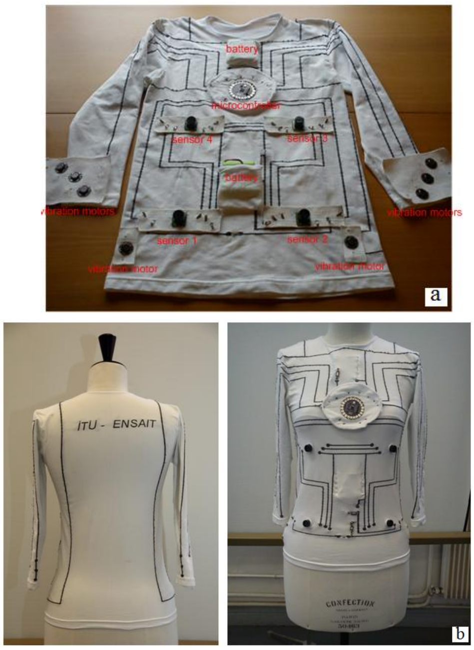

Fig.1. Developed interactive garment

3. Navigation problem of visually impaired people

3.1 *Kinematics analysis of walking person*

Assume that person position is $P_b=(x_b , y_b, \Theta_b)$, where (x_b , y_b) represents the coordinate of the person body and Θ_b represents his heading angle from the horizontal axis as seen in Fig. 2

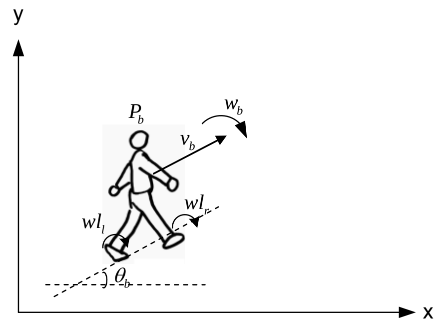

Fig. 2. Model of walking person in the coordinate system

In the figure, w_b and v_b are the angular and linear velocities of walking person's body, respectively. The angular velocity of person depends on both angular velocities of the left (wl_l) and right (wl_r) legs where it can be demonstrated as $w_b=(wl_l , wl_r)$

According to heading angular velocity w_b, the corresponding motion state of walking person can be summarized as noted in Table 1.

Table 1. Motion state of walking person

w_b	$w_b=0$	$w_b>0$	$w_b<0$
Motion state	Go straight	Turn right	Turn left

If $wl_l = -wl_r$, then the angular velocity is $w_b = 0$ which implies that there is no turning action: go straight. Thus, the desired state of motion can be obtained by changing wl_l and wl_r.

Motion state of the walking person can be shown as $M_b = (v_b , w_b)^T$ by using his linear v_b and angular w_b velocities. Thus, the first kinematic equation can be written as follows:

$$\dot{P}_b = \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{\theta}_b \end{bmatrix} = \begin{bmatrix} \cos\theta_b & 0 \\ \sin\theta_b & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_b \\ w_b \end{bmatrix} \quad \mathbf{(1)}$$

and the coordinate of moving person is

$$\begin{bmatrix} x_{b(i+1)} \\ y_{b(i+1)} \\ \theta_{b(i+1)} \end{bmatrix} = \begin{bmatrix} x_{b(i)} \\ y_{b(i)} \\ \theta_{b(i)} \end{bmatrix} + \begin{bmatrix} \cos\theta_b & 0 \\ \sin\theta_b & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_b \\ w_b \end{bmatrix} \times \Delta t \quad \mathbf{(2)}$$

where Δt is the sampled time, i is the current time index, and $i+1$ is the next time index.

Hence, according to above equations, the position of walking person can be estimated by controlling his/her angular w_b and linear v_b velocities.

4. Neuro-fuzzy control algorithm implemented in the smart clothing system

In our smart clothing system, four sensors integrated to front side of garment perceive surroundings. While the wearer navigates in an unknown environment, ultrasonic sensors detect the presence of obstacles as well as measure the distance to obstacles. During the design process, four sensors were divided into two groups (see Fig. 3 and Fig. 1). In order to differentiate height of obstacles, two ultrasonic sensors were considered to be placed up position on the garment while the other two placed down position.

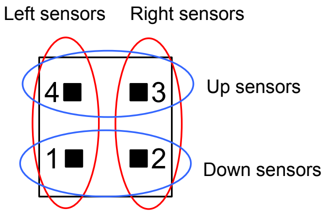

Fig. 3. Sensor's position on the garment

Besides, in order to differentiate position of obstacles whether they are on the left side or right side due to wearer's position, two sensors were considered to be placed at left part of the garment while the other two at right part. Thus, by considering two groups of four-sensor situation; probable cases for detection of obstacles were determined and obstacles' potential positions with regard to person position were examined. Fig. 4. shows some cases for obstacle's position.

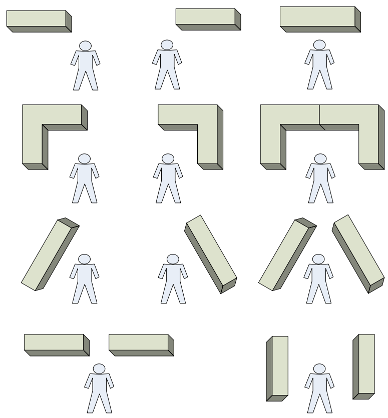

Fig. 4. Different cases between obstacles and user

Before developing control system, at first some assumptions were made according to our study. The target location and user's location was considered to be known variables by the user heuristically. Thus, in our system only the data got by sensors were used as inputs of controller. The framework of the proposed control system is shown in Fig. 5.

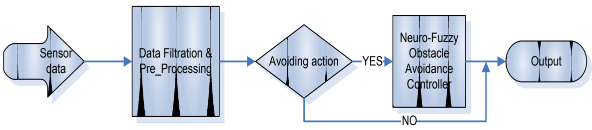

Fig. 5. Framework of control system for proposed smart clothing

In the control system, data filtration and pre-processing is conducted based on data from all sensors in order to understand if there are any obstacles or not.

When the user's path is blocked by an obstacle, the avoiding action is necessary not to crash obstacle by this way neuro-fuzzy obstacle avoidance controller takes place and gives output to make turns to avoid collision. When all distance values got from sensors are larger than a predefined value range, this situation is regarded as there is no obstacle to be avoided. As a result, user is guided to go straight (zero/no turn) as an output response.

