ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/221921104

Fuzzy Logic Based Navigation of Mobile Robots

Chapter · December 2011

DOI: 10.5772/25621 · Source: InTech

CITATIONS
6

READS
37

2 authors, including:

Amur Salim Al Yahmedi
Sultan Qaboos University

32 PUBLICATIONS **125** CITATIONS

SEE PROFILE

All content following this page was uploaded by Amur Salim Al Yahmedi on 26 February 2017.

The user has requested enhancement of the downloaded file.

14

Fuzzy Logic Based Navigation of Mobile Robots

Amur S. Al Yahmedi and Muhammed A. Fatmi
Sultan Qaboos University,
Oman

1. Introduction

Robots are no longer confined to engineered, well protected sealed corners, but they are currently "employed" in places closer and closer to "us". Robots are getting out of factories and are finding their way into our homes and to populated places such as, museum halls, office buildings, schools, airports, shopping malls and hospitals.

The gained benefit of the potential service and personal robots comes along with the necessity to design the robot in a way that makes it safe for it to interact with humans and in a way that makes it able to respond to a list of complex situations. This includes at least the possibility to have the robot situated in an unknown, unstructured and dynamic environment and to navigate its way in such an environment. One of the fundamental issues to be addressed in autonomous robotic system is the ability to move without collision. An "intelligent" robot should avoid undesirable and potentially dangerous impact with objects in its environment. This simple capability has been the subject of interest in robotic research.

Behavior based navigation systems (Arkin, 1987, 1989; Arkin & Balch, 1997; AlYahmedi et al., 2009; Brooks, 1986, 1989; Fatmi et al. 2006 and Ching-Chih et al. 2010) have been developed as an alternative to the more traditional strategy of constructing representation of the world and then reasoning prior to acting. The main idea of behavior based navigation is to identify different responses (behaviors) to sensory inputs. For example, a behavior could be "avoiding obstacles" in which sonar information about a close obstacle should result in a movement away from the obstacle. A given set of behaviors is then blended in a certain way to produce either a trade off behavior or a more complex behavior. However, a number of issues with regard to behavior based navigation are still under investigation. These issues range from questions concerning the design of individual behaviors to behavior coordination issues, to intelligently improve "behaviors" through learning.

An important problem in autonomous navigation is the need to deal with the large amount of uncertainties of the sensory information received by the robot which is incomplete and approximate as well as with the fact that the environment in which such robots operate contains dynamics and variability elements.

A fuzzy logic behavior based navigation approach is introduced in this chapter in order to deal with the uncertainty and ambiguity of the information the system receives. Issues of individual behavior design and action coordination of the behaviors will be addressed using fuzzy logic.

The approach described herein, consists of the following four tasks,

- The use of fuzzy sets to represent the approximate positions and possibly shapes of objects in the environment.
- The design of simple fuzzy behaviors (avoiding obstacles, goal reaching, wall following…etc.).
- The blending of the different fuzzy behaviors.

2. Behavior based navigation

One of the long standing challenging aspect in mobile robotics is the ability to navigate autonomously, avoiding modeled and unmodeled obstacles especially in crowded and unpredictably changing environment. A successful way of structuring the navigation task in order to deal with the problem is within behavior based navigation approaches (Arkin, 1987, 1989; Arkin & Balch, 1997; AlYahmedi et al., 2009; Brooks, 1986, 1989; Fatmi et al. 2006; Ching-Chih et al. 2010; Maes, 1990; Mataric, 1997; Rosenblatt et al. 1989, 1994, 1995; Saffiotti, 1997 and Seraji & Howard, 2002).

2.1 Introduction

The basic idea in behavior based navigation is to subdivide the navigation task into small easy to manage, program and debug behaviors (simpler well defined actions) that focus on execution of specific subtasks. For example, basic behaviors could be "avoid obstacles" or "moving to a predefined position". This divide-and-conquer approach has turned out to be a successful approach, for it makes the system modular, which both simplifies the navigation solution as well as offers a possibility to add new behaviors to the system without causing any major increase in complexity. The suggested outputs from each concurrently active behaviors are then "blended" together according to some action coordination rule. The task then reduces to that of coupling actuators to sensory inputs, with desired robot behaviors. Each behavior can take inputs from the robot's sensors (e.g., camera, ultrasound, infrared, tactile) and/or from other behaviors in the system, and send outputs to the robot's actuators(effectors) (e.g., wheels, grippers, arm, and speech) and/or to other behaviors.

A variety of behavior-based control schemes have been inspired by the success of (Brooks, 1986, 1989), with his architecture which is known by the subsumption architecture. In this architecture behaviors are arranged in levels of priority where triggering a higher level behavior suppresses all lower level behaviors. (Arkin, 1987, 1989; Arkin & Balch, 1997), has described the use of reactive behaviors called motor schemas. In this method, potential field is used to define the output of each schema. Then, all the outputs are combined by weighted summation. Rosenblatt et al. (Rosenblatt et al. 1989, 1994, 1995), presented DAMN architecture in which a centralized arbitration of votes provided by independent behaviors combines into a "voted" output. Others (Saffiotti, 1997), (Seraji et al., 2001, 2002), (Yang et al. 2004, 2005; Selekwa et al., 2005 and Aguirre & Gonzales, 2006) used fuzzy logic system to represent and coordinate behaviors.

2.2 Fuzzy behavior based navigation

An important problem in autonomous navigation is the need to deal with the large amount of *uncertainties* that has to do with the sensory information received by the robot as well as with the fact that the environment in which such robots operate contains elements of *dynamics* and *variability* that limit the utility of prior knowledge. Fuzzy theory has the features that enable it to cope with uncertain, incomplete and approximate information. Thus, fuzzy logic stirs more and more interest amongst researchers in the field of robot navigation. Further, in the majority of fuzzy logic applications in navigation, a mathematical model of the dynamics of the robot nor the environment is needed in the design process of the motion controller.

The theory of fuzzy logic systems is inspired by the remarkable human capacity to reason with perception-based information. Rule based fuzzy logic provides a formal methodology for linguistic rules resulting from reasoning and decision making with uncertain and imprecise information.

In the fuzzy logic control inputs are processed in three steps (Fuzzification, Inference and Deffuzification) as seen in Fig. 1.

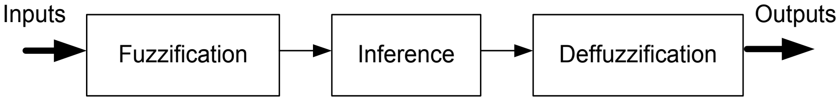


Fig. 1. Fuzzy logic control steps.

In the fuzzification block one defines for example fuzzy set A in a universe of discourse X defined by its membership function $\mu_A(x)$ for each x representing the degree of membership of x in X. In fuzzy logic control, membership functions assigned with linguistic variables are used to fuzzify physical quantities. Next, in the inference block, fuzzified inputs are inferred to a fuzzy rules base. This rules base is used to characterize the relationship between fuzzy inputs and fuzzy outputs. For example, a simple fuzzy control rule relating input v to output u might be expressed in the condition-action form as follows,

$$IF\ v\ is\ W\ then\ u\ is\ Y \tag{1}$$

Where W and Y are fuzzy values defined on the universes of v and u, respectively.

The response of each fuzzy rule is weighted according to the degree of membership of its input conditions. The inference engine provides a set of control actions according to fuzzified inputs. Since the control actions are in fuzzy sense. Hence, a deffuzification method is required to transform fuzzy control actions into a crisp value of the fuzzy logic controller.

In behavior based navigation the problem is decomposed into simpler tasks(independent behaviors). In fuzzy logic behavior based navigation systems each behavior is composed of a set of fuzzy logic rule statements aimed at achieving a well defined set of objectives, for example a rule could be:

If goal is near and to the left then turn left and move forward with a low speed

In general the actions recommended by different behaviors are compiled to yield the most appropriate action according to certain criteria.

