JJMIE

Volume 3, Number 1, March. 2009
ISSN 1995-6665
Pages 77 - 88

Jordan Journal of Mechanical and Industrial Engineering

A Neuro-Fuzzy Reasoning System for Mobile Robot Navigation

Khaldoun K. Tahboub [a,*], Munaf S. N. Al-Din [b]

[a] Industrial Engineering Department, University of Jordan, Amman, Jordan.

[b] Department of Electrical Engineering, Tafila Technical University,Tafila, Jordan.

Abstract

An Autonomous Mobile Robot is an artificially intelligent vehicle capable of traveling in unknown and unstructured environments independently. Among the proposed approaches in the literature to handle the navigation problem of a mobile robot is the simple fuzzy reactive approach. This approach, however, occasionally suffers from two major problems, i.e., escaping from trap situations and the combinatorial explosion of the if-then rules in the inference engine. This paper presents a neuro-fuzzy reasoning approach for mobile robot navigation. The proposed approach has the advantage of greatly reducing the number of if-then rules by introducing weighting factors for the sensor inputs, thus inferring the reflexive conclusions from each input to the system rather than putting all the possible states of all the inputs to infer a single conclusion. Four simple neural networks are used to determine the weighting factors. Each neural network is responsible for determining the weighting factor for one sensor input. Simulation results are presented to demonstrate the merits of the proposed system.

Keywords: Neural networks; Fuzzy systems, neuro-fuzzy reasoning; Mobile robot navigation

1. Introduction

The ability of a mobile robot to navigate in unknown and unstructured environments by relying only on its sensory system is regarded as the key issue in an enormous number of research publications during the past 15 years [1]. In general sensor-based data acquired by the mobile robot presumably provides the necessary information to determine the appropriate control actions to the actuators so that the mobile robot can travel safely in cluttered environments with static and/or moving obstacles. In order to achieve its goal, the robot is usually required to determine in real-time a safe and smooth path from a starting location to an end location (target). Consequently, the main issues that need to be addressed in mobile robot navigation are reactive obstacle avoidance, and target acquisition [2]. It is well known that classical robot control methods that are based on precise models are only appropriate for industrial mobile robots that are designed to perform simple tasks and operate in structured and known environments. However, uncertainty is a major problem in mobile robot navigation process, and a robot is expected to deal and react robustly with present environment.

The evolvement of soft-computing paradigms have provided a powerful tool to deal with mobile robot navigation process, which exhibits incomplete and uncertain knowledge due to the inaccuracy and imprecision inherent from the sensory system [3,4]. Among all the soft-computing methods fuzzy logic based decision-making and neural networks have been found to be the most attractive techniques that can be utilized for this purpose. Fuzzy system is tolerant to noise and error in the information coming from the sensory system, and most importantly it is a factual reflection of the behavior of human expertise. In general, there are two approaches to the application of fuzzy logic in mobile robot navigation, namely, behavior-based approach [5-9] and classical fuzzy rule-based approach [1, 10-17]. However, the design of fuzzy logic rules is often reliant on heuristic experience and it lacks systematic methodology, therefore these rules might not be correct and consistent, do not possess a complete domain knowledge, and/or could have a proportion of redundant rules. Furthermore, these fuzzy logic rules can not be adjusted or tuned on real-time operation, and the off-line adjustment of their parameters is a time consuming process. Another problem could be raised when better precision is needed which is the huge expansion in the fuzzy rule-based system. Several approaches have been proposed in the recent literature to approach the above problems. A new grid-based map model called "memory grid" and a new behavior-based navigation method called "minimum risk method" was proposed by [18]. An integrated fuzzy logic and genetic algorithmic approach was presented by [19]. A hybrid controller that includes a support vector machine and a fuzzy logic controller was proposed by [20].

On the other hand, several successful reactive navigation approaches based on neural networks have been suggested in the literature [21-27]. In spite of the different suggested network topologies and learning methods,

* Corresponding author : ktahboub@ju.edu.jo

neural reactive navigators perceive their knowledge and skills from demonstrating actions. Therefore, they suffer from a very slow convergence, lack of generalization due to limited patterns to represent complicated environments, and finally information encapsulated within the network can not be interpreted into physical knowledge.

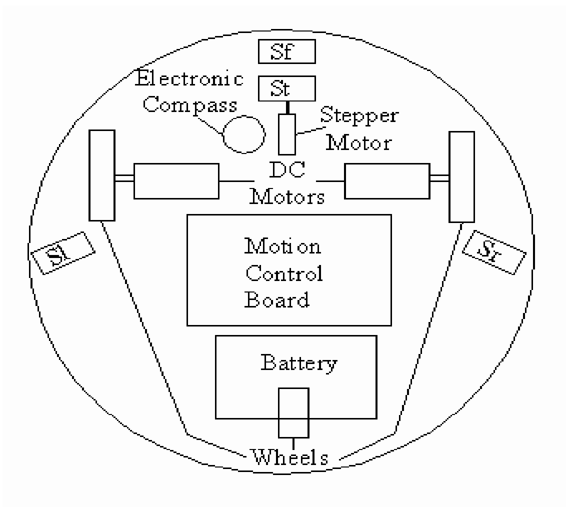
Recently the role of neural networks has been found to be very useful and effective when integrated with fuzzy control systems to produce what is called Neuro-fuzzy systems, and sometimes called fuzzy neural networks. Neuro-fuzzy systems provide an urgent synergy can be found between the two paradigms, specifically the capability to mimic human experts as in fuzzy logic, and learning from previous experience capability as in neural networks. In general, neuro-fuzzy systems can be classified into two categories, adaptive neuro-fuzzy inference system and hybrid neuro-fuzzy systems. The first category is the most widely used neuro-fuzzy systems, and they are designed to combine the learning capabilities of neural networks and reasoning properties of fuzzy logic. The main function of neural network is to learn about the fuzzy inference system behavior and uses this knowledge to adaptively modify its parameters [ANFIS, and others]. The adaptability of the fuzzy inference system can be achieved by either rule base modification and/or membership functions modifications. Rules can be generated, modified, and/or eliminated, while membership functions of the input variables can adjusted and tuned by scaling mechanism. The basic idea behind the use of the second category is to replace all or parts of the basic modules that builds a FIS. The only advantage that can be gained from such arrangements is the high processing speed, presuming that a hardware implementation of such neural networks exists.

In this paper a new approach to the design of a simple hybrid neuro-fuzzy navigation system is described. The suggested system has two apparent advantages. First, the if-then rule base is replaced by a set of simple neural networks. Second, inference is on the reflexive conclusions from each input to the system, rather than putting all the possible states of all the inputs to infer a single conclusion. Four parallel simple neural networks are utilized to generate weighting factors for the distance readings acquired by the robot’s sensory system. These weighting factors represent the degree of collision avoidance by the robot from a certain side. These weighting factors are then treated as fuzzy values that are input to a defuzzifier to come up with a crisp value for the robot’s steering angle and speed.

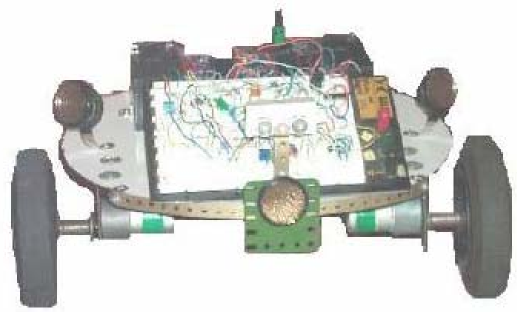
2. Experimental Prototype

SALIM, Simple Autonomous LIght weight Mobile robot, which was constructed at the authors’ universities, has been used to conduct practical experiments. SALIM has a cylindrical shape with a radius of 30 cm, and travels at a maximum speed of 8m/min. The robot has two independent wheels, driven by geared PM DC motors, located at the ends of an axis near to one of the ends of the circular base, and one free caster at the other end of the base. Such arrangement provides a simple and effective differential-velocity steering control by varying the applied voltage to the motors. The motion control of the two PM DC motors is accomplished by a simple motion control board designed by the authors, which consists a full bridge chopper circuit, and PIC16f877 micro-controller. The advantage of using this micro-controller is that it accepts velocity commands from a remote computer and to control two DC motors independently.