After the data filtration and pre-processing by using neural network and fuzzy logic principles, a neuro-fuzzy obstacle avoidance controller for smart clothing system was designed. The structure of the proposed neuro-fuzzy controller is shown in Fig. 6.

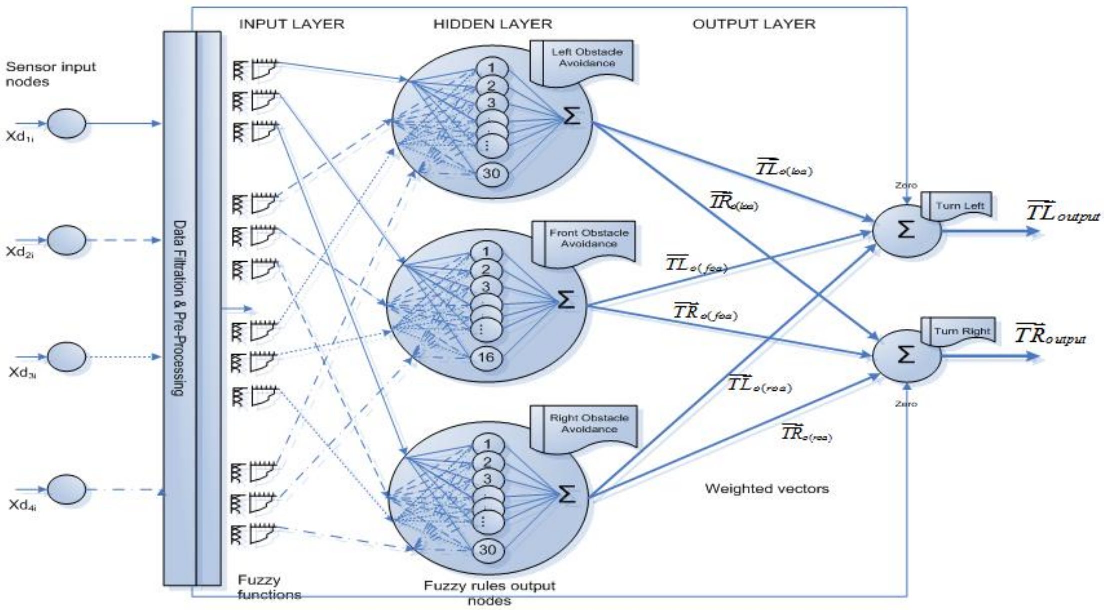

Fig. 6. Proposed neuro-fuzzy control system for the smart clothing

The inputs of controller are the outputs of sensors: the distances to obstacles Xd_{1i}, Xd_{2i}, Xd_{3i}, Xd_{4i} obtained from the sensor 1, sensor 2, sensor 3, and sensor 4 respectively. For instance, the measured distances by sensors to an obstacle located at (-10, 60) cm are shown in Fig. 7.

As seen in the figure, sensor 1 and sensor 2 detect the obstacle around 60 cm while sensor 3 and sensor 4 does not detect this obstacle.

The output signal from the neuro-fuzzy controller is the turning angle and direction ($TR = TurnRight$; $TL = TurnLeft$). The algorithm starts with the data filtration process.

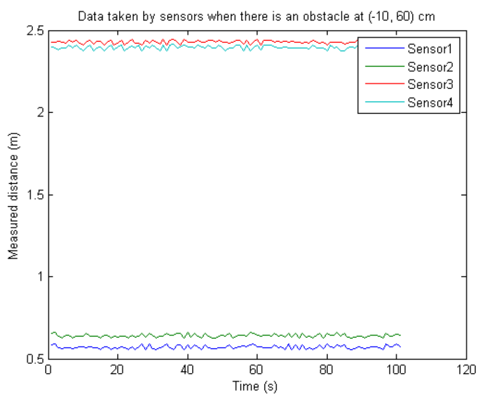

Fig. 7. Measurement results when the obstacle at (-10,60) cm

4.1. *Data filtration and pre-processing*

In this process, data got by sensors are either eliminated or transmitted to controller. It was known that our sensor detection range is up to 6.45 meter[35]. In order to give controller decision, by considering our requirements a predefined value was determined at first. Indeed, in order to give output order to guide user at a right time interval before crashing obstacle, first walking speed of visually impaired people was investigated. Some studies reported that walking speed of normal pedestrian is between 1.22 m/sec (younger pedestrians) and 0.91 m/sec (older pedestrians)[36-37]. Considering this known fact and our observations, walking speed of visually impaired person was assumed as 0.6 m/sec. Then, during walking the distance to be checked for obstacles was defined as 2.5 meters.

By this way, a value of 2.5 meter (predefined value) was considered for elimination of data that means if the sensor detects the distance to an object larger than 2.5 meter or in other words if the object locates 2.5 meter or further away from user's location then, the data is eliminated and considered as there is no object on the way of user.

Thus, in the first algorithm averaged input data larger than 2.5 meter for all sensors is being filtered, assigned to 2.5 m and then directly sent to go straight position (zero) which is interpreted as no turning action. Secondly, for the averaged data smaller than 2.5 meter was interpreted as there is an object/s on the way of user, and according to decision of position of object, it is sent to avoidance strategy to be processed. Fig. 8. explains the data elimination process.

In fact, when the all sensor values are between 2 and 2.5 meter, they are interpreted as there is an object at very far and it is not necessary to avoid this obstacle at this time interval quickly. Thus, this situation is again assigned to go straight position (zero) as if there is no obstacle that should be avoided.

However, sometimes one, two or three of sensors may measure between 2 and 2.5 m because of detection of obstacle at far away or noisy data, while the other/s detects an obstacle within 2 meter. In this case, if at least one of the sensor values is less than 2 meter, it is interpreted as there is an obstacle that should be extremely avoided.

In this manner, the position of object plays an important role to guide user with right decision output order in particular to give right turning angle to user in order to avoid obstacle. In order to decide object's position, experiments were conducted with various object's position in x and y-axis in a real-time environment.

For each sensor 9900 data was obtained. In this concept, possible scenarios for detection of objects by using four sensors were formed.

After determining object's position, data is sent to one of the avoidance strategy: left, front, and right obstacle avoidance. This time, neuro-fuzzy controller starts processing data.