2.3 Behavior coordination

The main problem in robot behavior based navigation is how to coordinate the activity of several behaviors, which may be active concurrently with the possibility of having behavior conflict. For example, one may have “goal reaching” behavior and “ obstacle avoidance” behavior active at the same time as seen in Fig 2.

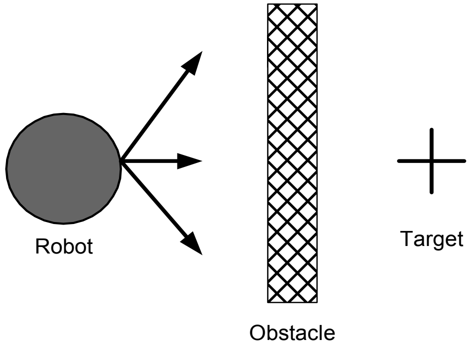


Fig. 2. Conflicts in bahviors.

The coordination task will be to reach a trade-off conclusion that provides the suitable command to the robot actuators which can result in choosing one behavior or a combination of all activated behaviors as shown in Fig.3. Behavior coordination is the point at which most strategies differ. Some of the earlier strategies are based on Brooks subsumption architecture (Brooks, 1986, 1989) uses a switching type of behavior coordination. In the Subsumption approach a prioritization scheme is used in which recommendation of only one behavior with the highest priority is selected, while recommendations of the remaining competing behaviors are ignored. This approach however, leads to inefficient results or poor performance in certain situations. For example if a robot is to encounter an obstacle right in front of it the action that will be selected is "avoid obstacle", the robot then decides to turn left to avoid the obstacle while the goal is to the right of the robot, so the "seek goal" behavior is affected in a negative way.

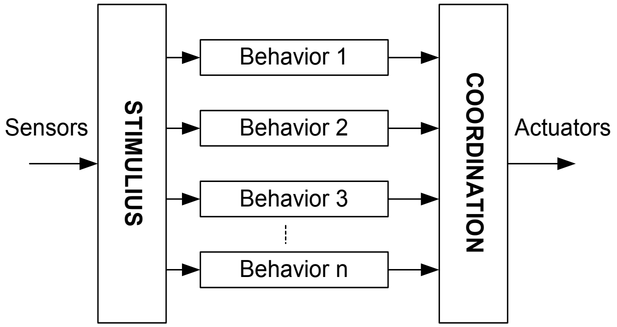


Fig. 3. Structure of behavior based navigation.

Other techniques combine the output of each behavior based on predetermined weighting factors, for example Arkin's motor schema approach(Arkin, 1987, 1989; Arkin & Balch, 1997), or Philipp A. and H.I. Christensen (ALthaus & Christensen, 2002) and the work of Rosenblatt(Rosenblatt et al. 1989, 1994, 1995), who developed the distributed architecture for mobile robot navigation, in which a centralized arbitration of votes provided by independent behaviors. In this method each behavior is allowed to vote for or against certain vehicle actions. The action that win the vote is carried out. These techniques may as well lead to poor performance in certain situations, for example if the robot is to encounter an obstacle right in front of it the “avoid obstacle” behavior may recommend the robot to turn left, while the “seek goal” behavior may request the robot to turn right since the goal is to the right of the robot, this may lead to trade off command that directs the robot forward resulting in a collision with the obstacle.

To deal with these limitations other schemes were recommended that achieve the coordination via considering the situation in which the robot is found, i.e each behavior is allowed to affect the robot motion based on the situational context. (Saffiott, 1997) uses the process of context-dependent-blending in which the current situation is used to decide the action taken using fuzzy logic. Independently (Tunstel et al., 1997) developed an approach similar to context-dependent-blending, in which adaptive hierarchy of multiple fuzzy behaviors are combined using the concept of degree of applicability. In this case certain behaviors are allowed to affect the overall behavior as required by the current situation and goal. The behavior fusion methodology in this chapter is motivated by the approaches used by Saffiot and Tunstel et al.

2.4 Context-dependent behavior coordination

The robot navigation tasks are divided into small independent behaviors that focus on execution of a specific subtask. For example, a behavior focuses on reaching the global goal, while another focuses on avoiding obstacles. Each behavior is composed of a set of fuzzy logic rules aimed at achieving a given desired objective. The navigation rules consist of a set of fuzzy logic rules for robot *velocity* (linear velocity m/s) and *steering* (angular velocity rad/s) of the form

$$IF\ \ C\ then\ \ A \tag{2}$$

Where the condition *C* is composed of fuzzy input variables and fuzzy connectives (And) and the action *A* is a fuzzy output variable. Equation (2) represents the typical form of natural *linguistic* rules .This rules reflect the human expert and reason to ensure logic, reliable and safe navigation. For example, obstacle avoidance behavior has inputs sensory data which can be represented by fuzzy sets with linguistic labels, such as {Near, Medium, Far}, corresponding to distance between robot and obstacle. Typical examples of fuzzy rules are as follow,

If *Front left* is Near And *Front right* is Far, Then *Steering* is Right

If *Front left* is Far And *Front right* is Near, Then *Velocity* is Zero

Where *Front left* and *Front right* are the distances acquired from sensors located in different locations on the robot.

Many behaviors can be active simultaneously in a specific situation or *context*. Therefore, a coordination technique, solving the problem of activation of several behaviors is needed. We call the method *context dependent behavior coordination*. The coordination technique employed herein is motivated by the approaches used by Saffiotti (Saffiotti, 1997). The supervision layer based on the context makes a decision as to which behavior(s) to process (activate) rather than processing all behavior(s) and then blending the appropriate ones, as a result time and computational resources are saved. Fig.4 and Fig.5 represent the architecture of the compared method and our approach, respectively.

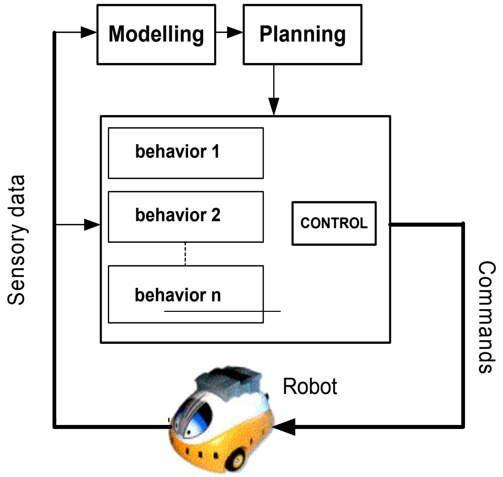


Fig. 4. Architecture of compared method.

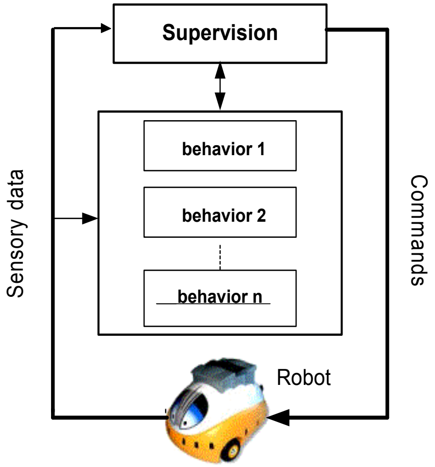


Fig. 5. Architecture of Context Dependent Behavior Coordination.

Our approach consists of the following characteristics.

- The robot navigation is comprised of four behaviors as follows,
  1. Goal reaching behavior
  2. Wall following behavior
  3. Emergency situation behavior
  4. Obstacle avoidance behavior.
- Each behavior is composed of a set of fuzzy logic rules achieving a precise goal.
- The output of each behavior represents the Steering angle and the Velocity.
- The supervision layer defines the priority of each behavior. It selects or activates and blends behaviors depending on situation or context.

3. Simulation & experimental results

3.1 Introduction

The navigation task can be broken down to a set of simple behaviors. The behaviors can be represented using fuzzy *if-then* rules. A context dependent coordination method can be used to blend behaviors.

To validate the applicability of the method simulation and experimental studies were performed.