Three groups of ultrasonic sensors are mounted at the front, and at the two ends of the central axis of the robot, where the right and the left sensors are directed at 45° from the central axis as shown in Figure 1. Target's orientation with respect to the center of the robot is obtained by an electronic compass. The actual angle between the robot frontal axis and the target can be found by simple manipulation to the robot's heading angle, which is updated instantaneously by the microcontroller, and that measured by the electronic compass. According to instantaneous value of this angle another ultrasonic sensor is utilized to detect the existence of any obstacle in the virtual target direction. This sensor is allowed to rotate, using a small stepper motor, in the range (-5° to 5°) with respect to the frontal axis of the robot. The reason in mounting the ultrasonic sensors in such arrangement will be mentioned in section (3). The error eliminating rapid ultrasonic firing (EERUF) method is used to minimize the error in distance measurements due to the noise that affect the ultrasonic sensors, and the crosstalk problem was eliminated by using alternating delays method.



(a)



(b)

Figure 1: Schematic of SALIM with sensor locations, a, b .

3. Navigation Process

Autonomous mobile robots at least need to achieve a simple goal of traveling safely and purposefully from one location to another in an environment that is unstructured and subjected to unpredictable changes. Like human beings, AMR should be self-reactive in the real world through decisions produced by a real-time navigation system. The reactions to the perceived surrounding can be inferred from either reflexive behavior or logical behavior. In general, the robot navigation problem is decomposed mainly into goal reaching and/or obstacle avoidance problems.

A typical trajectory of a mobile robot when navigating through an environment with unknown obstacles cannot be generated by the reactions to the sensed obstacle alone, but the direction of the target with respect to the robot should be considered. Depending on the location of obstacles with respect to the robot together with the robot's orientation with respect to the target, the navigation system should be capable of generating the right decisions to enable the robot to perform the necessary maneuvers to avoid the obstacles and not losing its sense of orientation towards the target. If the target reaching and obstacle avoidance behavior to be integrated together, the reactions of the robot can be categorized into four main types. If the robot is moving and it is not sensing any obstacles in its vicinity, or the obstacles are not blocking the target, then it can be said that robot is in the *free-heading mode* Figure 2(a). The second scenario is called *partial-front blocking mode*. In this situation the lines of attraction force due to the target will be disturbed and bend over the obstacle, therefore the turning reaction of the robot will be towards the varying direction of these lines, even if it moves away from the target, until it passes the obstacle and change its mode to the free-heading reaction mode, as shown in Figure 2(b).

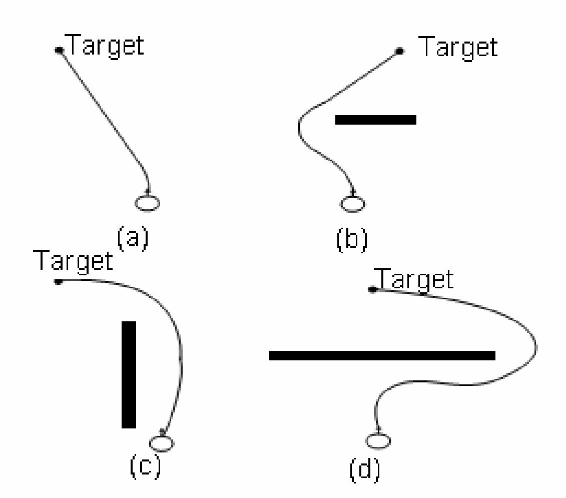


Figure 2: Typical Trajectory of a Mobile Robot While Avoiding Obstacles.

In the case of sensing an obstacle close to one of the robot sides that is blocking the straight-line path towards the target from the current position of the robot for some distance, as in the robot will be influenced by the side-blocking mode. In this case the lines of attraction force due to the target will be disturbed and bend over the obstacle as in the previous case except that it will follow the contour of the obstacle as shown in Figure 2(c). Therefore, the robot will move along this line while keeping a safe distance from the obstacle until it reaches the bent part of the line to change its mode to the free-heading mode. The final situation that might face the robot is when it is trapped due to the target attraction by a wide obstacle. The robot may make a significant turn to the left or to the right, and the robot here is under the total-front blocking mode. Once the robot turns to one of the two directions it will be then under the influence of side-blocking mode, and it will proceed in that direction while keeping a safe distance from the obstacle until it reaches the of obstacle’s end and again change its mode to the free-heading mode, as shown in Figure 2(d).

3.1. Navigation Methodology

When a mobile robot is traveling towards its final target it might face a variety of obstacles having different shapes and they may be randomly located in the path of the robot. Often in the literature, static obstacles can be classified into eight basic categories as shown in Figure 3. When fully conscious attention is paid to the environment, a navigation system could deal with large amounts of input information concerning near obstacles, and it should react instantaneously to provide a robust real-time reactions towards the foreseen surrounding. Therefore, its behavior should be how to avoid these obstacles, which simply can be answered move away from close obstacle by desired safety distance.

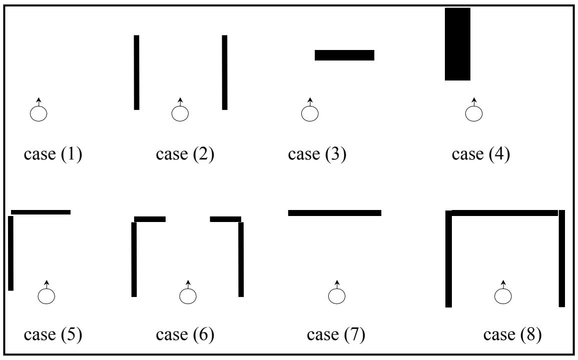


Figure 3: Classification of Obstacle Configurations.

Various algorithms have been proposed to attack the problem of generating collision free trajectories for a mobile robot by utilizing neuro-fuzzy systems. One of the methods used in designing a neuro-fuzzy navigation system, is based on training a neural network patterns of sensor readings corresponding to a variety of obstacles. Usually in such a methodology complete operator's experience is provided to the network and its training is supervised and performed off-line. Alternatively, partial operator's experience is provided to the network and its training is supervised and performed off-line at the first stage, and then navigator performance is enhanced by using on-line reinforcement learning. It should be noted that a robot may face during its course of navigation a variety of obstacles of different and complicated shapes that could be present in the surrounding, and they could be randomly located and oriented. Hence a huge number of patterns are required for the obstacle recognition methodologies, or a very long time will be required for unsupervised learning methodologies. ANFIS

From the above discussion, it is believed that it is better to consider the navigation process to be based on a very simple human experience to generate the reaction of the mobile robot towards the surrounding through neuro-fuzzy reasoning. Such system is based on two facts, first humans cannot get used to all the possible arrangements of obstacles and second cannot build a huge fuzzy model that contains all the possibilities of the 'If…Then…' rules. Instead, they give weighting factors to which direction they are going (left, ahead, right) and what their speed should be, and this possibly can be made based on information from the different senses. The neuro-fuzzy reasoning system suggested in this work, see figure 5, consists of a neural network/s responsible for generating independent certainty weighting factors for the three basic directions (left, ahead, and right) corresponding to instantaneous sensory information. These decisions are then combined, with the same level of simplicity by a diffuzifier, to obtain a final conclusion. The main objective of the proposed method is to reduce the size and time required by a fuzzy inference system by combining the learning capability in neural networks and reasoning capability in fuzzy inference systems, without affecting the efficiency and performance of the navigation system when compared to other classical implementations of reactive fuzzy and neuro-fuzzy navigators.

3.2. Learning Methodology and System Structure

For the first glance, the neuro-fuzzy system architecture was intended to have a multi-layer standard feed-forward neural network. The inputs to the NN are sonar data, representing the distribution of obstacles in foreseen surrounding, and the virtual angle between the robot and the target, while the output that should be produced from the network are certainty weighting factors for the three basic directions and a weighting factor for the target orientation. The training set was supposed to be obtained during driving sessions of the robot by a human operator in different situations, while sensed distances and the four weighting factors are to be stored during these sessions. Two factors resulted in total failure of such method. The first was the difficulties that faced all the operators to give four answers corresponding for the certainty weighting factors. The second factor is related to that a good quality of learning requires huge, significant and complete training examples. In these training examples human operators should guarantee the consistency of their reactions without any contradictions.