4.2. *Neuro-Fuzzy Controller*

The aim of the neuro-fuzzy controller is to compute the turning angle when the avoidance strategy is required (output of data filtration process). Neuro-fuzzy algorithm is composed of

1- Input layer
2- Hidden layer (rule layer and consequence layer)
3- Output layer

In the input layer and hidden layer of algorithm, fuzzy inference system (FIS) takes place. To set up the fuzzy inference system, MATLAB® Fuzzy Logic Toolbox was used. As shown in Fig. 6, three types of fuzzy inference system was developed namely; (i) left obstacle avoidance, (ii) front and (iii) right obstacle avoidance fuzzy inference system.

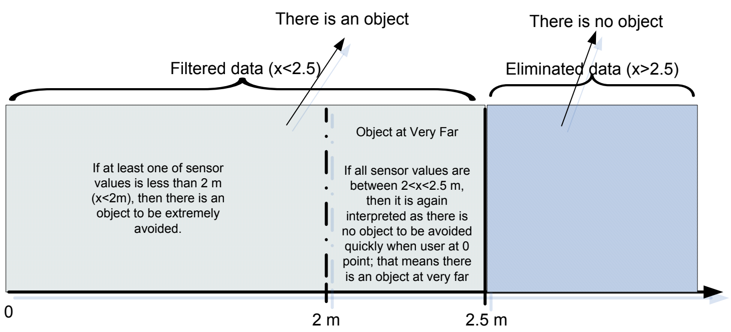

Fig. 8. Data elimination process

4.2.1. Input layer

a) Fuzzification

The fuzzification procedure maps the crisp input values to the linguistic fuzzy terms with membership values between 0 and 1.

In this layer, the inputs are the filtered data and each of these inputs is classified to fuzzy set membership functions. The inputs of fuzzy inference system are "distances to an obstacle" information from sensor 1, sensor 2, sensor 3, and sensor 4, which are described by three linguistic variables: Near, Far and Very Far. The domain of functions is being from 0 (minimum) to 2.5 meter (maximum) for each sensor. The two linguistic variables near and far were described by triangular membership functions, whereas very far described by trapezoidal membership function as shown in Fig. 9. Indeed, the input values between $2 < Xd_i \leq 2.5$ was regarded as there is no detected obstacles neither at far nor near, thus they were interpreted as Very Far as explained in Section 4.1.

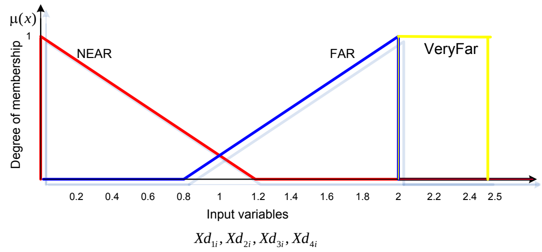

Fig. 9. The membership functions for input variables

The outputs of fuzzy inference system were also described by fuzzy linguistic variables, which are turn left small (S), medium (M), large (L), and very large (VL), and similarly turn right small (S), medium (M), large (L), and very large (VL) as shown in Fig. 10 and Fig. 11, respectively.

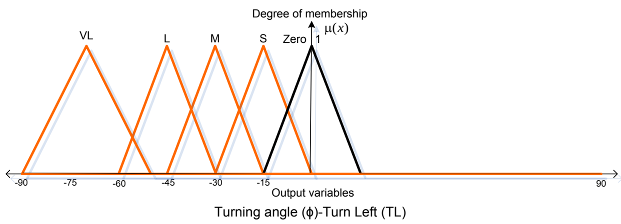

Fig. 10. The membership functions for output variables "turn left"

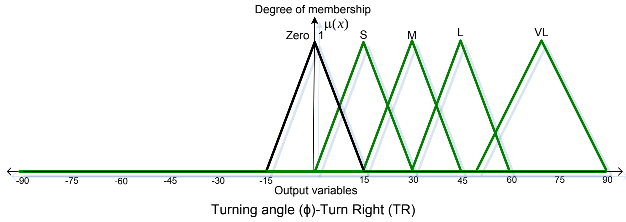

Fig. 11. The membership functions for output variables "turn right"

As mentioned in Section 2, the aim of using four vibration motors in one side of the garment is to satisfy turning angles. For instance; when the output order is turn left small, only the first vibration motor on the left will act. When the output order is turn left medium, large or very large, then two vibration motors, three vibration motors or four vibration motors on the left will simultaneously act, respectively. The domain of functions is [-90 90]. All the linguistic variables were denoted by triangular membership functions (MF).

Triangular MF was mainly selected because of limited computational resources of microcontroller. In general, it is specified by three parameters {a, b, c}[16]:

$$triangle(x;a,b,c) = \begin{cases} 0, x < a \\ \dfrac{x-a}{b-a}, a \leq x \leq b \\ \dfrac{c-x}{c-b}, b \leq x \leq c \\ 0, c \leq x \end{cases} \qquad \textbf{(3)}$$

The parameters {a, b, c} determine the x coordinates of three corners of the underlying membership function.

4.2.2. Hidden layer

4.2.2.1 Hidden layer 1/ Rule Layer

b) Fuzzy rule layer

In this layer in order to control user's motion in an environment as well as establish the relation between sensor values and turning angle, 77 rules were designed. The rules are defined by human knowledge by using observed data taken by real time measurements. Therefore, training of data was done off-line. According to object's position determined, the turning angle of user was decided.

Table 2 shows the recommended turning angle for user to avoid obstacle concerning its position. In the table "R" and "L" indicate the turn right and turn left, respectively. Additionally, as mentioned above, {Z, S, M, L, VL} values denote the turning angle in terms of linguistic variables.

Table 2. The relation between turning angle and detected object position

Object at y-axis (cm) \ Turning angle (φ)	Object at x-axis (cm) -∞	-40	-30	-20	-10	0	10	20	30	40	-∞
0-100	Z	RS	RS	RM	RL	RVL/ LVL	LL	LM	LS	LS	Z
100-200	Z	RS	RS	RS	RM	RL/LL	LM	LS	LS	LS	Z
∞	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z

During the fuzzy rules design; one rule was designed for the situation when there is an obstacle at very far (all sensor values are between 2 and 2.5m) or in other words, there is no detected obstacles neither at far nor near. Thus, this rule was assigned to go straight position (zero-no turning action) such that it is interpreted as if there is no obstacle that should be avoided quickly as mentioned in Section 4.1. Besides, when there is an obstacle/s on the left, right or front of the user, 30, 30 and 16 rules were designed for left, right and front obstacle avoidance, respectively (see APPENDIX-A). The reason for designing separate fuzzy inference systems (left, right, front) is that there are some rules which are common for left and right positioned obstacles.

