A Comparative Study of the Accuracy of Various Artificial Intelligence Techniques in Avoiding Obstacles by Mobile Robots

* Ayman Abbas 1)

1) Department of Mechanical Engineering, The British University in Egypt, Sherouk City 11837, Egypt
1) aabbas@bue.edu.eg

ABSTRACT

The aim of this paper is to describe a comparative study of using Artificial Intelligence (AI) techniques to direct mobile robots from an outset position to a target position in a two-dimensional environment while avoiding obstacles. The goal is to identify the AI technique which successfully allows the mobile robot to avoid the obstacles which lay between the outset and target positions whilst taking the minimum path. This will lead efficient utilisation of the robot through a reduction in the time taken to reach the target. The methodology taken in this research commences by identifying the AI techniques which are suited to the intended problem application. These identified techniques are: Fuzzy Logic, Neural Networks, and hybrid Neuro-Fuzzy. Subsequently, for each identified technique, a system for avoiding detected obstacles is developed. Next, each developed system is applied to a 2D obstacle filled plane. Consequently, the position of the mobile robot at key points along the output path are determined. Finally, comparing the results of the positions of these points for all the selected AI techniques has resulted in the identification of best performing technique for this application to be Fuzzy Logic.

1. INTRODUCTION

Obstacle avoidance is a critical aspect for the operation of autonomous mobile robots. Not only must the mobile robot detect and circumvent obstacles in its path but also it must minimize the path taken from the start position to the target position. The basic concept is having an “intelligent agent” able to perceive inputs from the environment, learn from those inputs and act accordingly, that is what defines artificial intelligence. The goal of this field is to have any task that requires intelligence by a human be accomplished through the use of a machine. Artificial intelligence now includes several techniques that emulate forms of human intelligence. A general taxonomy would categorize current AI techniques into the following topics, Fuzzy Logic systems, Artificial Neural Networks (ANN), Genetic Algorithms and hybrid systems.
Both Fuzzy Logic and ANN provide a means of mapping the non-linear relationships which exist between the inputs and outputs of the desired process. Since Genetic Algorithms are more suited to optimisation problems, they are not included in this comparative study.

Fuzzy logic utilises a modified non-discrete Boolean logic set which mimics the human brain in that information is modelled with a degree of truthiness/falseness associated with it. This fuzzy set is represented as the interval \([0,1]\) where zero represents information which is 100% false and one represents information which is 100% true. Information with a membership value of the fuzzy set in between have a degree of truthiness/falseness thus identifying the fuzziness of an input (Zadeh, 1965). The use of linguistic variables in fuzzy logic is very effective since it is used to represent the data that is used in the fuzzy systems yet it does not mean that numerical values are not needed. In fact, it is required to have those parameters in order to operate and be able to identify significant error and significant rate-of-change-of-error. Membership functions are used to measure the quantity of a linguistic variable and to define the functional overlap between the inputs and the resulting signal is the output response. There are many different forms of the membership functions; the most commonly used are triangular and trapezoidal due to the easiness of their formulae and high computational effectiveness. Defuzzification is the process in the fuzzy logic system by which a crisp output is computed by combining the results of the inference mechanism.

As fuzzy logic caters to uncertainty in decision making, it can act as a suitable AI techniques to assist mobile robots to navigate obstacle filled unknown and unstructured environments whose description in inexact by nature. Since fuzzy logic processes information linearly, the drawback of using fuzzy logic on its own is that there is no iterative learning procedure.

ANN work by utilising historical data with certain known sets of output values based on given weighted inputs. Similar to a human brain, the ANN develops connects between these given input and output to derive a correlation network between them. This is described as the training data. Subsequently, another set of testing data, also with known output values related to their input values, are fed into the ANN where the inputs are run through the developed network to determine the desired output. This output is checked against the output from the test data set. This process is iterated until convergence occurs between the output data and the set output with a previously set allowable error.

Due to the iterative nature of ANN, they provide an opportunity to mobile robots to learn from the navigational decisions made. This is particularly useful regarding obstacle avoidance judgements. The drawback of using ANN on its own is that the output of the obstacle avoidance network is as accurate as the provided historical data.

Proposal of hybrid techniques encompassing fuzzy logic and ANN merges the strengths of each technique leading to the development of a more competent navigation system than one which is based on one technique only (Negnevitsky, 2011).
2. REVIEW OF RECENT RELATED WORK

The use of fuzzy logic and ANN in effectively navigating a mobile robot to a target goal in an unstructured environment laden with obstacles whose prior position is not known has been described since the late 1980s as reviewed by Saffiotti (Saffiotti, 1997).