Under such difficulties an alternative learning approach was considered. The approach is based on a divide-and-concur strategy; where instead of having single multi-input neural network four three-layer neural networks were used. Each network is designed to receive only the distance measured by the corresponding group of ultrasonic sensors from the robot to any possible obstacle that may detected in that direction and generates a weighting factor that represents the degree of certainty to avoid the collision with the obstacles at that side. To generate the required data to train the neural networks a group of operators were required to answer a questionnaire asking them to represent their judgment to the measured distance and the degree of certainty weighting factor in a fuzzy format. Each measured distance was represented by five fuzzy values, Very Far (VF), Far (F), Medium (M), Close (C) and Very Close (VC), while the weighting factors were also represented by five fuzzy values, Very High (VH), High (H), Medium (M), Low (L) and very Low (VL). By averaging all the answers, the universe of discourse of both variables and their representation in terms of fuzzy sets were defined as shown in Figure 4. Consequently, a single input single output fuzzy system designed to provide a mapping function between the measured sonar distance and certainty weighting factor, from which a training data for the network training were obtained.

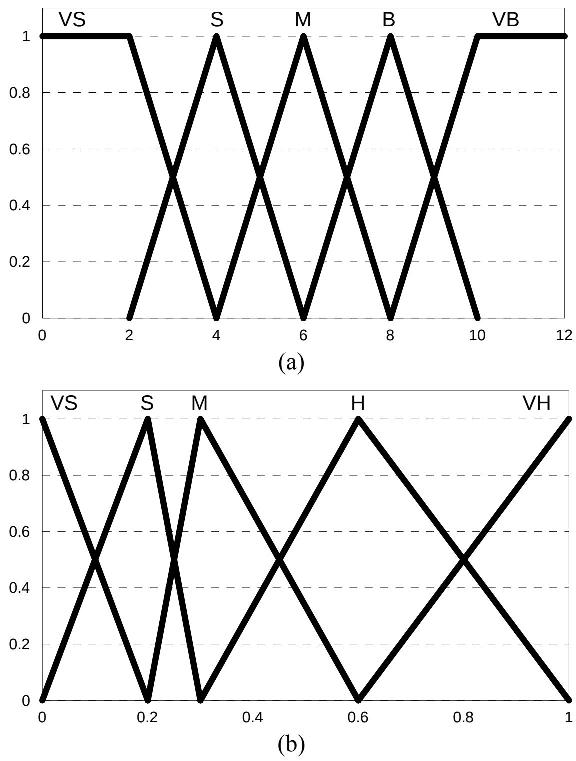


Figure 4: Fuzzy sets representations of (a) sonar measured distance, (b) weighting factor.

The structure of the proposed system is shown in Figure 5. Four input variables are required to provide the necessary information for the navigation system to safely drive the mobile robot to reach the desired target. These inputs are: distances d_f, d_r, and d_l, measured by three ultrasonic sensors. These distances are the distances between the robot and any possible obstacle with respect to the local front, right, and left directions of the robot, respectively. The forth input, d_{vt}, is the distance directed in a global virtual direction between the robot and the target. The outputs of the system are the steering angle θ and the speed of the robot ν. The idea of using a virtual target orientation instead of the real orientation comes from a realistic representation to the behavior of expert driver, where it is impossible for a driver to abandon his attention to the frontal sight when leaving a one-sided blocked target behind him and concentrates on the real target orientation. Under this situation the driver put some concentration towards a virtual orientation at the same side of the target, which should not exceed a certain limit in the range of the frontal sight.

Each distance variable from the corresponding sensor is then input to a simple neural network to generate a weighting factor that represents the degree of collision avoidance of the robot from the side of the corresponding sensor. The output weighting factors of each network were values between 0 and 1, with 0 value stating that the robot is very close to an obstacle and a value of 1 stating that the robot is very far from the obstacle. The structure of the four neural networks is identical and is depicted in Figure 6. Each network consists of a single input node, a single output node, and a single hidden layer with ten nodes. Back propagation has been used to train the networks. The middle block in the system is a simple defuzzifier that receives the four weighting factors coming from the previous neural network subsystems, and treats these factors as the degree of fulfillment for the corresponding fuzzy values of the steering angle of the robot. The Center of Area method is used in this block to obtain the final crisp value for the steering angle of the robot. The membership functions for the fuzzy values of the output variable θ are shown in Figure 7. Only three of the fuzzy values are shown in Figure 7, i.e., the turning angle to the left, center, and right, respectively. The fuzzy set that represents the steering angle towards the target orientation is similar except that it is designed to be floating with its center moving in the range [-30°, 30°].

Once the final value for the steering angle θ is obtained, the robot's speed can be computed by a two-input neural network as shown in Figure 8. The inputs to the neural network in Figure 8 are the steering angle θobtained from the previous stage and the distance d_t that represents the distance between the robot and the obstacles with respect to a virtual target. The input distance d_t is used to slow down the robot as it approaches the target. The training sets for the neural network in Figure 8 were obtained by simulating the robot's motions and estimating the required speeds of the robot for different values of d_t and θ.

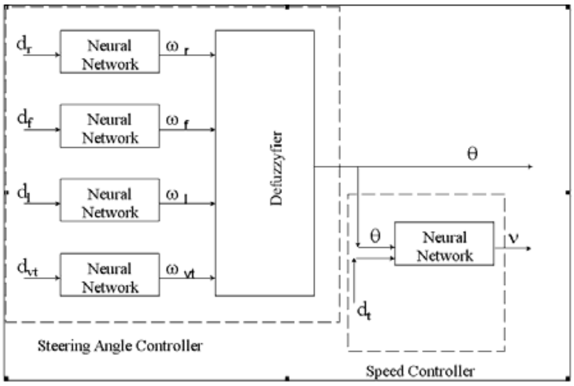


Figure 5: Structure of the proposed system.

4. Simulation Results

In order to confirm the efficiency of the proposed method, a simulation program with a graphical user interface has been developed on a Pentium III personal computer using Visual Basic 6. The robot is depicted in the simulation as a circle to resemble a prototype mobile robot that the authors have designed and constructed for experimental purposes. It is noted here, that errors due to

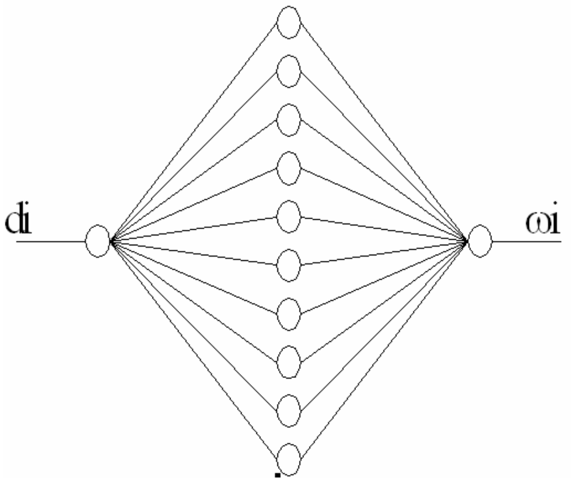


Figure 6: Neural network structure for weighting factors.

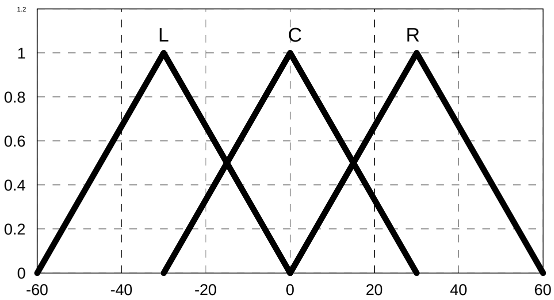


Figure 7: Fuzzy set definition for the output variable θ.