For instance; consider algorithm 1, rule 6 (see APPENDIX-A):

Rule 6: Xd_{1i} <2 & Xd_{2i} <2 & Xd_{3i} >2 & Xd_{4i} >2

According to fuzzy inference system, the values Xd_{1i} and Xd_{2i} can correspond both Near or Far. Consider, both Xd_{1i} and Xd_{2i} correspond to Near; in this case by one fuzzy inference system, the position of obstacle cannot be determined correctly whether the obstacle is on the left or right. However, it is known that if $Xd_{1i} > Xd_{2i}$ then, obstacle at the right; if $Xd_{1i} < Xd_{2i}$ then, obstacle at the left. For instance; if Xd_{1i} =0.8, Xd_{2i} =0.85, since $Xd_{1i} < Xd_{2i}$, it can clearly be deduced that obstacle at the left (It is more close to left sensor).

However, with FIS they are only interpreted as "Near" and the output decision is uncertain or cannot be defined clearly.

Therefore, since there is no rule definition in FIS that identifies $Xd_{1i} > Xd_{2i}$ or $Xd_{1i} < Xd_{2i}$ condition, three fuzzy inference systems (left, right, front) have been separately considered to overcome this problem and not to guide user wrongly. Particularly, rules 21-22 and 24-25 in the left obstacle avoidance neuron and in the right obstacle avoidance neuron (see APPENDIX-A) explain this situation clearly.

4.2.2.2. Hidden layer-2/ Consequence layer

c) Fuzzy Implication

The choice of fuzzy implication rule is very important while designing a fuzzy control system. Fuzzy implication evaluates the consequent part of each rule. After the inputs have been fuzzified and degree of each rule is calculated using AND operator (see rules; Appendix-A), the output membership function is then truncated by fuzzy implication.

In this research, among the various implication methods, Larsen product implication method was used. The Larsen product implication is given by

$$\mu_{A\to B}(x,y) \equiv \mu_A(x).\mu_B(y) \qquad \textbf{(4)}$$

where A→B denotes an implication in the universe U and V. It uses the arithmetic product between the two membership functions in the universe of discourses U and V[38].

All the rules were evaluated in this manner and output membership functions were aggregated using MAX operator to result in fuzzy output.

d) Defuzzification

The fuzzy implication and as well as aggregation yield the fuzzy output, which is the union of all individual rules that are validated for the control action in a cumulative manner using MAX (OR) operator.

Conversion of this fuzzy output to crisp output is defined as defuzzification. In our research the centroid method, which returns the center of area under the curve, was used for the proposed controller. Let μ_{out} (TL_r) and μ_{out} (TR_r) show the center of membership functions of the output variables for left *(l)*, front *(f)*, and right *(r)* obstacle avoidance neurons after the evaluation of rules, where r=1,2,3…n are the rule numbers for each avoidance neuron and TL_o and TR_o are the crisp values which describe the outputs for Turn Left and Turn Right commands. The value of the output control for each avoidance by centroid method is described as:

For left obstacle avoidance *(loa)* neuron, final output:

$$TR_{o(loa)} = \frac{\sum_{lr=1}^{30} TR_{lr}\mu_{out}(TR_{lr})}{\sum_{lr=1}^{30} \mu_{out}(TR_{lr})}, \quad TL_{o(loa)} = 0 \qquad \textbf{(5)}$$

For right obstacle avoidance *(roa)* neuron, final output:

$$TL_{o(roa)} = \frac{\sum_{rr=1}^{30} TL_{rr}\mu_{out}(TL_{rr})}{\sum_{rr=1}^{30} \mu_{out}(TL_{rr})}, \quad TR_{o(roa)} = 0 \qquad \textbf{(6)}$$

For front obstacle avoidance *(foa)* neuron, final output:

$$TL_{o(foa)} = \frac{\sum_{fr=1}^{16} TL_{fr}\mu_{out}(TL_{fr})}{\sum_{fr=1}^{16} \mu_{out}(TL_{fr})}, \quad TR_{o(foa)} = \frac{\sum_{fr=1}^{16} TR_{fr}\mu_{out}(TR_{fr})}{\sum_{fr=1}^{16} \mu_{out}(TR_{fr})} \qquad \textbf{(7)}$$

When there is no obstacle: $TL_o = 0$ and $TR_o = 0$

4.2.3. Output Layer

Fuzzy-Neural Approximation and Final Outputs

Fig. 12 shows the basic diagram of fuzzy neural approximation of proposed controller. Here the weight functions are approximated by fuzzy sets. In this output layer, the outputs of consequence layer will be the inputs of output layer and the final output will be desired turning angle in order to avoid obstacle. Thus, the weight functions of output layer will be the output functions *(O_{NTL}, O_{NTR})* as shown in Fig. 12, where N denotes the number of neurons that are left obstacle avoidance neuron (1), front obstacle avoidance (2), and right obstacle avoidance (3), and TL denotes the turning position to left whereas TR denotes the turning position to right.

Unless data is not processed in the avoidance neurons, which means it is eliminated by the data filtration and pre-processing, zero function (Z(0)=0) that demonstrates there is no obstacle on the way of user will come additionally, thus the final outputs are calculated as follows:

$$\overrightarrow{TL}_{output} = \sum_{N=1}^{3} O_{N_{TL}} \vee Z(0) \qquad \textbf{(8)}$$

$$\overrightarrow{TR}_{output} = \sum_{N=1}^{3} O_{N_{TR}} \vee Z(0) \qquad \textbf{(9)}$$

By combining equations 5-9, the final outputs of the proposed neuro-fuzzy controller can be written finally as

$$\overrightarrow{TL}_{output} = \left(\frac{\sum_{fr=1}^{16} TL_{fr}\mu_{out}(TL_{fr})}{\sum_{fr=1}^{16} \mu_{out}(TL_{fr})} + \frac{\sum_{rr=1}^{30} TL_{rr}\mu_{out}(TL_{rr})}{\sum_{rr=1}^{30} \mu_{out}(TL_{rr})} \right) \vee Z(0) \qquad \textbf{(10)}$$

$$\overrightarrow{TR}_{output} = \left(\frac{\sum_{lr=1}^{30} TR_{lr}\mu_{out}(TR_{lr})}{\sum_{lr=1}^{30} \mu_{out}(TR_{lr})} + \frac{\sum_{fr=1}^{16} TR_{i}\mu_{out}(TR_{fr})}{\sum_{fr=1}^{16} \mu_{out}(TR_{fr})} \right) \vee Z(0) \qquad \textbf{(11)}$$