3.2 Design of individual behaviors

To provide the robot with the ability to navigate autonomously avoiding modeled and unmodeled obstacles especially in crowded and unpredictably dynamic environment the following behaviors were designed: Goal reaching , Emergency situation, Obstacle avoidance, Wall following. Each behavior was represented using a fuzzy *if- then* rule base. The fuzzy rule base comprises the following *if-then* rules:

$$R^{(l)} : IF\ x_1 \text{ is } A_1^l \text{ and ... and } x_n \text{ is } A_n^l, THEN\ y \text{ is } B^l \tag{3}$$

Where l= $1…m$, and m is the number of rules in a given fuzzy rule base, x_1 … x_n are the input variables which are the sensor data of the mobile robot, $A^l_1…A^l_2$ are the input fuzzy sets, B^l is the output fuzzy set and y is the output variable.

3.2.1 Goal reaching behavior

The goal reaching behavior tends to drive the robot from a given initial position to a stationary or moving target position. This behavior drives the robot to the left to the right or forward depending on θ_{error}, the difference between the desired heading (the heading required to reach the goal) and the actual current heading.

Fig.6 gives a schematic block diagram of the goal reaching architecture. From this figure we can notice that the inputs of the goal reaching controller are the distance robot to goal (D_{rg}) and θ_{error} which are given by Equation (4) and Equation (5), respectively.

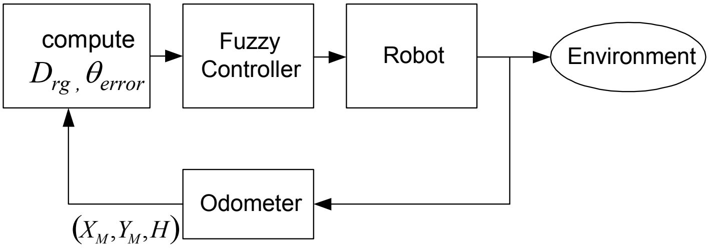


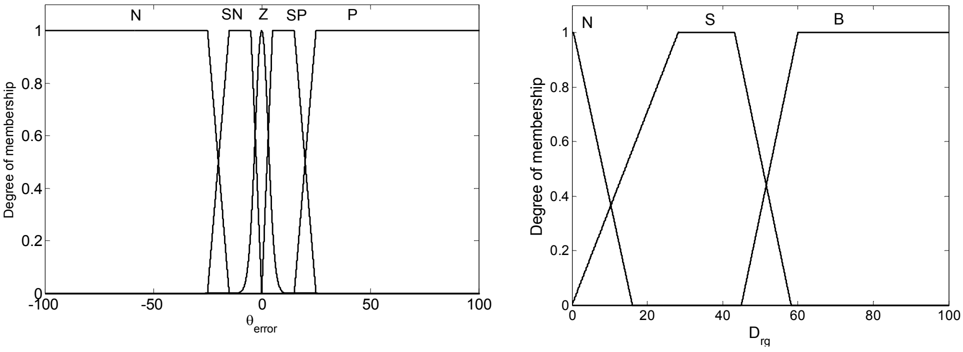
Fig. 6. Control architecture for goal reaching Behavior.

$$D_{rg} = \sqrt{(X_{Goal} - X_M)^2 + (Y_{Goal} - Y_M)^2} \tag{4}$$

$$\theta_{error} = \tan^{-1}\left(\frac{Y_{Goal} - Y_M}{X_{Goal} - X_M}\right) - H \tag{5}$$

Where, *(X_M, Y_M, H)* are the robot position and the heading measured by the robot odometer. *(X_{Gaol}, Y_{Gaol})* is the target position

Although, there is no restriction on the form of membership functions, the appropriate membership functions for D_{rg} (in mm) and θ_{error} (in degrees) shown in Fig.7 were chosen.



N: Negative, SN: Small Negative, Z: Zero, SP: Small Positive, P: Positive.

N for Near, S for Small, B for Big.

Fig. 7. Input membership functions for D_{rg} and θ_{error}.

The Goal reaching is expected to align the robot's heading with the direction of the goal so when θ_{error} is positive, the Steering is left or when θ_{error} is negative, the Steering is Right in a way that minimizes θ_{error}. For The Velocity it is proportional to the distance to goal D_{rg}. Example of goal reaching fuzzy rules.

If θ_{error} is *Positive* And D_{rg} *is Big then Velocity is Small Positive*
If θ_{error} is *Positive* And D_{rg} *is Big then Steering is Left*
If θ_{error} is *Negative* And D_{rg} *is Small then Velocity is Small Positive*
If θ_{error} is *Negative* And D_{rg} *is Small then Steering is Right*
If θ_{error} is *Small Negative* And D_{rg} *is Big then Velocity is Positive*
If θ_{error} is *Small Negative* And D_{rg} *is Big then Steering is Right Front*

Velocity

D_{rg} \ θ_{error}	*Z*	*SN*	*N*	*SP*	*P*
Near	*Z*	*Z*	*Z*	*Z*	*Z*
Small	*P*	*P*	*SP*	*SP*	*SP*
Big	*P*	*P*	*SP*	*P*	*SP*

Steering

D_{rg} \ θ_{error}	*Z*	*SN*	*N*	*SP*	*P*
Near	*F*	*RF*	*R*	*LF*	*L*
Small	*F*	*RF*	*R*	*LF*	*L*
Big	*F*	*RF*	*R*	*LF*	*L*

Table 1. Fuzzy table rules for goal reaching behavior.

For other behaviors the robots needs to acquire information about the environment. The Pekee Robot (Pekee is the robotic platform that will be used to validate the functionality of the proposed scheme experimentally) is endowed with 14 infrared sensors (See Fig.8). These sensors are used to detect obstacles in short-range distances and in a cone of 10 degrees. These sensors are clustered into 6 groups in the simulated robot in such a way as to be similar to Pekee(the simulated robot is very similar kinematically to Pekee).

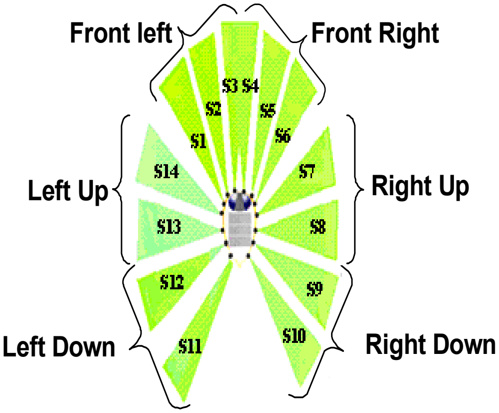


Fig. 8. Clustered sensors arrangement.

For each group, we refer to the minimum of distance measurement by group of sensors as:

Right Down= min (D_9, D_{10})	*Right Up= min* (D_7, D_8)	*Front Right= min* (D_4, D_5, D_6)
Left Down=min (D_{11}, D_{12})	*Left Up=min* (D_{13}, D_{14}).	*Front Left=min* (D_1, D_2, D_3),

D_i is the distance acquired by the sensor $S_{i.}$ i=1….14.

These distances represent the inputs of the fuzzy controller for behaviors like Emergency situation, Obstacle avoidance, Wall following behaviors. Each distance is fuzzified using the following membership function described in Fig.9.