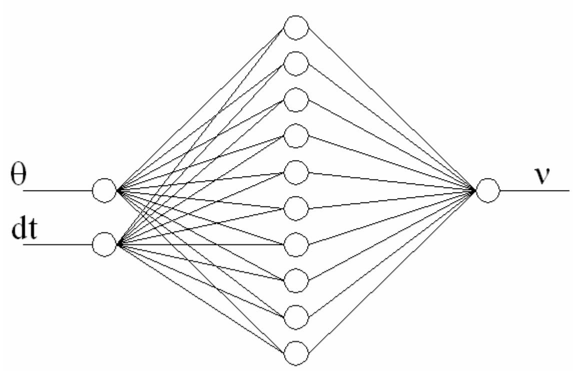


Figure 8: Neural network structure for speed calculation.

wheel slippage and other motion errors were not considered in the simulation. Distance measurements were acquired by four ultrasonic sensors mounted on the robot. Each sensor was modeled by a number of rays within a sector region of a wide beam-angle. The distance measured by each sensor is considered to be equal to the minimum distance obtained within the sector of each sensor while taking into consideration the minimum reliable distance that can be measured by actual ultrasonic sensors. Six different simulation cases are presented in this section to analyze the reaction behaviors of a mobile robot in avoiding a variety of unknown static obstacles placed randomly in a portion of an unknown environment. The aim here is to study the performance of the proposed approach under the most possible situations. In all these cases the robot is assumed to be initially moving with full

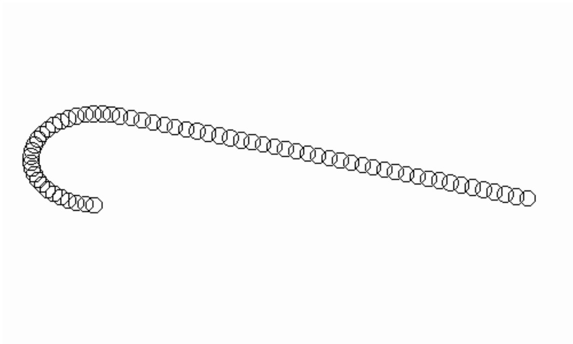


Figure 10: Case 1: Robot path with no obstacles.

speed and its relative steering angle is assumed to be zero. The analysis of the reaction behaviors of the robot is based on observing the instantaneous variation of the four weighting factors and their influence on both the steering angle and speed.

In the first case, Figure 10, the robot is initially oriented in an opposite direction to the target. In this case no obstacle is sensed by any of the four sensors. Hence, the values of the four weighting factors are all equal to 1 (see Figure 11). Consequently, the robot will be in the *free-heading mode*. The immediate reaction of the robot will be biased to turn towards the side at which the target sensor is located at that instant since the *Turn to Left* and *Turn to Right* sets are equally scaled. The variation of the steering angle and its influence on speed, Figure 12, depends on the location of the center of the *Turn to Target* set, which is allowed to move in the range [-30, 30] depending on which side the target is at that instant.

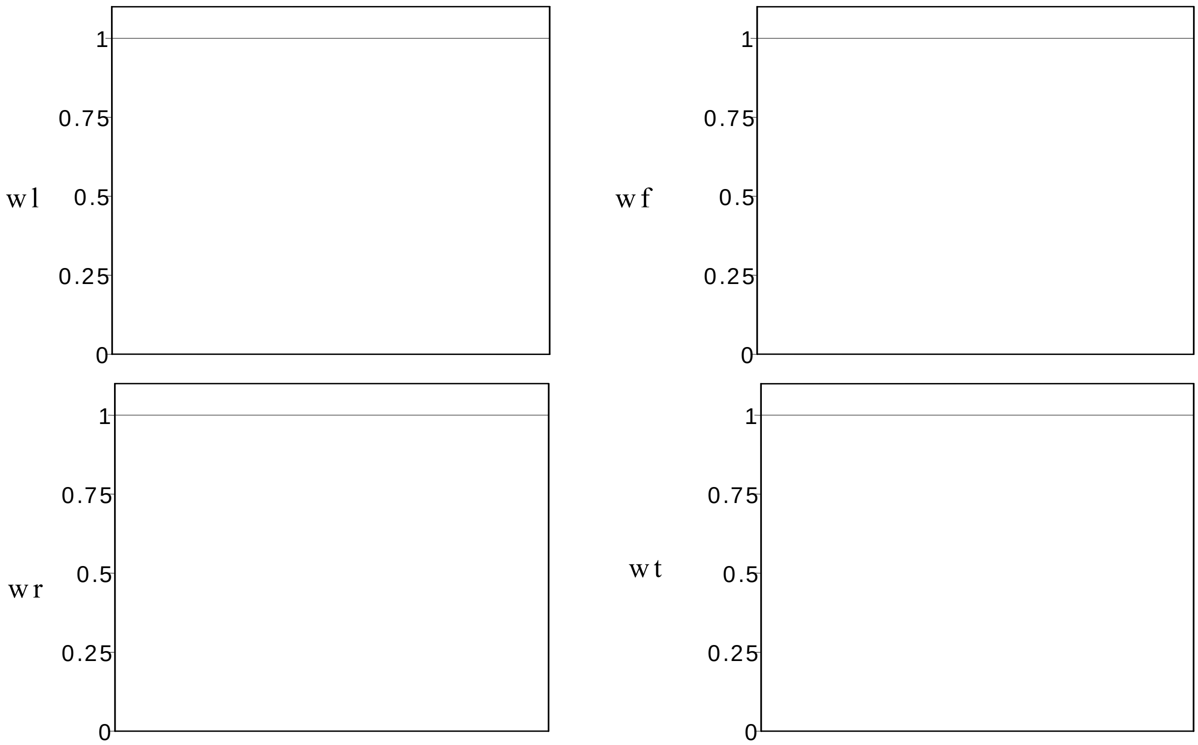


Figure 11: Behavior of weighting factors for case 1.

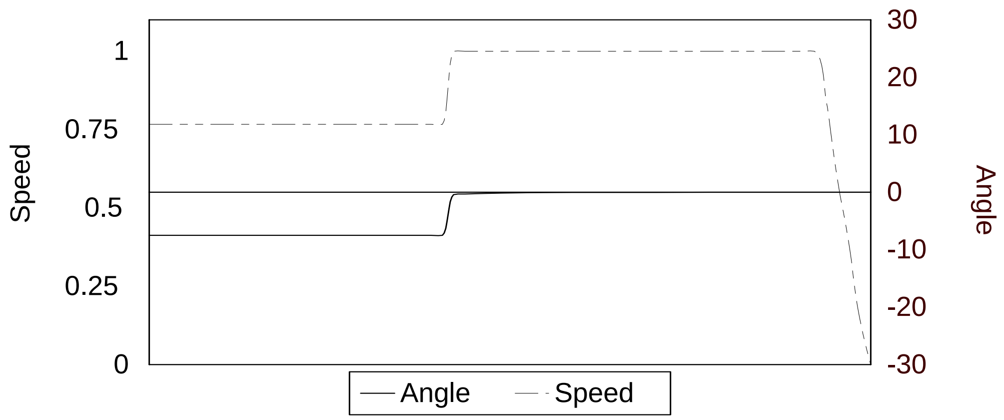


Figure 12: Behavior of steering angle and speed for case 1.

In the second case, Figure 13, the robot is passing through an empty L-shaped tunnel where the only present obstacles are the parallel bounding walls of the tunnel. The response of the robot towards the walls is influenced by the instantaneous variation of the weighting factors as shown in Figure 14. At the first instant the level of cautions towards the obstacles at both sides are equal, while the weighting factors for the front and target sensors indicate that there are no obstacles in either direction, thus the robot will move towards the target. As the robot proceeds in moving towards the target the steering angle will be gradually reduced because of the continuing increase in the difference between the right and left weighting factors and the fall of both front and target weighting factors (see Figure 15). Once the robot becomes close to the right wall the left weighting factor will rebalance the congregated right and target weighting

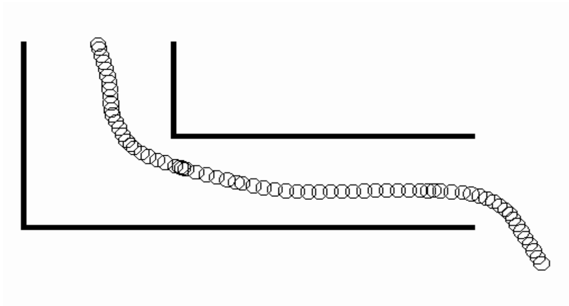


Figure 13: Case 2: Robot path through L-shaped tunnel.

factors. Thus, the robot will slightly turn to the left until it aligns itself to move later in parallel with the right wall. When reaching the end of the tunnel, the target weighting factor will rapidly increase to 1. Hence, the robot will noticeably reduce its speed for a short while until it is completely turned in the direction of the target.