Recent research which incorporates the use of a typical fuzzy logic controller to generate a path through an indoor environment filled with obstacles include work by Li and Byung-Jae. The approach taken involves the development of an efficient rule base which generates an independent angular velocity for each wheel. This efficiency resulted in a much improved response time for the mobile robot (Li and Byung-Jae, 2013). Other researchers have elected to develop a single fuzzy controller which adopts an obstacle avoidance trajectory while controlling the navigation towards the goal target (Omrane et al., 2016). This is in contrast to many research where a separate controller is developed for the obstacle avoidance than the controller for the trajectory planning (Yu et al., 2009 and Xiong and Shiru, 2010). A similar combined controller approach aims to attained multiple goals simultaneous. These goals are; to seek the target position, to avoid obstacles, and to maintain the heading towards the target position. This is achieved by the fusion of multiple sensory data regarding the robot’s linear velocity and the distance to detected obstacles. The developed fuzzy logic controller generates inferred weights for the commands of each goal. These are then utilised to generate a combined optimum robot rotation angle and the associated angular velocity (Chang and Jin, 2013).

Advanced fuzzy logic techniques have been used in robot positional navigation for a long time. One of the early examples is the Potential Field Method (PFM). The operating principle of this method is that the target goal is an attractive field pole while obstacle along the path are repulsive field poles. The robot controller optimizes the resultants of these positive and negative fields (Khatib, 1986). Although PFM is computationally fairly simple, it is not effective in avoiding obstacles where the space between obstacles is very narrow. An alternative approach titled as the Vector Field Histogram (VFH) where a suitable path through the obstacles are identified by using a polar histogram plot of the spacing between obstacles (Ulrich and Borenstein, 2000). This method is successful in discarding narrow gaps altogether and eliminating oscillations of the steering control. On the other hand, the VFH is computationally expensive in terms of processing and storage. Additionally, both the PFM and the VFH heavily depend on generating a path based on discrete sensory data which does not cater for uncertainty. The drawbacks of the PFM and the VFH have been addressed through the development of the Advanced Fuzzy Potential Field Method (AFPFM). This method utilizes a discrete system with sampling time as the input to a fuzzy controller. The fuzzy logic is used to encapsulate the heuristics applied by humans when navigating around several detected objects (Park et al., 2016). Although the AFPFM reduces the oscillations associated with navigating narrow spaces, it is still quite computationally complex.
Hybrid controllers for mobile robots which are to avoid obstacles combine an ANN with fuzzy logic. This maintains the ability to cater for uncertainty in decision making while reduces the computational and storage complexity. These combined methods are referred to as Adaptive Neuro Fuzzy Inference Systems (ANFIS). The neural network part of ANFIS maybe cooperative where the ANN determines the fuzzy rulebase, or maybe concurrent where the ANN preprocesses the inputs and/or post processes the outputs. The ANFIS may also be a hybrid between the cooperative and concurrent modes (Ibrahim et al., 2014). The ANN may also be used to fine tune the developed membership functions of the inputs and outputs of the fuzzy logic controller (Wang and Yang, 2003). Conversely, fuzzy logic may be utilized to collect qualified positional data of obstacles encountered and to feed this data into an ANN (Jeffril and Sariff, 2013). More recently, an ANFIS which utilised type-2 fuzzy membership functions combined with the ANN have been developed. Type-2 functions use the fuzzy mean and fuzzy standard deviation in addition to the crisp values to smooth the navigational path of the robot (Kim and Chwa, 2015).

3. EXPERIMENTAL SETUP AND STRATEGY

The mobile robot used to conduct this comparison is Robotino by Festo. It is equipped with 9 infrared (IR) sensors for distance measuring located 40° apart from each other, which covers the whole 360° degrees environment surrounding the robot effectively. It is also fitted with a camera which can swivel around its vertical axis. In addition, the robot utilises a three omni-wheeled driving system, which allows it to move in any direction freely with great flexibility.

Fig. 1 Robotino with IR sensor positions
Only the five forward bearing IR sensors are used in order to offer the coverage range necessary for an effective obstacle detection while manoeuvring. Such large number of sensors used eliminates the possibility of the robot colliding with an obstacle on its way to the target due to failure in detection. Moreover, the locations of these infra-red sensors used, allow the classification of the surrounding regions as follows.