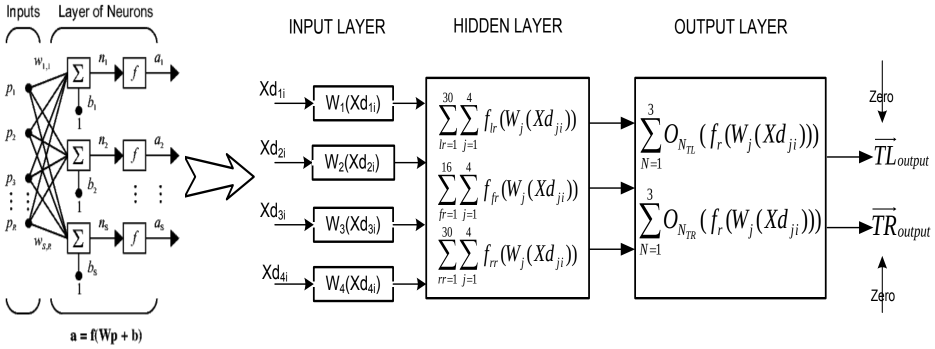

Fig. 12. Basic diagram of fuzzy neural approximation of proposed controller

As a result, the outputs of described neuro-fuzzy controller are processed by microcontroller and thus, they are transmitted to vibration motors as signals defined by intervals of “S”, “M”, “L”, “VL” for turning action in order to guide user.

Additionally, by using the outputs of proposed neuro-fuzzy controller, the angular velocity of the user is known. Hence, the position of walking person can be estimated by using these outputs associated with the Equations (1) and (2) given in Section 3.1.

5. Implementation of the proposed system

To analyze the detection and avoidance capability of the proposed system, experiments were conducted as seen in Fig. 13.

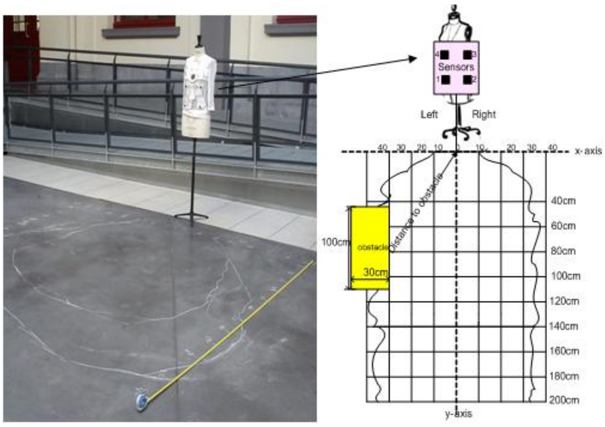

Fig. 13. Overview of experiment set-up for obstacle detection

First, the mannequin on which smart clothing system including ultrasonic sensors and vibration motors were placed, was positioned at (0, 0) to detect the obstacle (see Figure 13). Obstacle was positioned to a distance starting from (0, 0) to (±40, 200). Measurements were repeated every 20cm starting from 40cm to 200cm of y-axis. Measurements were recorded in MATLAB by using National Instruments® DAQ (Data Acquisition) Card. The actual position of obstacle with the one measured by the sensors were recorded.

During the experiments, sensor 1 and 4 locates at left side; sensor 2 and 3 locates at right side. Furthermore, both sensor 1 and 2 were placed at down position, on the other hand both sensor 3 and 4 were placed at the up position as mentioned in Figure 3. For each position of obstacle 100 data was taken and analyzed for each sensor, separately.

By this way, the success of the proposed system was tested in real environment for its detection and avoidance capability. Additionally, data taken by real time experiment results were used to compare multi-layer fuzzy inference systems (left, right, and front) with one layer fuzzy inference system.

6. Results and comparison of multi-layer fuzzy controller with one layer fuzzy controller

One layer fuzzy system was designed using the same 77 rules presented in Appendix-A. The rules in left, right and front obstacle avoidance neurons (30, 30 and 16 rules) were gathered into one fuzzy inference to define one layer fuzzy inference system.

In order to validate the efficiency of the proposed system, it has been implemented on the real data acquired in the experimentation during experimental phase as mentioned in Section 5. Our system is then compared with a basic one layer FIS described above.

The outputs of these two systems are presented in Figures 14, 15 and 16. These results demonstrate that in most of cases, the outputs of the two systems are the same or very close (less than 15°) of the target angles especially for the case when obstacle is in the front of the interactive garment (see Fig.15). For instance, when obstacle is in the front, the expected output of the controller should be turn left and turn right command at the same time in order to let the user choose his/her turning direction randomly and the angle should be larger than 45 to avoid this obstacle. Based on this fact, with reference to Fig. 15, the outputs of both controllers (one layer and multi-layer FISs) generally matched with the targets as expected.

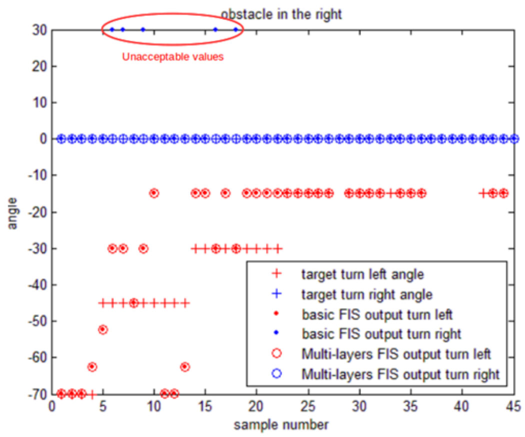

Fig. 14. Outputs of the multi-layer FIS and one-layer FIS and target angles when the obstacle is at the right

However, in specific cases that we explained earlier (see Section 4.2.2.1), the multi-layers FIS outperforms the basic FIS. Indeed, the basic FIS is not able to find the right direction especially for right and left avoidances (see Fig.14 and Fig.16).

In particular; when the obstacle is at the right, user should be guided by turning left and when the obstacle is at the left, user should be guided by turning right in order to avoid it.

Therefore; in case of right obstacle, the expected output value of the fuzzy controller should be "Turn left with an angle and Turn right=0". On the contrary, in case of left obstacle, the expected output value of the fuzzy controller should be "Turn right with an angle and Turn left=0".