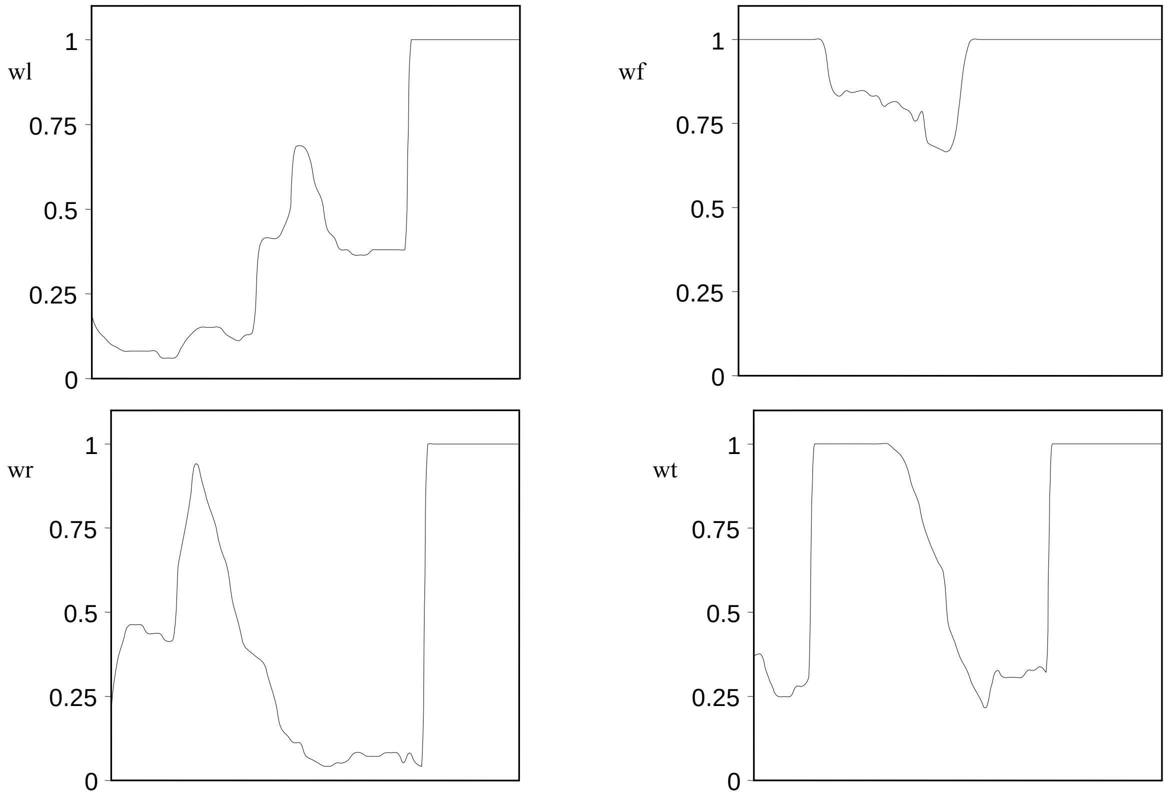


Figure 14: Behavior of weighting factors for case 2.

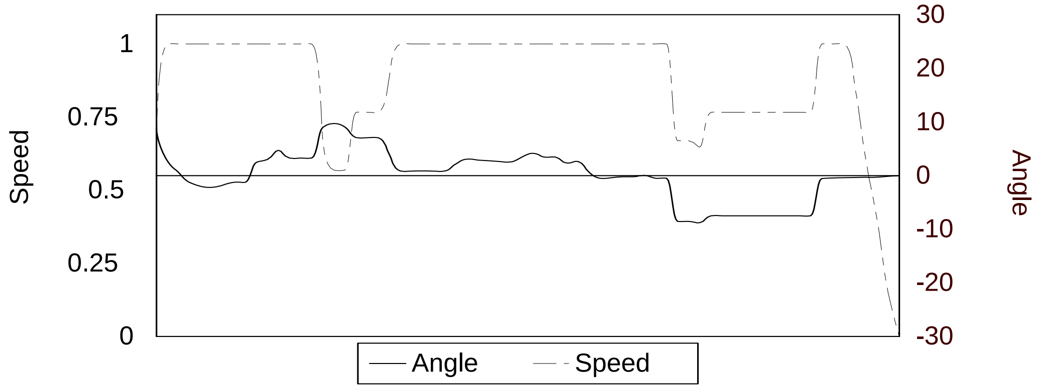


Figure 15: Behavior of steering angle and speed for case 2.

The third case, Figure 16, the robot is heading towards the target at full speed when a square obstacle blocks its path. The obstacle totally blocks the robot from the right side and slightly extended above the line that connects the location of the robot and the target. As can be seen in Figure 17 the first apprehension is from both the front and target sensors through the decreasing values of their corresponding weighting factors. The right sensor detects the presence of the obstacle, and the robot immediately reacts by turning gradually to the left while reducing its speed due to the dominance of the left weighting factor (see Figure18). Once the robot passes the obstacle both the front and the target weighting factors increase sharply. The right weighting factor follows and rises sharply to indicate the absence of any obstacle in all directions. Reacting immediately to this situation, the robot reduces its speed and turns to the right side to align itself again with the target direction (Figure 15).

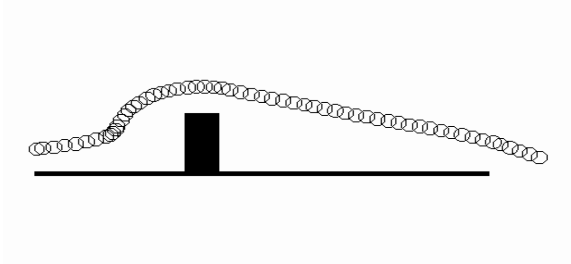


Figure 16: Case 3: Robot path blocked by a square obstacle.

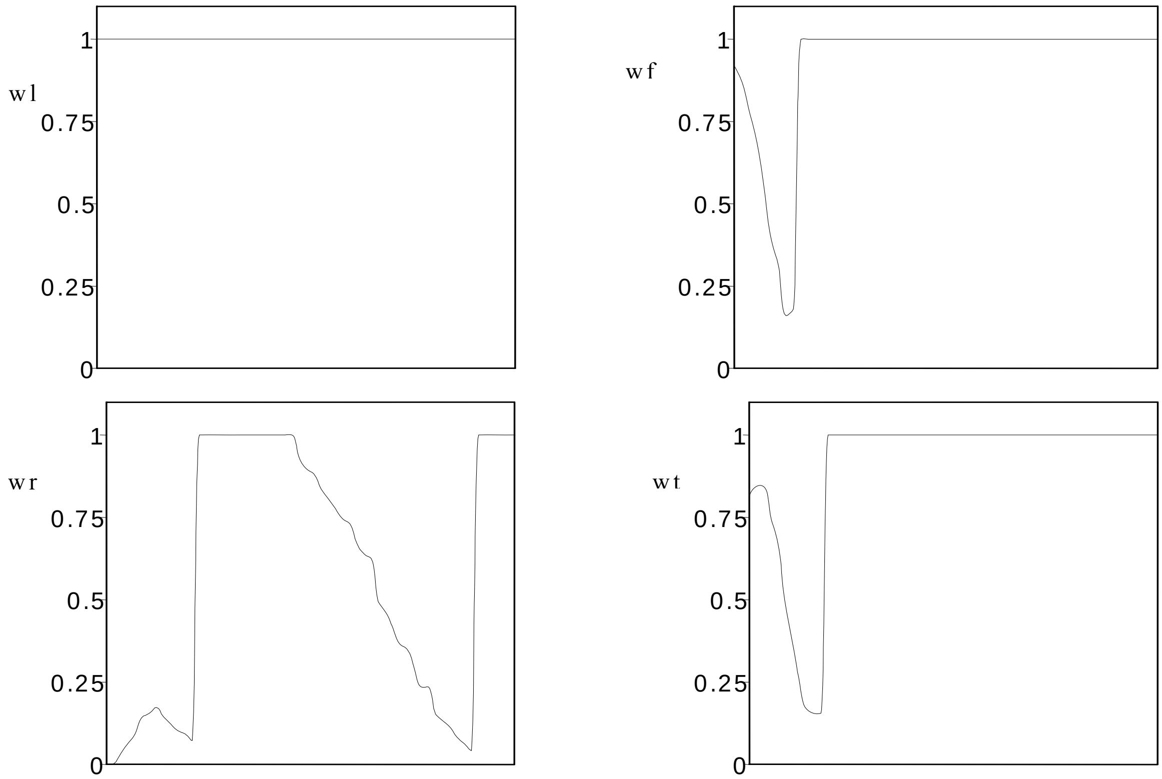


Figure 17: Behavior of weighting factors for case 3.