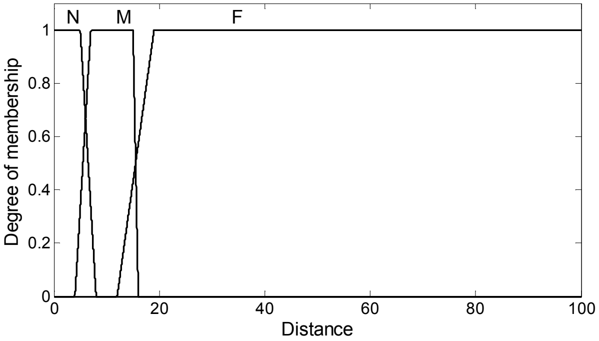
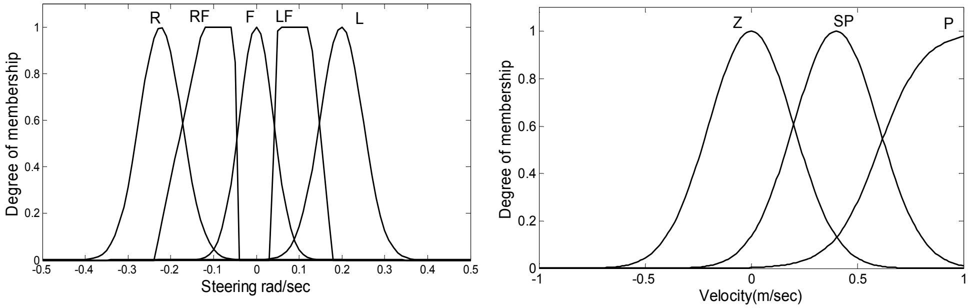


Fig. 9. Input membership function for distance to obstacle, N:Near, M:Medium, F:Far

The output variable are the Steering and the Velocity. Fig.10 illustrates the membership functions for these outputs.



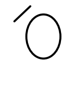
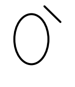
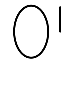
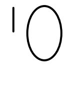
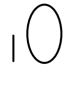
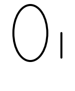
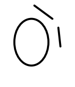
R: Right, RF: Right Front, FL: Front Left, F: Front, L: Left

Z: Zero, SP: Small Positive, P: Positive

Fig. 10. Output membership functions for steering & velocity.

3.2.1 Obstacle avoidance behavior

The obstacle avoidance behavior tends to avoid collisions with obstacles that are in the vicinity of robot. We take into account different cases as shown in table2.

		Inputs						Outputs	
Cases		Right Down	Right Up	Front Right	Front Left	Left Up	left Down	Steering	Velocity
1				F	N			R	Z
				F	M			R	SP
2				N	F			L	Z
				M	F			L	SP
3			N					L	Z
			M					L	SP
4						N		R	Z
						M		FR	P
5							N	R	Z
6		N						L	Z
7			N	N	F			L	Z
			M	M	F			L	SP

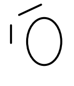
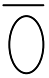
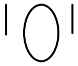
		Inputs						Outputs	
Cases		Right Down	Right Up	Front Right	Front Left	Left Up	left Down	Steering	Velocity
8				F	N	N		R	Z
				F	M	M		R	SP
9				N	N	F		L	Z
				M	M	F		L	SP
				F	F			F	P
10			N	F	F	N		F	SP
			N		M	N		F	SP
			N	M		N		F	SP

Table 2. Fuzzy table rules for obstacle avoidance behavior.

Example of obstacle avoidance rules
If Front left is Near And Front right is Far, Then Steering is Right
If Front left is Near And Front right is Far, Then Velocity is Zero
If Front left is Far And Front right is Near, Then Steering is left
If Front left is Far And Front right is Near, Then Velocity is Zero

3.2.2 Wall following behavior

The objective of the control of the wall following behavior is to keep the robot at a safe close distance to the wall and to keep it in line with it. Example of wall following fuzzy rule:
If Right down is Medium and Front Right is Medium then Steering is Front
If Right down is Medium and Front Right is Medium then Velocity is Positive

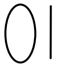
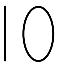
	Inputs						Outputs	
Cases	Right Down	Right Up	Front Right	Front Left	Left Up	left Down	Steering	Velocity
	M	M	F	F			F	P
	M	M	M	F			F	P
			F	F	M	M	F	P
			F	M	M	M	F	P

Table 3. Fuzzy table rules for wall following behavior.

3.2.3 Emergency situation behavior

The emergency situation behavior drives the robot to the left or to the right when it is surrounded by obstacles in away depicted in Table 4.

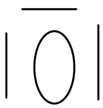
	Inputs				Outputs	
Cases	Right Up	Front Right	Front Left	Left Up	Steering	Velocity
	Far	M	M	M	R	SP
	M	M	M	F	L	SP
	M		F	M	L	Z
	M	F		M	L	Z
	M	M	M	M	R	Z

Table 4. Fuzzy table rules for “emergency” behavior.

3.3 Blending of behaviors

The question to answer once the behaviors are designed is how best decide what the actuators shall receive(in terms of steering and velocity)taking into account the context in which robot happens to be in and relative importance of each behavior. To achieve that the work herein proposes the following architecture with details.

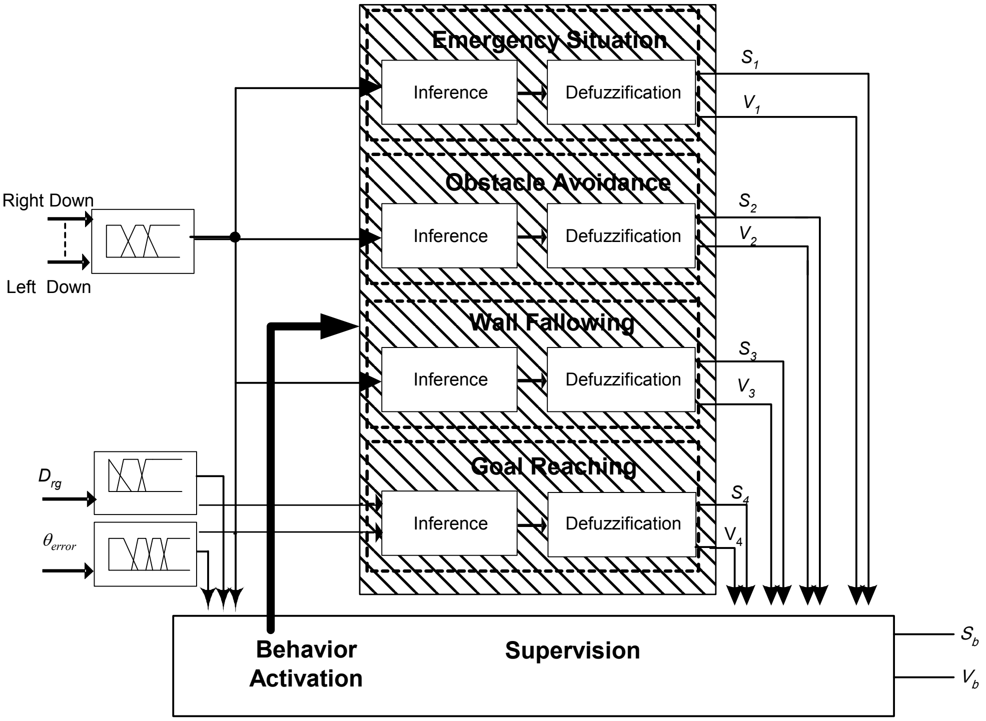


Fig. 11. Supervision Architecture.

S_i and V_i are the output Steering and Velocity of each behavior, i=1...4
S_b and V_b are the output steering and velocity to motor.
The inputs of supervision layer are the degree of membership of each group of sensors in all membership function, D_{rg} and θ_{error},in addition to the Steering and Velocity outputs of all behaviors ($S_1, V_1, S_2, V_2, S_3, V_3, S_4$ *and* V_4).
The supervision layer program is based on fuzzy rules such as,

$$IF\ context\ then\ \ behavior \tag{6}$$

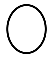
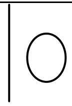
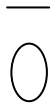
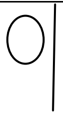
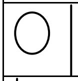
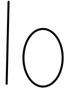
Cases	RD	RU	FR	FL	LU	LD	θ_{error}	D_{rg}	Behavior
		F	F	F	F				Goal Reaching
								N	
				F	F	F	SP		Goal Reaching
				F	F	F	P		
				F	F	F	Z		
			N	N			SN		Fusion
			N	N			N		
			M	M			SN		
			M	M			N		
	F	F	F				SP		Goal Reaching
	F	F	F				P		Goal Reaching
	M	M		F					Wall Following
			F		M	M			Wall Following

Table 5. Context priority.