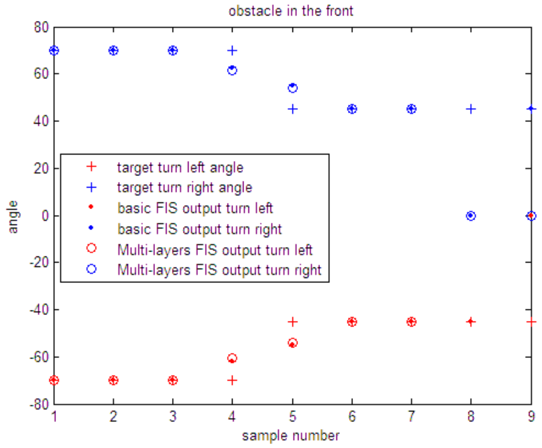

Fig. 15. Outputs of the multi-layer FIS and one layer FIS and target angles when the obstacle is in the front

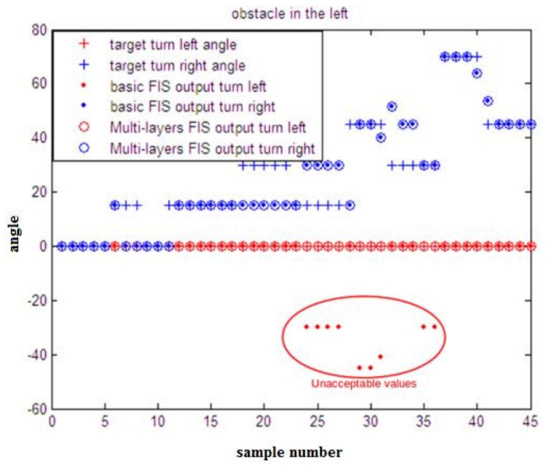

Fig. 16. Outputs of the multi-layer FIS and basic FIS and target angles when the obstacle is at the left

When one layer FIS is compared with multi-layer FIS, it was observed that in some cases, basic one layer FIS presented unacceptable output values such as Turn Right with an angle of 30° instead of 0° (see Fig.14).

In more detail, when the obstacle is at the right, fuzzy controller should present just turn left command corresponding to turning angles from 0 to -70 (TL → [-70, 0) & TR=0). According to Fig.14, in some cases, one layer basic FIS generated TR=30 instead of TR=0 compared to multi-layer FIS.

Similarly, as seen in Fig.16, single FIS presented again some unacceptable output values for left obstacle by giving wrong decision output as "Turn Left with an angle" instead "Turn Left=0". That means in case of left obstacle, the turning angle should range from 0 to +70 (TR→(0, 70] & TL=0).

However, for that case in some instances basic FIS generated Turn Left command ranging from -45 to -30 instead of TL=0 (see Fig.16). Therefore, it is obvious that for special cases single FIS did not present right decision output in terms of direction for right and left avoidances.

These errors are not acceptable in this kind of application since this leads to wrong orientation of the user and a probable collision with the obstacle.

On the other side, for all the cases, multi-layer FIS presented right decision output in terms of direction. It has only errors in definition of angles like single FIS also has. For instance, in some cases instead of 45°, it presented 30°, or instead of 30°, it presented 15°. Indeed, the error in terms of angles can be acceptable because of the noisy data taken by sensors. It does not let the user a direct collision with an obstacle.

As a result, it is apparent that the multi-layer FIS gives better results than the one layer FIS and it is capable of guiding user in right decision output in terms of direction.

7. Conclusion

In this study, an algorithm based on neuro-fuzzy controller composed of multi layer FIS was developed in order to detect obstacles as well as avoid obstacles. The proposed algorithm was implemented in a smart clothing system developed for visually impaired people and then it was tested and compared with one layer FIS for its detection capability and avoidance in real time environment. It is possible to make an experiment with dynamical object avoidance in order to collect data and extract the rule set and the membership function from the data.

It presented better and compromising results according to one layer FIS and it is capable of guiding user to a right orientation especially in terms of direction in order to avoid obstacles.

The proposed solution provided an interactive interface for visually impaired people navigation concerns. In this sense, the successful outcome of this research is supposed to provide an impetus for the future improvements in the field of disabled people.

As our future research work; we aim to develop a kind of system for visually impaired people that can be fully integrated with GPS, RFID, camera and vocal guidance, not only can it track the user, but also find a route to specific destination, and then guide the user to this destination using synthesized speech by ensuring localization information to user such as the street address of the current location etc.

Acknowledgement

This work was accomplished through the collaboration of Istanbul Technical University, Istanbul (Turkey) and Ecole Nationale Supérieure des arts et Industries Textiles, Roubaix (France). Appreciation is extending to ENSAIT, GEMTEX Laboratory and ITU, Textile-Clothing Control and Research Laboratory for their support and to perform experimental work. We also wish to thank to French Embassy in Turkey for the financial support. Furthermore we would like to present our special thanks to Sim-Art Tekstil San. Tic. Ltd. Şti. to produce smart clothing prototype.