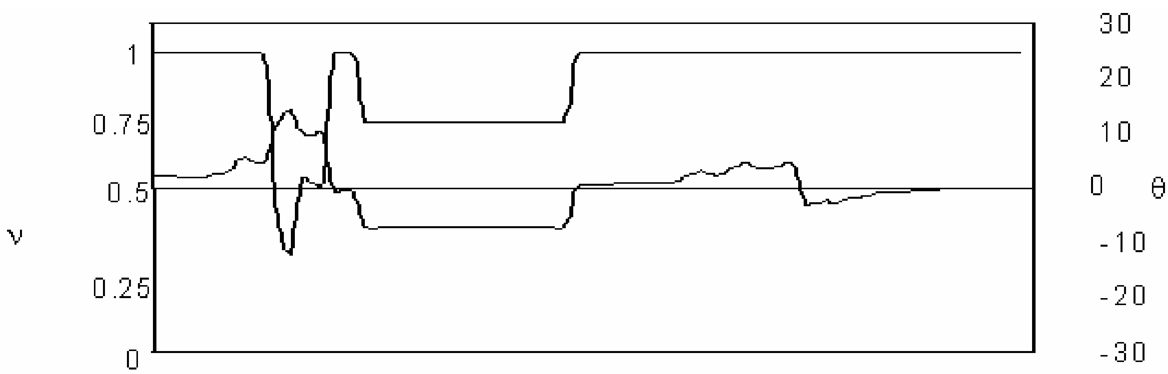


Figure 18: Behavior of steering angle and speed for case 3.

The next case presented is depicted in Figure 19. The case describes the situation when the robot is moving in a room while the target is in another room. The behavior of the robot in the beginning is similar to that in the previous case until the robot is faced with the corner walls. At this time only the right weighting factor is active will all other factors are zero. Consequently, the robot turns to the right. Once the robot turns to the right its path becomes blocked from both the left and front again. Hence, it keeps on turning to the right until the front weighting factor is active. The robot keeps on moving in the same direction until the right weighting factor is 1 at which time the robot turns back towards the target.

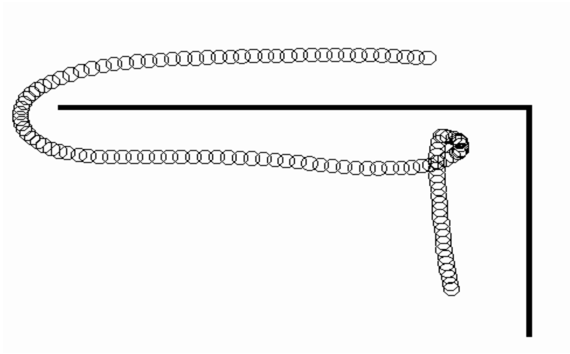


Figure 19: Case 4: Robot path when target is in another room.

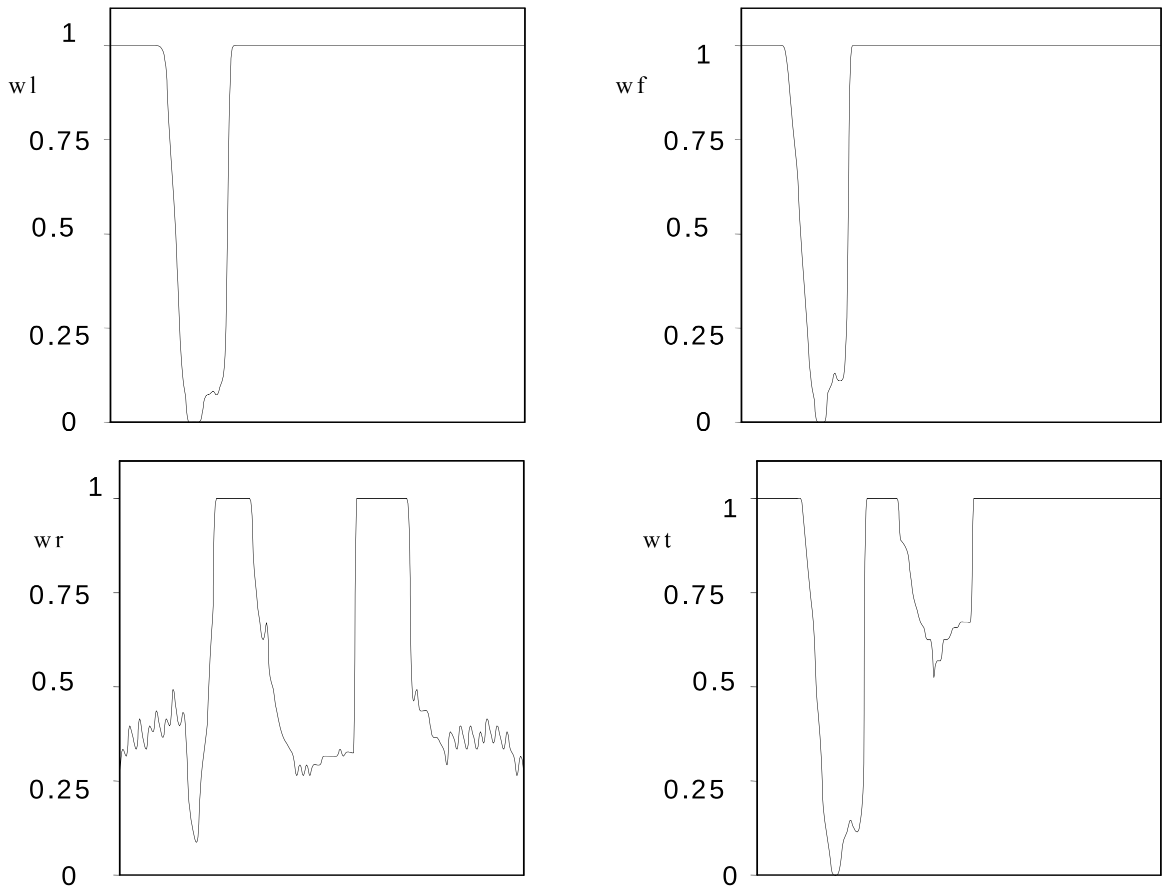


Figure 20: Behavior of weighting factors for case 4.

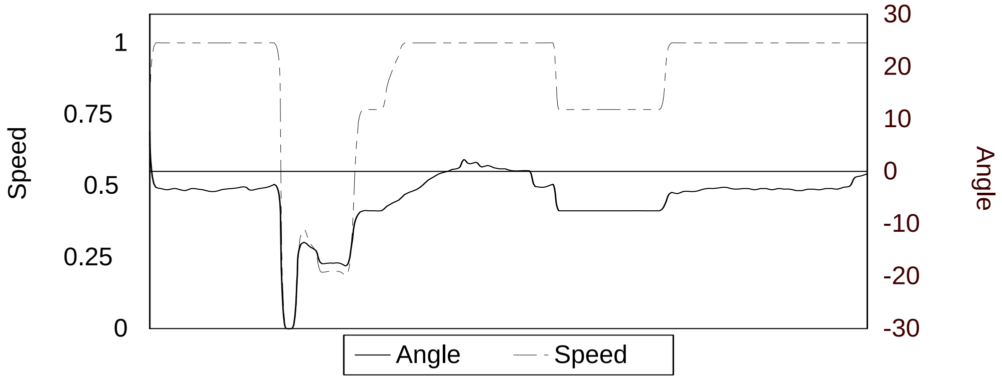


Figure 21: Variations in steering angle and speed for case 4.

 - Volume 3, Number 1 (ISSN 1995-6665)

The case depicted in Figure 22 tests the reaction of the robot when trapped by a wide obstacle while the target lies along the robot heading direction. In this situation the effects of both the right and the left weighting factors will cancel each other, and the robot will continue moving along its initial heading direction. As the robot gets very close to the obstacle all the weighting factors fall to zero (see Figure 23). In this situation, an assisting rule within the defuzzifier is activated and the robot will turn 90° to the left. Immediately after activating the rule, the left weighting factor rises to 1, while the other factors remain zero for a short while. This results in getting the robot to turn to the left until it is away from the obstacle by a safe distance. The effect of the target weighting factor rebalances the turn to the left behavior. Once the critical situation is overcome, the robot behaves in a similar manner to that of case 3.

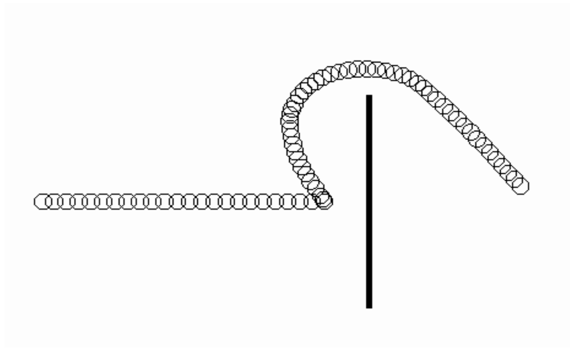


Figure 22: Case 5: Robot path totally blocked by a wide obstacle.