S_b and V_b are the appropriate velocity and steering control commands sent to the motors in a given robot situation as a result of the decision to activate a certain behavior (avoid obstacle, wall following...). However, S_b and V_b can be the fusion result of many behaviors. As seen at Table 5 the direction of robot to avoid obstacles is taken as default to the left, but, the supervision layer takes into account the position of goal by θ_{error} in such away as to minimize the distance traveled towards the goal as shown in Fig.12.
The advantage of this method is the fact that the number of rules are reduced by reducing the number of input variables to the fuzzy rules of each behavior, for example in the obstacle avoidance behavior there was no need to add θ_{error} as an input.

3.4 Simulation results

To verify the validity of the proposed scheme, some typical cases are simulated in which a robot is to move from a given current position to a desired goal position in various unknown environment. In all cases the robot is able to navigate its way toward the goal while avoiding obstacles successfully.

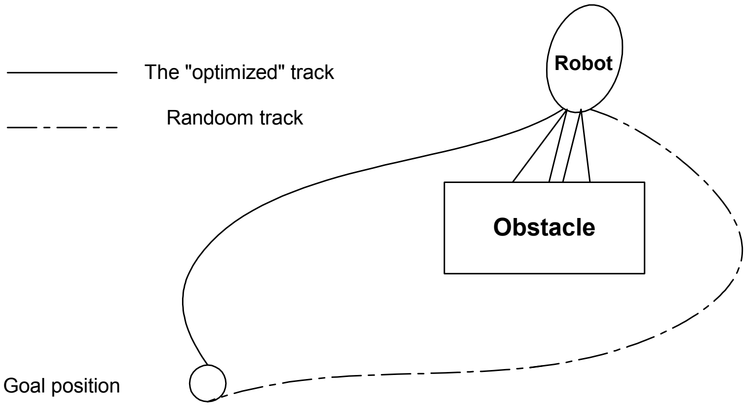


Fig. 12. Fusion of obstacle avoidance and goal reaching behaviors.

In the experimental work, the functionality of the proposed scheme was tested in an environment mimicking a crowded dynamic environment. The environment was designed taking into account several situations such as: simple corridor, narrow space and an area with many different shapes as obstacles (round, rectangle, trapezoidal)in away that mimics an image of office or equipped room (indoor environment).

In experiment (1) Fig.13 the robot has to reach Goal 1 for the start point placed between two walls. The robot begins to execute the behavior according to the rule base of the supervision layer depending on the current context. First, the robot follows wall 1 with maximum velocity until it senses obstacle 1, then it changes its behavior to obstacle avoidance at point A up to point B during which the robot crosses a narrow space between obstacle 1 and wall 2. The goal reaching behavior and the obstacle avoidance behavior are activated separately or fused to leave the narrow space, until point C. The robot then was encountered by obstacle 2, wall 2 and wall 3, so the emergency situation behavior was active. Next, the presence of obstacle 2 in front of the robot makes the obstacle avoidance behavior active until point D. For the route between of point D to E the robot just follows wall 2. From point E, three behaviors are activated (wall following, obstacle avoidance and goal reaching). The wall follow behavior is activated when a medium distance between the robot and obstacle 4 and 5 is established(corresponding to the last two cases in Table 5). The goal reaching behavior is activated to guide the robot to the goal. Between point F and G the robot is situated far from obstacles which in turn makes the goal reaching behavior active up to the presence of obstacle 6 at which point both goal reaching and obstacle avoidance are active. Finally, the goal reaching behavior is activated to reach the goal when it is near to the robot.

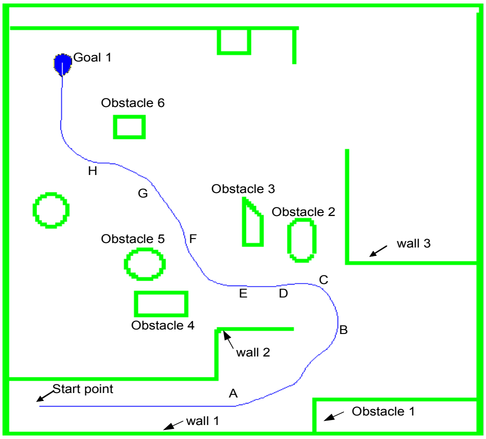


Fig. 13. Experiment 1.

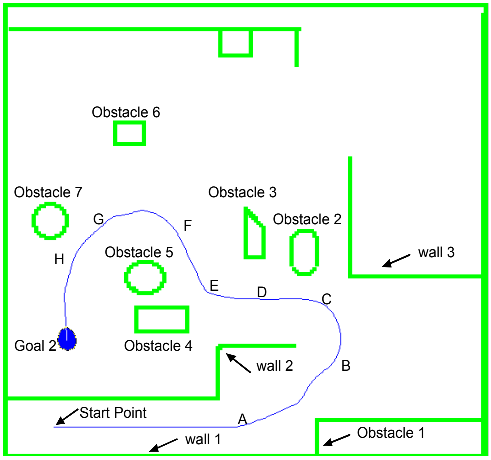


Fig. 14. Experiment 2.

For the experiment (2) Fig.14 the robot had to reach goal 2. It uses the same path as experiment 1 up to the point C. Then, avoiding obstacle 2 and goal reaching behavior are activated up to point D. From this point the robot follows obstacle 3 as a wall up to point E. The presence of obstacle 4 changes the behavior of the robot to obstacle avoidance. The robot avoids obstacle 4 then it follows obstacle 5 as wall by keeping medium distance to it. After that, the robot is situated in a free space and goal reaching behavior is activated up to point G. The obstacle avoidance behavior is activated when the robot senses obstacle 7 the robot avoids obstacle 7 and the behavior is changing to wall following when a medium distance to obstacle is measured by the robot. Finally the robot reaches the goal.

Figs.15-16 illustrate the levels of activation of each behavior during experiment 1 and experiment 2, respectively.

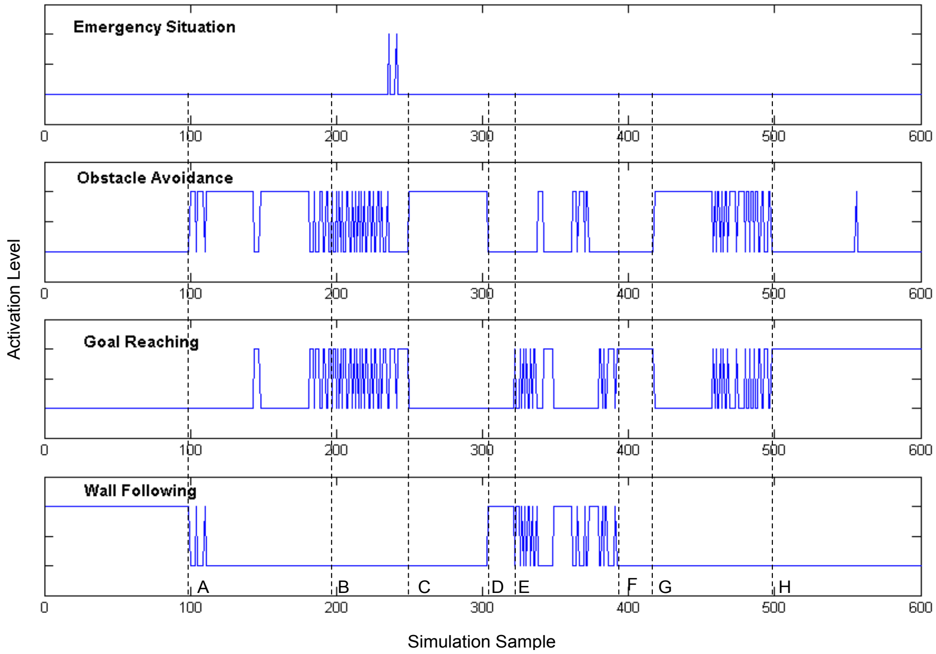


Fig. 15. Levels of activation of each behavior during Experiment 1.