References

1. Zong, G., Deng, L., & Wang, W. (2006). A Method for Robustness Improvement of Robot Obstacle Avoidance Algorithm. *2006 IEEE International Conference on Robotics and Biomimetics* , pp. 115-119
2. Borenstein J, Koren Y. (1991). The vector field histogram fast obstacle avoidance for mobile robots, *IEEE Int. Conf. Robotics and Automation*, pp.278-288
3. Deng, M., Inoue, a, Sekiguchi, K., & Jiang, L. (2010). Two-wheeled mobile robot motion control in dynamic environments. *Robotics and Computer-Integrated Manufacturing*, **26**(3), 268-272
4. Baltes J., Hildreth N, (2001). Adaptive path planner for highly dynamic environments, *RoboCup-2000: Robot Soccer World Cup iV*. Berlin: Springer Verlag, p. 76-85
5. Ku, A. L., C.-heng, Tsai, W.-hsiang, & Member, S. (2001). Obstacle Avoidance in Person Following for Guidance Using Vehicle Location Estimation and Quadratic Pattern Classifier. *System*, **48**(1), 205-215.
6. Toibero, J. M., Roberti, F., Carelli, R., & Fiorini, P. (2010). Switching control approach for stable navigation of mobile robots in unknown environments. *Robotics and Computer-Integrated Manufacturing*, **27**(3), 558-568.
7. Ray, A. K., Gupta, M., Behera, L., & Jamshidi, M. (2008). Sonar based Autonomous Automatic Guided Vehicle (AGV) navigation. *IEEE International Conference on System of Systems Engineering*, pp. 1-6
8. Ray, K.A., Behera, L., Jamshidi., M., (2009). Sonar Based Rover Navigation for Single or Multiple Platforms : Forward Safe Path and Target Swithcing Approach, *IEEE Systems Journal*, 2(2), 258-272
9. Marques, Carlos, Lima, P. (2004). Multisensor Navigation for Nonholonomic Robots in Cluttered Environments. *IEEE Robotics & Automation Magazine, September*, pp. 70-82.
10. Jincong, Y, Zhang, X., Ning, Z., & Huang, Q. (2009). Intelligent Robot Obstacle Avoidance System Based on Fuzzy Control. *First International Conference on Information Science and Engineering*, pp. 3812-3815
11. Chen, Y.-sheng, & Juang, J.-gau. (2009). Intelligent Obstacle Avoidance Control Strategy for Wheeled Mobile Robot. *ICROS-SICE 2009*, pp. 3199-3204.
12. Raguraman, S. M., Tamilselvi, D., & Shivakumar, N. (2009). Mobile Robot Navigation Using Fuzzy logic Controller. *Control Automation, Communication and Energy Conservation*, pp. 3-7.
13. Guo, H., Cao, C., Yang, J., & Zhang, Q. (2009). Research on Obstacle-Avoidance Control Algorithm of Lower Limbs Rehabilitation Robot Based on Fuzzy Control. *Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 151-155
14. Pradhan, S., Parhi, D. Panda, A, (2009). Fuzzy Logic techniques for several mobile robots, *Applied Soft Computing*, **9**, 290-304
15. Park, K., & Zhang, N. (2007). Behavior-Based Autonomous Robot Navigation on Challenging Terrain: A Dual Fuzzy Logic Approach. *IEEE Symposium on Foundations of Computational Intelligence*, pp. 239-244.
16. Farooq, U., Hasan, K. M., Amar, M., Khan, S., & Javaid, S. (2010). A Low Cost Microcontroller Implementation of Fuzzy Logic Based Hurdle Avoidance Controller for a Mobile Robot. *Electrical Engineering*, 480-485.
17. Maaref, H., & Barret, C. (2000). Sensor-based fuzzy navigation of an autonomous mobile robot in an indoor environment. *Control Engineering Practice*, **8**(7), 757-768.
18. Thongchai, S., & Kawamura, K. (2000). Application of fuzzy control to a sonar-based obstacle avoidance mobile robot. *IEEE International Conference on Control Applications. Conference Proceedings*, pp. 425-430
19. Liata, J.R, Sarabia, E.G., Arce J., Oria, J.P., (1998). Fuzzy controller for obstacle avoidance in robotic manipulators using ulstrasonic sensors, *AMC-98* Coimbra, pp. 647-652

20. Gharieb, W., Nagib, G. (2000). Fuzzy Guidance Control For a Mobile robot, *Current Advances in Mechanical Design and Production*, MDP-7, Cairo, pp. 67-74
21. Li, W. 1994. Fuzzy logic based robot navigation in uncertain environments by multisensor integration. *IEEE International Conference Multisensor Fusion and Integration for Intelligent Systems*, pp. 259-264
22. Choi, J. W., Kwon, S. H., Lee, H. Y., & Lee, S. G. (1998). Navigation strategy of an intelligent mobile robot using fuzzy logic. *International Conference on Fuzzy Systems Proceedings*, pp. 602-605
23. Kim, B. N., Kwon, O. S., Kim, K. H., Lee, E. H., & Hong, S. H. (1999). A study on path planning for mobile robot based on fuzzy logic controller. *Proceedings Multimedia Technology for Asia-Pacific Information Infrastructure*, pp. 1002-1005
24. Hui, NB, Pratihar DK, (2006). Mobile robot navigation: potential field approach vs. genetic-fuzzy system,
    Advances in Soft Computing, 36, 67-76
25. Liu, Q. (2006). Fuzzy Obstacle-avoiding Controller of Autonomous Mobile Robot Optimized by Genetic Algorithm under Multi-obstacles Environment. *6th World Congress on Intelligent Control and Automation*, pp. 3255-3259
26. Liu, Q., Lu, Y, Xie C., (2006). Optimal Genetic fuzzy obstacle avoidance controller of autonomous mobile robot based on ultrasonic sensors, *Int. Conf. on Robotics and Biomimetics*, China, pp. 125-129
27. Szemes, P. T., Hashimoto, H., & Korondi, P. (2005). Pedestrian-Behavior-Based Mobile Agent Control in Intelligent Space. *IEEE Transactions on Instrumentation and Measurement*, **54**(6), 2250-2257
28. Mahyuddin, M. N., Wei, C. Z., & Arshad, M. R. (2009). Neuro-fuzzy algorithm implemented in Alteras FPGA for mobile robot's obstacle avoidance mission. *TENCON IEEE Region 10 Conference*, pp. 1-6
29. Nasiruddin Mahyuddin, M. (2009). Neuro-fuzzy algorithm for obstacle avoidance mission of a mobile robot using FPGA. *Innovative Technologies in Intelligent Systems and Industrial Applications*, 305-310
30. He, K., Sun, H., & Cheng, W. (2008). Application of fuzzy neural network based on T-S model for mobile robot to avoid obstacles. *7th World Congress on Intelligent Control and Automation*, pp. 8282-8285
31. Ray, A. K., Behera, L., & Jamshidi, M. (2009). GPS and sonar based area mapping and navigation by mobile robots. *7th IEEE International Conference on Industrial Informatics*, pp. 801-806
32. Hui, N. B., & Pratihar, D. K. (2009). A comparative study on some navigation schemes of a real robot tackling moving obstacles. *Robotics and Computer-Integrated Manufacturing*, **25**(4-5), 810-828
33. Kursun, Bahadir, S., Koncar, V., Kalaoglu, F., (2012). Wearable obstacle detection system fully integrated to textile structures for visually impaired people, *Sensors and Actuators A: Physical*, **179**, 297-311
34. Bajo, J., Sanchez, M., Alanso, V., Berjon, V., Fraile, J., Corchado, J., (2010). A distributed architecture for facilitating the integration of blind musicians, *Expert Systems with Applications* **37**, 8508–8515
35. Maxbotix Inc., LV-Maxsonar ®-EZ3™ Data Sheet and LV Chaining Application Notes, avalaible at <www.maxbotix.com> accessed February 2011
36. Knoblauch, R.L, Pietrucha, M.T, And Nitzburg, M., (2006). Field Studies of Pedestrian Walking Speed and Start-Up Time, *Transportation Research Record*, 1538, 27-38
37. Tim J. Gates, David A. Noyce, Andrea R. Bill, Nathanael V., (2006). Recommended Walking Speeds for Pedestrian Clearance Timing Based on Pedestrian Characteristics, *TRB 2006 Annual Meeting*, Paper No. 06-1826
38. Gaweda A.E., Scherer R., (2004), Fuzzy Number-Based Hierchical Fuzzy Systems, *Artificial Intelligence and Soft Computing*, LNAI 3070, pp. 302–307