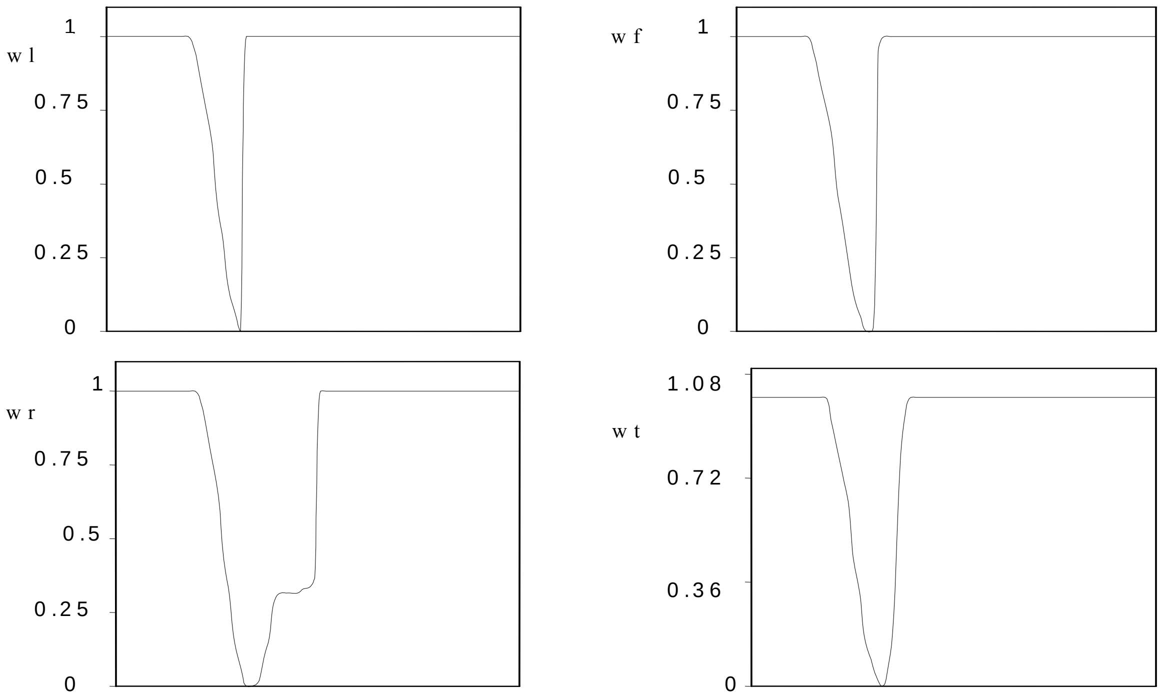


Figure 23: Variations of weighting factors for case 5.

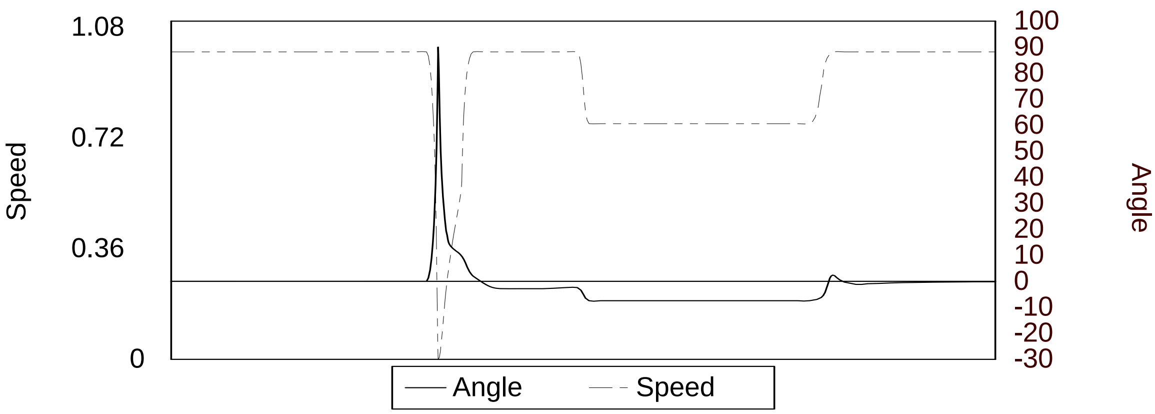


Figure 24: Variations in steering angle and speed for case 5.

The final case, Figure 25, presents the situation where the robot is trapped by a dead-end. The behavior of the robot in this case is very much similar to that of the previous case. When faced by the dead-end, the robot starts turning left under the effect of the assisting rule. Once the robot turns left, the other side of the concave obstacle will block its path. Hence the robot keeps on turning left until it is totally away from the target. Furthermore, due to the narrowness of the tunnel, the robot will keep on moving away from target until it gets close to the opening. At this time the target attraction behavior becomes dominant and the robot turns and moves until it reaches the target.

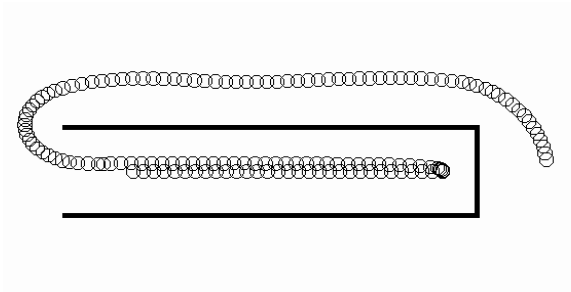


Figure 25: Case 6: Robot path when trapped by a dead-end.

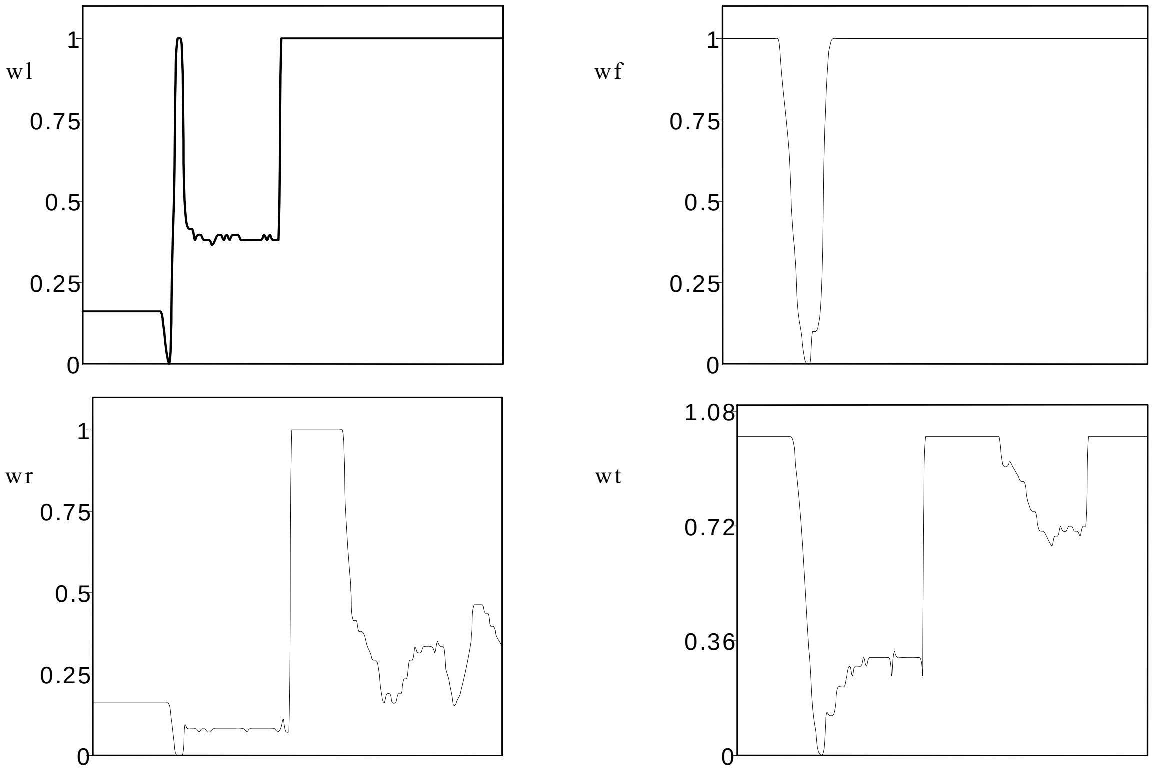


Figure 26: Behavior of weighting factors for case 6.