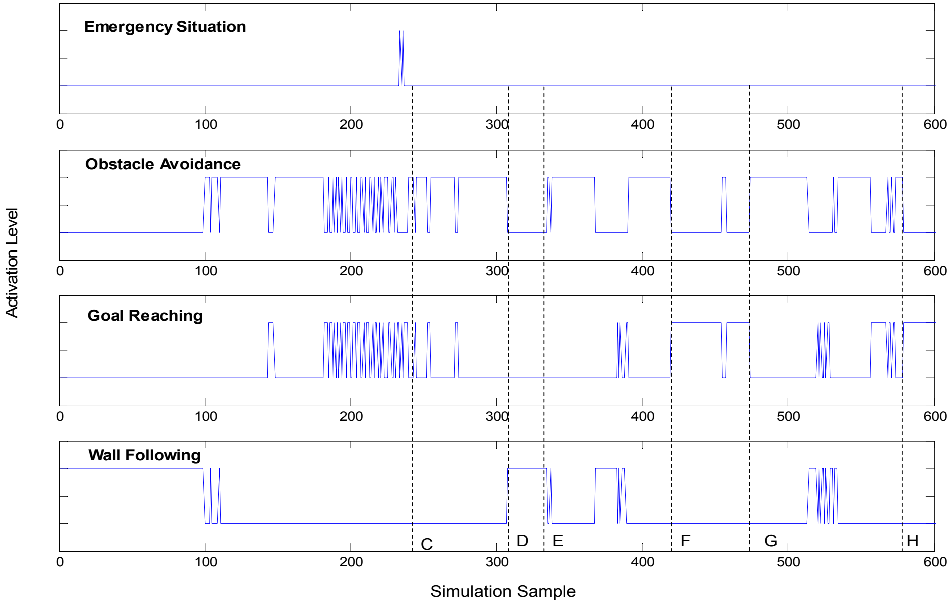


Fig. 16. Levels of activation of each behavior during Expriment 2.

The aim of experiment (3) (see Fig.17) is to show the ability of the robot to manage to escape from U shape obstacle and reaching a goal between two walls. From the start point the robot is situated in a U shape. In this context three behaviors are activated (the goal reaching behavior, obstacle avoidance and wall following). The avoid obstacle behavior always guided by the activation of goal reaching behavior especially at point A and B as shown in Fig.18.

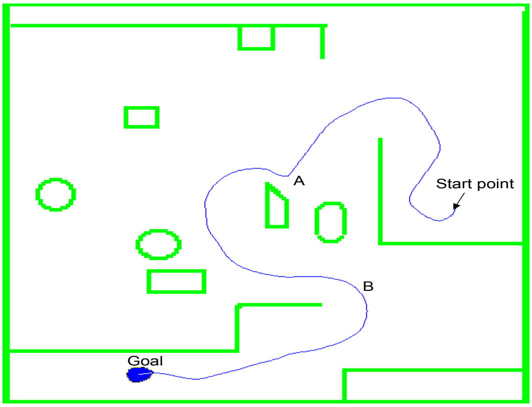


Fig. 17. Expriment 3.

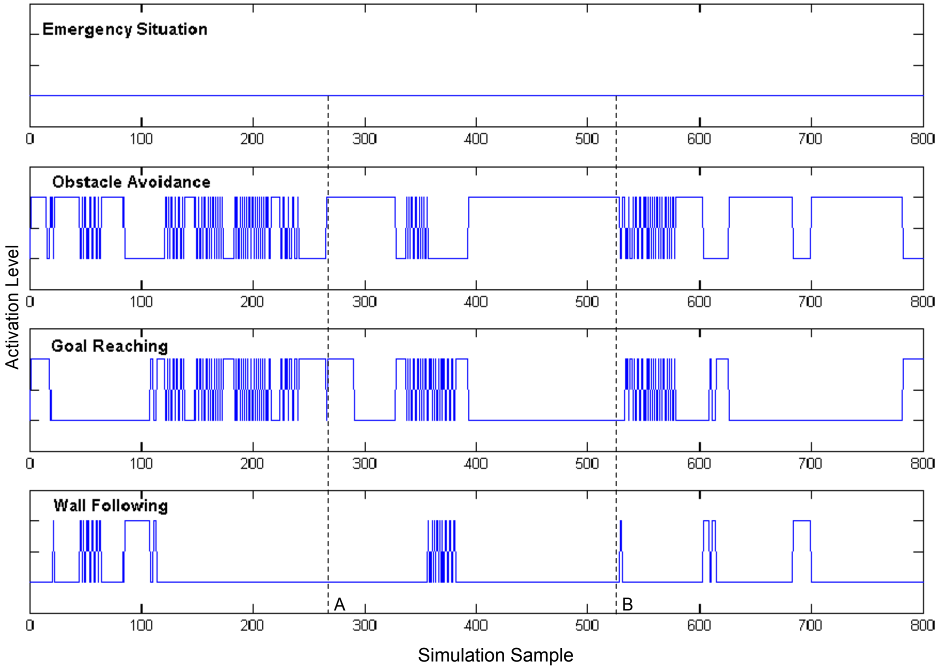


Fig. 18. Levels of activation of each behavior during Experiment 3.

Experiment (4) (Fig.19) shows the ability of the robot to escape from a trap situation and searching another path for reaching the goal.

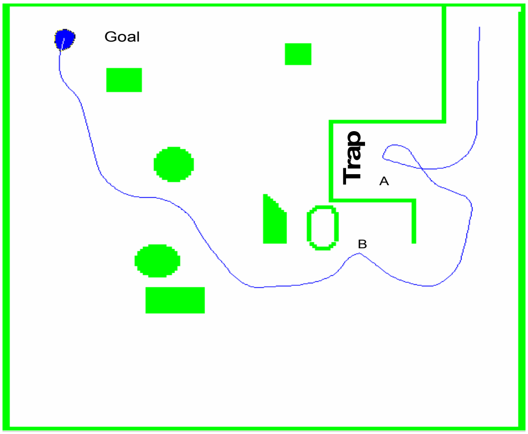


Fig. 19. Expriment 4.

After crossing the corridor, no obstacles are in front of the robot. Thus, the robot goes to the direction of the goal. At point A the robot senses the existence of obstacles around it, in the front, left and right, and then it made a round to right to escape from this trap and continues its navigation looking for another path.

In point B, the activation of goal reaching behavior and obstacle avoidance behavior is observed to take place concurrently. Here, the supervision layer fused these 2 behaviors to get the appropriate action as shown in section 3.3 Fig.12. The orientation of the robot depends on the goal position.

The Fig.20 shows the level of activation of emergency situation in the point A and the activation of goal reaching behavior and obstacle avoidance at the point B.

3.5 Experimental work

The effectiveness of the suggested navigation approach was experimentally demonstrated on a robotic platform named Pekee (Pekee™ robot is an open robotic development toolkit of Wany Robotics).

3.5.1 Pekee mobile robot

Pekee is equipped with two driving wheels with an additional supporting wheel. Its length is 40 cm and width is 25.5 cm, max speed 1 meter/second rotation 360 degree in a circle of 70 cm. The velocities of driven wheels are independently controlled by a motor drive unit. In addition the robot is endowed by,

- 2 odometers (180 impulses/wheel-turn).
- 2 gyro meters (pan and tilt)
- 2 temperature sensors.
- 1 variable frequency buzzer.
- Infrared link for communication between robots and peripherals.
- Serial infrared link for data transfer between Pekee and docking station or PC.