APPENDIX-A

<Rules for left obstacle avoidance>

Algorithm 1 (Rule-2): $Xd_{1i} < 2$ & $Xd_{4i} < 2$ & ($Xd_{2i} > 2$ | $Xd_{3i} > 2$)

- Rule 1:
  If Xd_{1i} = Near & Xd_{2i} =Near & Xd_{3i} =VeryFar & Xd_{4i} =Near =>Turn Right VL
- Rule 2:
  If Xd_{1i} = Near & Xd_{2i} =Near & Xd_{3i} =VeryFar & Xd_{4i} =Far => Turn Right VL
- Rule 3:
  If Xd_{1i} = Near & Xd_{2i} =Far & Xd_{3i} =VeryFar & Xd_{4i} =Near => Turn Right VL

…….

Algorithm 1 (Rule-6): $Xd_{1i} < 2$ & $Xd_{2i} < 2$ & $Xd_{3i} > 2$ & $Xd_{4i} > 2$

if $Xd_{1i} > Xd_{2i}$ obstacle at the right; else obstacle at the left

- Rule 21:
  If Xd_{1i} = Near & Xd_{2i} =Near & Xd_{3i} =VeryFar & Xd_{4i} =VeryFar =>Turn Right L
- Rule 22:
  If Xd_{1i} = Far & Xd_{2i} =Far & Xd_{3i} =VeryFar & Xd_{4i} =VeryFar => Turn Right M

…….

Algorithm 1 (Rule-7): $Xd_{1i} > 2$ & $Xd_{2i} > 2$ & $Xd_{3i} < 2$ & $Xd_{4i} < 2$

if $Xd_{4i} > Xd_{3i}$ obstacle at the right; else obstacle at the left

- Rule 24:
  If Xd_{1i} =VeryFar & Xd_{2i} =VeryFar & Xd_{3i} =Near & Xd_{4i} =Near =>Turn Right L
- Rule 25:
  If Xd_{1i} =VeryFar & Xd_{2i} =VeryFar & Xd_{3i} =Far & Xd_{4i} =Far => Turn Right M

…….

Algorithm 1 (Rule-8): $Xd_{1i} < 2$ & $Xd_{2i} > 2$ & $Xd_{3i} > 2$ & $Xd_{4i} < 2$

- Rule 27:
  If Xd_{1i} =Near & Xd_{2i} =VeryFar & Xd_{3i} =VeryFar & Xd_{4i} =Near =>Turn Right S
  ….continue

<Rules for right obstacle avoidance>

…..

Algorithm 1 (Rule-5): $Xd_{1i} > 2$ & $Xd_{4i} > 2$ & ($Xd_{2i} < 2$ | $Xd_{3i} < 2$)

….

- Rule 18:
  If Xd_{1i} =VeryFar & Xd_{2i} = Far & Xd_{3i} =VeryFar & Xd_{4i} =VeryFar =>Turn Left S

……

Algorithm 1 (Rule-6): $Xd_{1i} < 2$ & $Xd_{2i} < 2$ & $Xd_{3i} > 2$ & $Xd_{4i} > 2$

if $Xd_{1i} > Xd_{2i}$ obstacle at the right; else obstacle at the left

- Rule 21:
  If Xd_{1i} = Near & Xd_{2i} = Near & Xd_{3i} =VeryFar & Xd_{4i} = VeryFar => Turn Left L
- Rule 22:
  If Xd_{1i} = Far & Xd_{2i} = Far & Xd_{3i} =VeryFar & Xd_{4i} =VeryFar => Turn Left M

…..

Algorithm 1 (Rule-7): $Xd_{1i} > 2$ & $Xd_{2i} > 2$ & $Xd_{3i} < 2$ & $Xd_{4i} < 2$

if $Xd_{4i} > Xd_{3i}$ obstacle at the right; else obstacle at the left

- Rule 24:
  If Xd_{1i} = VeryFar & Xd_{2i} = VeryFar & Xd_{3i} =Near & Xd_{4i} = Near => Turn Left L
- Rule 25:
  If Xd_{1i} = VeryFar & Xd_{2i} = VeryFar & Xd_{3i} =Far & Xd_{4i} =Far => Turn Left M

….

Algorithm 1 (Rule-9): $Xd_{1i} > 2$ & $Xd_{2i} < 2$ & $Xd_{3i} < 2$ & $Xd_{4i} > 2$

- Rule 27:
  If Xd_{1i} = VeryFar & Xd_{2i} = Near & Xd_{3i} =Near & Xd_{4i} =VeryFar=> Turn Left S
  …..continue

<Rules for front obstacle avoidance>

Algorithm 1 (Rule-10): $Xd_{1i} < 2$ & $Xd_{2i} < 2$ & $Xd_{3i} < 2$ & $Xd_{4i} < 2$

- Rule 1:
  If Xd_{1i} =Near & Xd_{2i} =Near& Xd_{3i} =Near& Xd_{4i} =Near =>Turn Left/Right VL
  ……
- Rule 16:
  If Xd_{1i} = Far & Xd_{2i} = Far & Xd_{3i} =Far & Xd_{4i} =Far => Turn Left/Right L

<Rule for there is no obstacle>

Algorithm 1(Rule 1): $Xd_{1i} > 2.5$ & $Xd_{2i} > 2.5$ & $Xd_{3i} > 2.5$ & $Xd_{4i} > 2.5$

there is no obstacle (data filtration and pre-processing)➔ Go straight (Zero)

Algorithm 1(Rule-10): $Xd_{1i} < 2.5$ & $Xd_{2i} < 2.5$ & $Xd_{3i} < 2.5$ & $Xd_{4i} < 2.5$

there is no obstacle when:

- Rule 77:
  If Xd_{1i} =VeryFar & Xd_{2i} =VeryFar & Xd_{3i} =VeryFar & Xd_{4i} =VeryFar => Go straight (Zero)