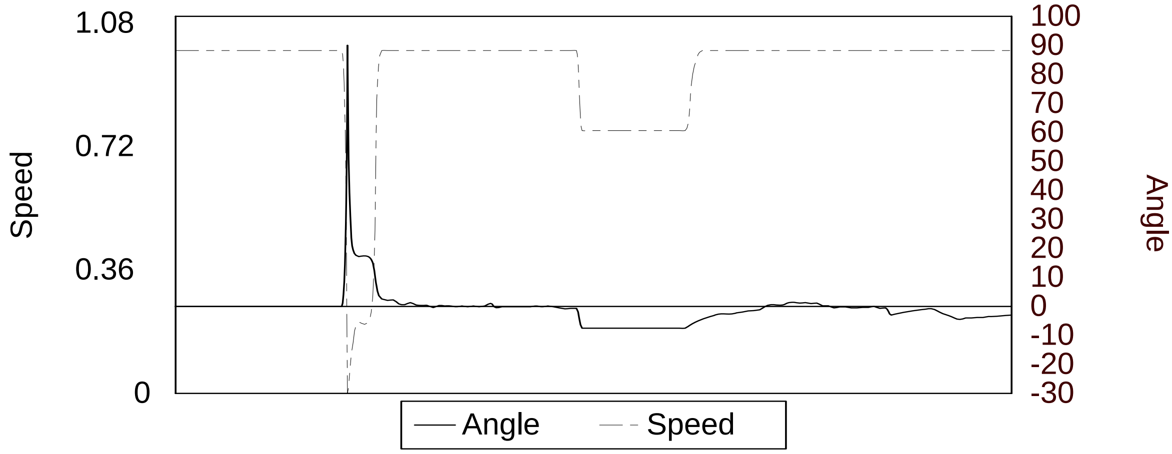


Figure 27: Behavior of steering angle and speed for case 6.

5. Concluding Remarks

A neuro-fuzzy reasoning scheme for mobile robot navigation has been presented in this work. The approach is based on decomposing a multidimensional fuzzy system into a set of simple parallel neural networks. This method relies upon finding quantifiable means to represent the expert's experience, and determines a mapping from current state of a system to the fuzzy measures with which the expert's knowledge is quantified. The concept of weighting factors for the sensor inputs expressing the reflexive conclusions of each input rather than having to go through a huge list of rules to infer a single conclusion is introduced here for the first time. Therefore, the method has the advantage of replacing the huge number of "If-Then" rules by simple parallel neural networks. The approach was tested in a number of simulated case problems to demonstrate its effectiveness, and it was found that the results compromise with reasonable satisfaction the obstacle avoidance and target reaching requirements. In addition to that, the proposed controller showed the capability of a mobile robot to escape from simple traps.

References

[1] P.G. Zavlangas, S.G. Tzafestas, K. Althoefer, "Fuzzy obstacle avoidance and navigation for omnidirectional mobile robots". ESIT 2000, Aachen, Germany, 2000.

[2] L. Mc Fetridge, M.Y. Ibrahim, "New technique of mobile robot navigation using a hybrid adaptive fuzzy-potential field approach". Computers Ind. Engng, Vol. 35, Nos. 3-4, 1998, 471-474.

[3] F. Hoffmann, "Soft computing techniques for the design of mobile robot behaviors". Information Sciences, Vol. 122, 2000, 241-258.

[4] M. Akbarzadeh, K. Kumbla, E. Tunstel, M. Jamshidi, "Soft computing for autonomous robotic systems". Computers and Electrical Engineering, Vol. 26, 2000, 5-32.

[5] Saffiotti A. Autonomous robot navigation. Handbook of Fuzzy computation, Eds, Ruspini E. Bonissone P. Pedrycz W. Oxford University Press; 1998.

[6] Saffiotti A, Ruspini EH, Konolige K. Using fuzzy logic for mobile robot control". In: Zimmermann HJ, editor. Practical applications of fuzzy technologies, Kluwer Academic; 1999, p.185-206.

[7] F. Michaud, "Selecting behaviors using fuzzy logic". IEEE International Conf. on Fuzzy Systems, Spain, 1997.

[8] S. Thongchai, S. Suksakulchai, D.M. Wilkes, N. Sarkar, "Sonar behavior-based fuzzy control for mobile robot". Proc. of IEEE International Conf. on Systems, Man, and Cybernetics, Nashville, Tennessee, 2000.

[9] E. Aguirre, A. Gonzalez, "Fuzzy behaviors for mobile robot navigation: Design, coordination and fusion". International Journal of Approximate Reasoning, Vol. 25, No. 3, 2000, 255-289.

[10] W.L. Xu, "A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behavior-based mobile robot". Ibid, Vol. 30, 2000, 315-324.

[11] W.L Xu, S.K. Tso, Y.H Fung, "Fuzzy reactive control of a mobile robot incorporating a real/virtual target switching strategy". Ibid, Vol. 23, 1998, 171-186.

[12] W.L. Xu, S.K. Tso, "Real-time self-reaction of a mobile robot in unstructured environments using fuzzy reasoning". Engng Applic.-Artif. Intell., Vol. 9, No. 5, 1996, 475-485.

[13] S. Lee, A. Adams, B.Y. Ryoo, "A fuzzy navigation system for mobile construction robot". Automation in Construction, Vol. 6, 1997, 97-107.

[14] R. Chatterjee, R. Matsuno, "Use of single side reflex for autonomous navigation of mobile robots in unknown environments". Robotics and Autonomous Systems, Vol. 35, 2001, 77-96.

[15] C.H. Lin, L.L. Wang, "Intelligent collision avoidance by fuzzy logic control". Robotics and Autonomous Systems, Vol. 20, 1997, 61-83.

[16] X. Jiang, Y. Motai, X. Zhu, "Predictive fuzzy logic controller for trajectory tracking of a mobile robot". IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications Helsinki University of Technology, Espoo, Finland, 2005.

[17] [K.B. Sim, K.S. Byun, D.W. Lee, "Design of fuzzy controller using schema". IEEE Transactions on Fuzzy Systems, Vol. 12, No. 4, 565-568, 2004.

[18] M. Wang, J. Liu, "Fuzzy logic-based real-time robot navigation in unknown environment with dead ends". Robotics and Autonomous Systems, Vol. 56, 2008, 625–643.

[19] M. Begum, G. Mann, R. Gosine, "Integrated fuzzy logic and genetic algorithmic approach for simultaneous localization and mapping of mobile robots". Applied Soft Computing, Vol. 8, 2008, 150–165.

[20] J.-S. Chiou, K.-Y. Wang, "Application of a hybrid controller to a mobile robot". Simulation Modelling Practice and Theory, Vol. 16, 2008, 783–795.

[21] D. Bajaj, M.H. Ang, "An incremental approach in evolving robot behavior". Proc. of Sixth Int. Conf. on Control, Automation, Robotics and Vision, ICARCV, 2000.

[22] J.G. Ortega, E.F. Camacho, "Mobile robot navigation in a partially structured static environment, using neural predictive control". Control Eng. Practice, Vol. 4, No. 12, 1996, 1669-1679.

[23] R.K. Pal, A. Kar, "Mobile robot navigation using neural net". Proc. IEEE Int. Conf. Robotics Automation, 1995, 1503-1508.

[24] P. Reignier, G. Hansen, J.L. Crowley, "Incremental supervised learning for mobile robot reactive control". Robotics and Autonomous Systems, Vol. 19, 1997, 242-257.

[25] J. Tani, N. Fukumura, "Self-organizing internal representation in learning of navigation: A physical experiment by mobile robot YAMABICO". Neural Networks, Vol. 10, No. 1, 1997, 153-159.

[26] J. Tani, "Model-based learning for mobile robot navigation from the dynamic systems Perspective". IEEE Trans. System, Man, and Cybernetics, Pt. B, Vol. 26, No. 3, 1996, 421-436.

[27] N. Tschichold-Gurman, "The neural network model rule net and its application to mobile robot navigation". Fuzzy Sets and Systems, Vol. 85, 1997, 287-303.