Table 1 Obstacle detection region specifications

<table>
<thead>
<tr>
<th>#</th>
<th>Region label</th>
<th>Region type</th>
<th>Starts at</th>
<th>Ends at</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Back Left (BL)</td>
<td>Output only</td>
<td>-110°</td>
<td>-90°</td>
</tr>
<tr>
<td>2</td>
<td>Sharp Left (SL)</td>
<td>Input / Output</td>
<td>-90°</td>
<td>-60°</td>
</tr>
<tr>
<td>3</td>
<td>Left (L)</td>
<td>Input / Output</td>
<td>-60°</td>
<td>-20°</td>
</tr>
<tr>
<td>4</td>
<td>Centre (C)</td>
<td>Input / Output</td>
<td>-20°</td>
<td>+20°</td>
</tr>
<tr>
<td>5</td>
<td>Right (R)</td>
<td>Input / Output</td>
<td>+20°</td>
<td>+60°</td>
</tr>
<tr>
<td>6</td>
<td>Sharp Right (SR)</td>
<td>Input / Output</td>
<td>+60°</td>
<td>+90°</td>
</tr>
<tr>
<td>7</td>
<td>Back Right (BR)</td>
<td>Output only</td>
<td>+90°</td>
<td>+110°</td>
</tr>
</tbody>
</table>

Fig. 2 Obstacle regions diagram

Objects falling inside the red region around the robot are considered to be “Near”, while objects falling inside the green region are considered to be “Far”. The exact specifications of the Near and Far regions however will differ depending on the type of AI technique used.
3.1 Inputs

Six inputs have been identified to feed the selected AI techniques with relevant data for efficient processing. The first input is the “Priority”, which is a crucial input for the decision making process. As it is required that the robot takes the shortest and quickest possible path towards its target. To achieve this, Robotino’s camera was used to detect the target that can be identified using image processing techniques. Subsequently, the output from the camera can be analysed to determine the angle between the target and the robot’s reference axis. This camera output can be then used as the Priority input, in order to take decisions based on the instantaneous location of the target, which may keep changing by time. Thus, requiring different decisions to ensure minimum deviation from the target during the obstacle avoidance process.

The remaining five inputs (SL, L, C, R and SR) are based on IR sensor readings, with each one named after the name of the region the sensor belongs to. Each of these five inputs defines if an obstacle is “Far” or “Near” in its corresponding region, by measuring the distance to the obstacle from the robot’s base. Since the IR sensors mounted on Robotino, measure distances in the range of 5 to 40 cm, the Near region must always be larger than 5 cm to cover the sensor’s blind region, and the Far region must always be less than 40 cm to cover the out-of-range region.

Table 2 Input specifications

<table>
<thead>
<tr>
<th>Input name</th>
<th>Input source</th>
<th>Input Type</th>
<th>Input Levels</th>
<th>Minimum value</th>
<th>Maximum value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority (P)</td>
<td>Camera</td>
<td>Angle</td>
<td>Left – Centre – Right</td>
<td>-90°</td>
<td>+90°</td>
</tr>
<tr>
<td>Sharp L (SL)</td>
<td>Infra-red 3</td>
<td>Distance</td>
<td>Near – Far</td>
<td>5 cm</td>
<td>40 cm</td>
</tr>
<tr>
<td>Left (L)</td>
<td>Infra-red 2</td>
<td>Distance</td>
<td>Near – Far</td>
<td>5 cm</td>
<td>40 cm</td>
</tr>
<tr>
<td>Centre (C)</td>
<td>Infra-red 1</td>
<td>Distance</td>
<td>Near – Far</td>
<td>5 cm</td>
<td>40 cm</td>
</tr>
<tr>
<td>Right (R)</td>
<td>Infra-red 9</td>
<td>Distance</td>
<td>Near – Far</td>
<td>5 cm</td>
<td>40 cm</td>
</tr>
<tr>
<td>Sharp R (SR)</td>
<td>Infra-red 8</td>
<td>Distance</td>
<td>Near – Far</td>
<td>5 cm</td>
<td>40 cm</td>
</tr>
</tbody>
</table>

3.2 Rules

A logical set of 96 rules has been conceived to cover all the possible scenarios the mobile robot can receive from its 6 inputs with their corresponding levels. Such a large number of rules serve as an effective starting platform for any AI technique to rely on for generating reliable algorithms that are able to give accurate outputs.