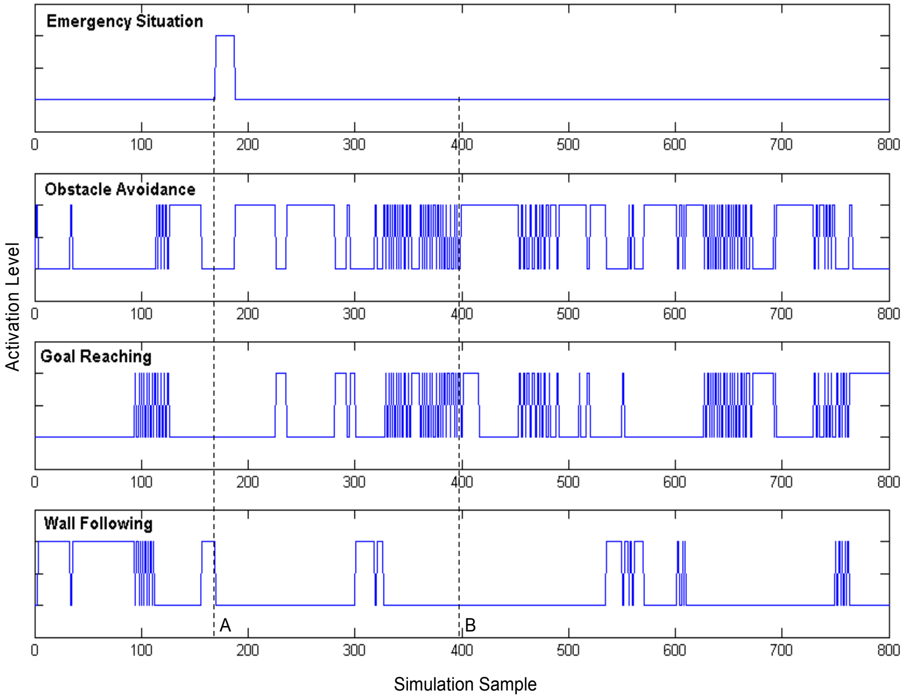


Fig. 20. Levels of activation of each behavior during expriment 4.

- Embedded Pc X86
- Camera
- 16-Mhz Mitsubishi micro-controller (16-bit), with 256 KB Flash-ROM and 20 KB RAM.
- 15 infrared sensors infrared telemeters (up to 1 measurement) arranged as shown in Fig21

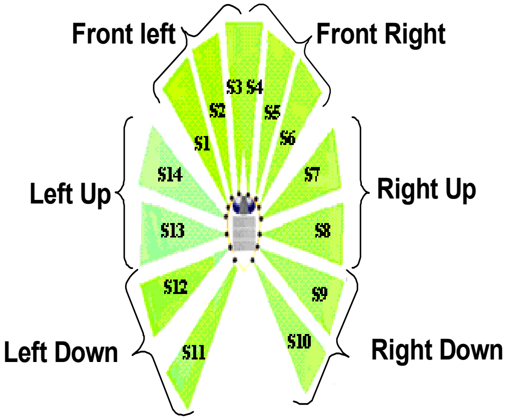


Fig. 21. IR sensors arrangement.

3.5.2 Pekee to PC communication

In the experimental work, the robot and the PC are connected via TCP/IP protocol using 2 network wireless cards one on the robot and the other on the computer or using a RJ45 cable.

The PC is equipped by a 54G network card compiles with the IEEE 802.11b standard in order to communicate with other 802.11b compliant wireless devices at 11 Mbps (the robot wireless network card). The card runs at speed of up to 54Mbps and operates on the same 2.4 GHz frequency band as 802.11b WI-FI products. This frequency band is suitable in industrial, science, and medical band operation. The work space of this card is as maximum 300 m.

The PC compiles, links the source code and executes the program. Then, it transmits the frames to the robot embedded PC via TCP\IP. These frames will be transmitted to the micro controller via the OPP bus. The micro controller is responsible to execute frames and transmits order to actuators (motors, buzzer…) also it sends data about the robot sensors status, robot position, measured distances ….

3.5.3 Experimental results

In order to validate the results of simulated experiments, a navigation task has been tested in a real world as an environment similar to a small equipped room. Fig. 22-26 shows that Pekee was able to navigate from a given starting point to a target point while avoiding obstacles.

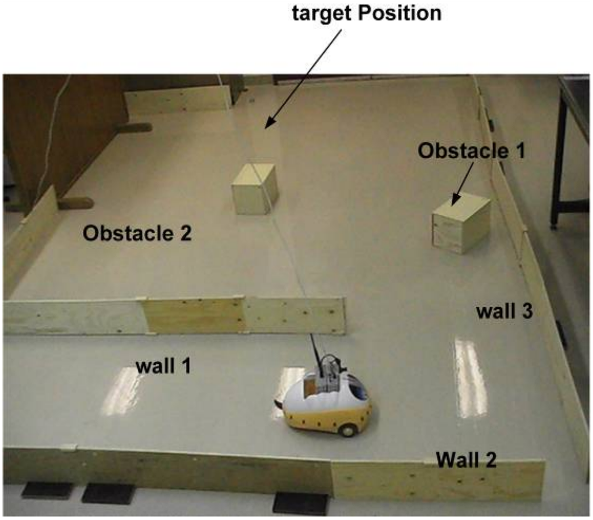


Fig. 22. Pekee Navigates in a crowded environment.

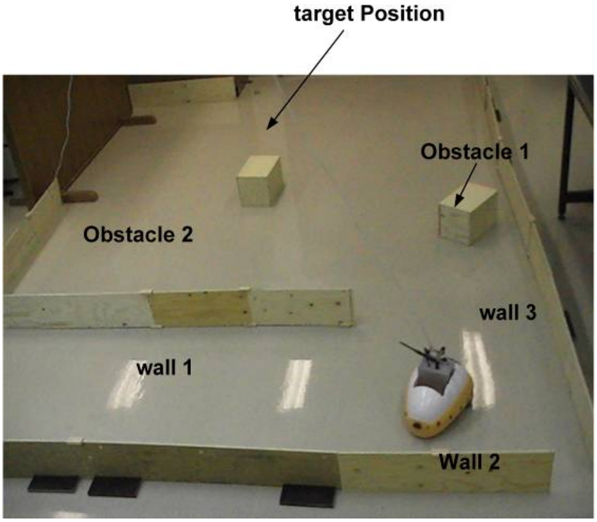


Fig. 23. Pekee Navigates in a crowded environment.

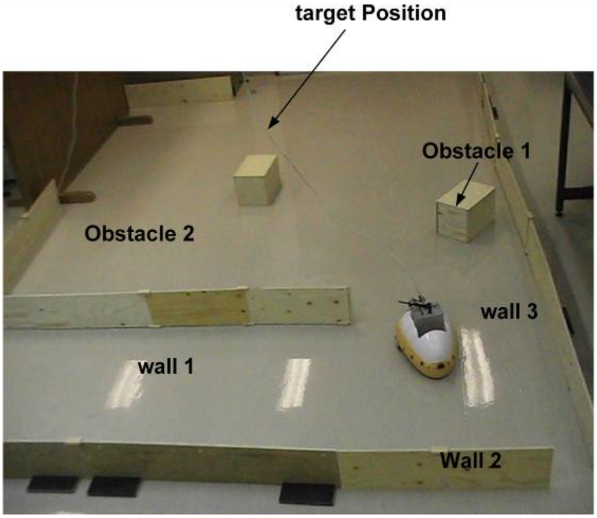


Fig. 24. Pekee Navigates in a crowded environment.

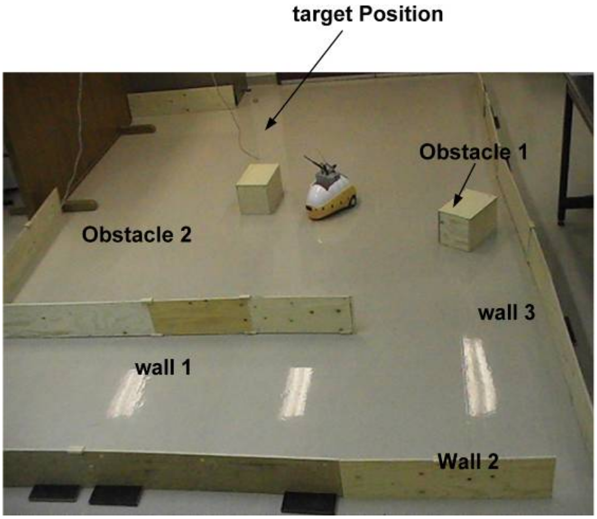


Fig. 25. Pekee Navigates in a crowded environment.

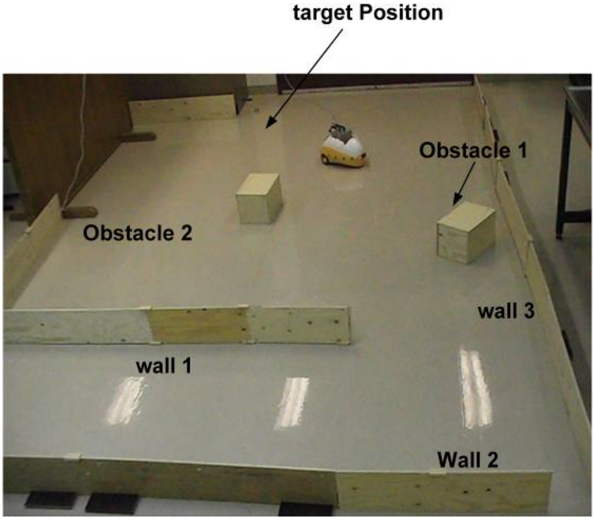


Fig. 26. Pekee Navigates in a crowded environment.

3.6 Conclusions

A successful way of structuring the navigation task in order to deal with the problem of mobile robot navigation is demonstrated. Issues of individual behavior design and action coordination of the behaviors were addressed using fuzzy logic. The coordination technique employed in this work consists of two layers. A Layer of primitive basic behaviors, and the supervision layer which based on the context makes a decision as to which behavior(s) to process (activate) rather than processing all behavior(s) and then blending the appropriate ones, as a result time and computational resources are saved. Simulation and experimental studies were done to validate the applicability of the proposed strategy.

As an improvement to our implemented system we are planning to incorporate learning to improve the effectiveness of the navigation approach.

As a further work the algorithm is to be tested on a “robotic cane”, which is a device to help the blind or visually impaired users navigate safely and quickly among obstacles and other hazards. During operation, the user will be able to push the lightweight “Robotic Cane” forward. When the “Robotic Cane’s ultrasonic sensors detect an obstacle, the behavior based navigation that will be developed determines a suitable direction of motion that steers the user around it. The steering action results in a very noticeable force felt in the handle, which easily guides the user without any conscious effort on his/her part. The navigation methodology that will be followed in the robotic cane will be based on the behavior based navigation developed herein, in particular the work will use fuzzy logic based navigation scheme to steer the “Robotic Cane”. Further the user is continually interacting with the robotic cane, this calls for the need to address the issue of cooperation and/or conflict resolution between the user and the robotic cane.

4. References

Aguirre E. & Gonzales A. (2000). Fuzzy behaviors for mobile robot navigation:design, coordination and fusion, *Int. J. of Approximate Reasoning*, Vol. 25, pp. 255-289.

Althaus P. & Christensen H. I.(2002). Behavior coordination for navigation in office environment, *Proceedings of 2002 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pp. 2298-2304, Switzerland, 2002

AlYahmedi, A. S., El-Tahir, E., Pervez, T. (2009). Behavior based control of a robotic based navigation aid for the blind, *Control & Applications Conference, CA2009*, July 13-July 15, 2009, Cambridge, UK.

Arkin, R. C. (1987). Towards Cosmopolitan Robots: Intelligent Navigation in Extended Man-made Environments, *PhD Thesis*, University of Massachusetts, Department of Computer and Information Science.

Arkin, R. C. (1989). Motor schema-based mobile robot navigation, *Int. J. of Robotic Research*, Vol 8, pp. 92-112.

Arkin, R. C. & Balch, T.(1997) AuRA: Principles and Practice in Review, *Journal of Experimental and Theoretical Artificial Intelligence(JETAI)*, Vol. 9, No. 2/3, pp. 175-188.

Brooks R. A.(1986). A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, Vol. 2, No. 1, (March 1986), pp. 14–23.

Brooks R. A.(1989). A Robot that Walks; Emergent Behavior from a Carefully Evolved Network, *IEEE International Conference on Robotics and Automation*, Scottsdale, AZ, pp. 292–296, May 1989

Ching-Chih, T., Chin-Cheng, C., Cheng-Kain, C, Yi Yu, L. (2010). Behavior-based navigation using heuristic fuzzy kohonen clustering network for mobile service robots, *International Journal of Fuzzy Systems, Vol. 12, No. 1, March, 2010* , pp. 25-32.

Fatmi, A., ALYahmedi, A. S., Khriji, L., Masmoudi, N(2006). A fuzzy logic based navigation of a mobile robot, *World academy of science, Engineering and Technology*, issue 22, 2006, pp. 169-174.

Huq, R., Mann, G. K. I., Gosine, R. G.(2008). Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation, *Applied soft computing*, 8, 2008, pp. 422-436.

Langer D., Rosenblatt J.K. & Hebert M. (1994). A Behavior-Based System For Off-Road Navigation, *IEEE Journal of Robotics and Automation* , Vol. 10, No. 6, pp. 776-782.

Maes P. (1990). How to do the Right Thing, *Connection Science Journal, Special Issue on Hybrid Systems,* Vol. 1.

Mataric´ M. J.(1997). Behavior-Based Control: Examples from Navigation, Learning, and Group Behavior, *Journal of Experimental and Theoretical Artificial Intelligence,* special issue on Software Architectures for Physical Agents, Vol. 9, No.2/3, pp. 323-336

Rosenblatt J. & Payton D. W.(1989). A Fine-Grained Alternative to the Subsumption Architecture for Mobile Robot Control, *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks*, Washington DC, June 1989, vol. 2, pp. 317-324.

Rosenblatt J.(1995). DAMN: A Distributed Architecture for Mobile Navigation, *Ph.D. dissertation, Carnegie Mellon University Robotics Institute Technical Report CMU-RI-TR-97-01*, Pittsburgh, PA, 1995.

Saffiotti A.(1997). The uses of fuzzy logic for autonomous robot navigation: a catalogue raisonn'e, *Soft Computing Research journal,* Vol. 1, No. 4, pp. 180-197.

Selekwa M. F., Damion D., & Collins, Jr. E. G. (2005). Implementation of Multi-valued Fuzzy Behavior Control for Robot Navigation in Cluttered Environments, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, pp., 3699-3706, April 2005

Seraji H. & Howard A.(2002). Behavior-based robot navigation on challenging terrain: A fuzzy logic approach, *IEEE Trans. Rob. Autom.* Vol. 18, No. 3, pp. 308-321

Seraji H., Howard A. & Tunstell E.(2001). Terrain-Based Navigation of Planetary Rovers: A Fuzzy Logic Approach, *Proceeding of the 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space*, Canada, June 18-22, 2001.

Tunstel E., Lippincott T. & Jamshidi M.(1997). Behavior Hierarchy for Autonomous Mobile Robots: Fuzzy-behavior modulation and evolution, *International Journal of Intelligent Automation and Soft Computing, Special Issue: Autonomous Control Engineering at NASA ACE Center*, Vol. 3,No. 1,pp. 37--49.

Yang S. X., Li H., Meng M. Q.-H , & Liu P. X. (2004). An Embedded Fuzzy Controller for a Behavior-Based Mobile Robot with Guaranteed Performance, *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 4, pp.436-446.

Yang S. X., Moallem M., & Patel R. V. (2005). A Layered Goal-Oriented Fuzzy Motion Planning Strategy for Mobile Robot Navigation, *IEEE transactions on systems, man, and cybernetics – part b: cybernetics*, Vol. 35, no. 6, 1214-1224.

View publication stats