The rules are divided into 3 main sets; each set corresponds to one of the 3 levels of the priority input. The first step for the robot before conducting any processing is to first identify the location of the target with respect to the robot reference. Based on the value of this input, the robot will decide whether its priority while making decisions will...
be Left, Centre, or Right. Hence, narrowing down the list of rules to be further considered to only 32 rules under the selected Priority level.

Moreover, the next step in the processing stage would be to check the readings of all of the 5 IR sensors, in order to define whether surrounding objects are Near or Far in each of the 5 input regions, using the algorithms of the AI technique being used. Rules from the rules set that best matches the current situation of the robot based on the combined readings of all the inputs, will be used in by the selected AI technique to calculate the required value for the output that achieves the obstacle avoidance task with minimum deviation from the target.

3.3 Output

The result of processing the discussed inputs generates a single output, which is the “Angle”. This output defines the value of the rotation angle that are executed by the robot DC motors in order to reach its destination target. This output angle has a range from -110° to +110°.

![Fig. 3 Output specifications](image)

The output levels shown in fig. 3 above are used in the rules set to define the range of output values that should be expected for each of the possible input possible combinations. Each output level corresponds to one of the regions previously defined above. The exact output value however, is determined by the algorithms involved in each AI technique implemented. Therefore, for each input combination, each of these techniques will result in a slightly different output value, but still they will all fall within the range of the selected output region corresponding to this inputs combination. Furthermore, it is important to clarify that the robot’s orientation doesn't change while executing the output value concluded. This is due to the fact that the omni-wheels allow the robot to move in any required angle while still maintain its original orientation. This is important to ensure that the robot’s reference from which the angles are measured will remain the same throughout the robot’s operation regardless of the path it takes.
4. FUZZY LOGIC CONTROLLER

The fuzzy logic AI technique was implemented using the fuzzy logic toolbox available in MATLAB, which provides a fast and reliable graphical user interface to implement a fuzzy logic control to a system. For the first input, which is the “Priority”, three membership functions of trapezoidal type have been defined to describe the priority input levels as shown in fig. 1. When the robot receives an angle value of the target’s current location as its Priority input, the value will be compared to the three membership functions to determine to which functions it belongs to. If the value falls in the confidence region of one function, then it is 100% confident to belong to this region only. However, if it falls within one of the fuzzy regions then this value can belong to two functions at the same time with a degree of certainty for belonging to each of these two regions.

![Fig. 4 Membership functions for “Priority” input](image)

As for the remaining inputs, they can be all described in a similar manner, since they all represent the input readings received from the Robot’s IR sensors. Two membership functions of the trapezoidal type have been defined to describe these inputs, so that one function defines the “Near” level, while the other defines the “Far” level as shown in fig. 5. Upon receiving an IR input reading, it is compared to the Near and Far functions to determine whether this value can be classified as being 100% near, 100% far, or falling within the fuzzy region with a degree of uncertainty. The existence of this fuzzy region gives the robot the ability to produce smooth output for all possible input combinations. The limitations of the IR sensors’ maximum range and dead regions have been taken into consideration while defining the ranges of the Confidence and Fuzzy regions such that the Near function has a confidence region of less than 7 cm and a fuzzy region from 7 cm to 35 cm. Whereas the Far function has a confidence region of greater than 35 cm and a fuzzy region from 7 cm to 35 cm.
With regards to the output variable, which is the “Angle”, 7 membership functions of the trapezoidal type have been developed to cover all the possible angle possibilities the robot may need to undertake in order to avoid an obstacle on its way, as shown in fig. 6.

The fuzzified inputs are compared to the set of rules developed. The fuzzy logic controller uses the root sum square method to determine the value of the output angle, by considering the degree of certainty this value of output belong to each of the output membership functions. The defuzzification of the output, produces an exact angle value as the final output of the fuzzy logic controller to be implemented by the motors of the mobile robot.

Representative sample surface plots of the relationship between the output angle and two inputs are illustrated in fig. 7 and fig. 8.
5. ARTIFICIAL NEURAL NETWORK CONTROLLER

The ANN AI technique was implemented using the Neural Network toolbox available in MATLAB, which provides a fast and reliable graphical user interface to design, implement and optimise the ANN. The most important aspects of any neural network are; its performance, the number of validation checks, and the magnitude of the slope of the performance. The number of validation checks and the magnitude of the slope are used to stop the training. The number of available validation checks indicates the number of successive iterations that the validation performance failed to
reduce considerably over many iterations. Training records are used to plot the performance progress. Hence, fig. 9 expresses the performance plot where it doesn’t display any major problems in the training.

![ANN performance graph](image)

Fig. 9 ANN performance

The subsequent stage in the network validation is to generate a regression plot, the purpose of which is to demonstrate the link between the output results of the network and the desired targets. Thus, if the training does not include any errors, the network outputs and the targets would be exactly identical, but the relationship is hardly done without errors in real case applications. The four plots shown in fig. 10 generated from the regression offer a relationship between the network outputs and targets which are related to the validation, testing and training. Best fit linear regression line among both outputs and targets is illustrated by solid lines. The value of R indicates the relation between the outputs and targets. For a value of R converging to 1, this indicates an exact linear relationship between outputs and targets. For values of R converging to zero, the relationship between outputs and targets is non-linear.
6. HYBRID NEURO-FUZZY CONTROLLER

Subsequent to the utilisation of the fuzzy logic controller and the ANN, it was decided to combine them in hybrid ANFIS. The ANFIS toolbox in MATLAB offers a class of adaptive networks that functionally corresponds to fuzzy logic system. In the fuzzy logic, the form of the membership functions relies on its’ parameters. Hence, if the parameters were modified, the form of the membership function will also be modified. Thus, rather than looking at the data to select the membership function parameters, the neuro-adaptive learning technique allows fuzzy systems to learn from the data they are modelling. The structure of the automatically generated ANFIS is illustrated in fig. 11.
Moreover, the training data set is used to train the fuzzy system by fine-tuning the membership function definitions providing the best fit for this data. These parameters appear as a group of circles in the plot as illustrated in the Graphical User Interface shown in fig. 12.
After the system has been trained, it has to be tested. Therefore, by defining the desired output for the system it will give the following results.

Fig. 12 Hybrid neuro-fuzzy training process

Fig. 13 Hybrid neuro-fuzzy testing process
7. COMPARATIVE STUDY OF AI TECHNIQUES

A two-dimensional layout map has been devised to evaluate the performance of the various AI techniques employed for obstacle avoidance by mobile robots. In this map, an ideal straight line path is defined between the initial robot position and the target goal position. However, due to the presences of randomly placed obstacles in the in the path of the robot as it moves towards its target goal, it will have to deviate from the shortest path in order to avoid these obstacles before it returns to its original path again. The magnitude of the deviation from the ideal shortest path will differ according to the algorithms in each tested AI technique.

Fig. 14 Snapshot layout map

Furthermore, in order to model the path of the robot through this map, six snapshots of the robot’s position are taken at different points along its path. The coordinates of these points are identified by relative angles measured from the mobile robot’s bearing direction. The corresponding IR sensors readings are tabulated in the
Table 3, in order to use them as the inputs for each evaluated AI technique. Hence, this ensures a unbiased and reliable comparison of the outputs in each case.

Table 3 Position parameters

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Input values for each position</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Position 0</td>
</tr>
<tr>
<td>Target</td>
<td>0</td>
</tr>
<tr>
<td>SR</td>
<td>40</td>
</tr>
<tr>
<td>Right</td>
<td>50</td>
</tr>
<tr>
<td>Centre</td>
<td>45</td>
</tr>
<tr>
<td>Left</td>
<td>50</td>
</tr>
<tr>
<td>SL</td>
<td>40</td>
</tr>
</tbody>
</table>

The input values for each position were applied to each AI technique to be tested, and the corresponding output values were recorded. In order to visualise these outputs relative to each other, they were plotted in the following diagram to illustrate the comparison between the magnitudes of the deviation resulting from each AI technique.
8. CONCLUSION

From the above results, it is clear that the fuzzy logic control provides the optimum path with the smallest possible deviations from the ideal straight path. A response with the shortest deviations from the ideal straight path consequently means that it’s the shortest path and hence the most accurate one. Also, by using the same fixed speed of the robot for each AI technique, the shortest path will result in the shortest time to reach the robot’s target. The only drawback is this technique will fail when the space between the obstacles is narrow. On the other hand, the ANN resulted in the most severe deviations throughout the path. This will result in large oscillations around the ideal path and will lead to increased travel time. As for the hybrid System, it can be seen that the response was an intermediate result between the fuzzy logic control and the ANN responses, which is consistent with the literature. It provides a balance between fast response and reduced oscillations in the generated path.

![Response Plot](image_url)

Fig. 15 Comparing output angle for each AI technique
REFERENCES


