

Teleoperated Autonomous Vehicle

El-Shaimaa Nada
dept. Computer and Systems
Zagazig University
Zagazig, Egypt

Mahmoud Abd-Allah
dept. Electric Communication
Zagazig University
Zagazig, Egypt

Ahmed Ahmed
dept. Computer and Systems
Zagazig University
Zagazig, Egypt

Magdy Tantawy
Modern Academy for
Engineering and Technology
Cairo, Egypt

Abstract— We presented our real project Teleoperated Autonomous Vehicle (TAV). Teleoperated Unmanned Ground Vehicle is a vehicle that is controlled by a human operator at a remote location via a communication link, while in the autonomous vehicle there is human interaction. A switch mode between teleoperated and autonomous vehicle (TAV) is applied. The main goal of TAV is to explore the environment without hitting any of the obstacles detected by the vehicle. In order to perform its task, it is mounted by 8 infrared sensors, camera and GPS receiver. Control architecture based on a behavior –based robotics is proposed where the task of the vehicle is divided into modules. These modules are Obstacle avoidance module (OAM), Line Following Module (LFM), Line Entering Module (LEM), Line Leaving Module (LLM) and U-Turn Module (UTM) which are described in details in the research. The algorithm of each module, the pseudo code and the functions are presented. The controller perceives the sensory information coming from the sensors values of TAV (IR sensors, camera) and therefore we will be able to control the actuators of TAV (motor). A combination of fuzzy neural controller is achieved to get better performance. The fuzzy control is used to collect data from the environment based on 8 infrared sensors and then fed them to the OAM. The neural network is used to make TAV learn to follow the line in the middle of the road by itself and to correct its path until it reaches the goal by using edge detection algorithm. The main software used in TAV are Webots, Matlab and ArcGIS.

Keywords— TAV; ArcGIS; MSDF

1. INTRODUCTION

As the AVs have been the subject of the research in recent years due to their future prospective of solving the traffic congestion and improving the safety on roads while having a more energy efficient profile. There are two general classes of unmanned ground vehicles (UGV): teleoperated and autonomous. Teleoperated Unmanned Ground Vehicle is a vehicle that is controlled by a human operator at a remote location via a communication link. All cognitive processes are provided by the operator based upon sensory feedback from remote sensory input such as video camera. While in autonomous vehicles there is no human contact.

In this paper, we design a switch mode teleoperated/autonomous vehicle (TAV). The vehicle is mounted with a camera, IR sensors and GPS. The vehicle will be controlled via wireless in case of teleoperated vehicle. Image sent by the camera will be used for navigating the vehicle to the desired location. User sitting in front of an end

device such as a computer would be receiving and viewing images sent by the camera. A wireless communication module was built on the vehicle in case of teleoperate unmanned vehicle. In this chapter, Teleoperated Autonomous Vehicle (TAV) will be introduced in detail especially on its software design algorithms used to reach goal in static environment.

TAV is an autonomous mobile robot that can moves successfully in the environment with some aid of navigation, kinematics, path planning and mapping system. The main goal is to explore the environment without hitting any of the obstacles detected by the vehicle. The main softwares used in this research are Webots and MATLAB. Webots is used to create a simulation world consists of the environment. The vehicle will be observed graphically using Webots. These softwares are interfaced together to achieve the research objectives. This work is proposed for real vehicle in real environment. In the next section we present the related work, in section three, we present TAV control architecture, in section four we propose TAV modules, in section five we present TAV in routing system, in section six we present Mukti Sensor Data Fusion (MSDF) in TAV, in section seven we present real TAV. at the last section there is a conclusion and a future work.

2. RELATED WORK

Autonomous Vehicle (AV) is vehicle performs a particular task, without the need for direct human intervention. AV is application of mobile robot that created a large interest in recently because their ability to execute frequent tasks in different environments. One of the principal functions of an AV is the transportation of merchandise ,persons ,and the vehicle itself from one place to another. AV is consists of a vehicle designed for human drive primarily, and systems and components are installed to perform the tasks with autonomy. A methodology for change a commercial vehicle (such as golf car) into an AV and organize and control all the element and functions with Webots simulation environment, that Allows fabricate own robots and virtual environments, or even can simulating multiple robots at the same time. It also include a programming interface where own robot controllers can be developed. He can tested controller algorithms in a 3D simulation environment. In addition, if a robot is available, the software allows the controller to be switch into a real mobile robot as well. There are many projects and systems

that show the steps of the evolution of the autonomous vehicles over time included the hardware the be used , the system architecture and the used software. Some of these autonomous vehicles projects had developing recently by universities teams and some participated in the DARPA Urban Challenge (DUC) such as OSU-ACT autonomous vehicle developed at the Ohio State University (OSU) (2007) , Boss autonomous vehicle developed at the Carnegie Mellon University (CMU) (2007) , Junior autonomous vehicle developed at the Stanford University (2007) and the Georgia Tech vehicle (2011). And the DARPA grand challenge (2007) and Team Caltech , Alice vehicle (2008) [1- 5].

3. TAV CONTROL ARCHITECTURE

The chosen control architecture is the Subsumption Architecture (behaviour-based robotics), the complex behavior of TAV is separated into several smaller behavioral modules. The Controller perceives the sensory information coming from the sensors values of TAV (ex: IR sensor, camera, etc.) and we be able to control the actuators of TAV (motors, LEDs, etc.), as shown in figure 3.1. The proposed external structure of a module is shown on fig. 1. Each module receives input values directly from the sensor or from another module. These values can be inhibited by another module. The implementation of TAV is done by simulating the vehicle behavior in simulated environment and then this simulation can be transferred to real vehicle. The software that will be used here is the Webots simulation software which uses several modules (OAM, LFM, LEM, LLM and UTM) to control the behavior of the autonomous vehicle. These modules use the sensor values by taking from the environment and calculate the corresponding values to determine which module will be used and which action will be taken.

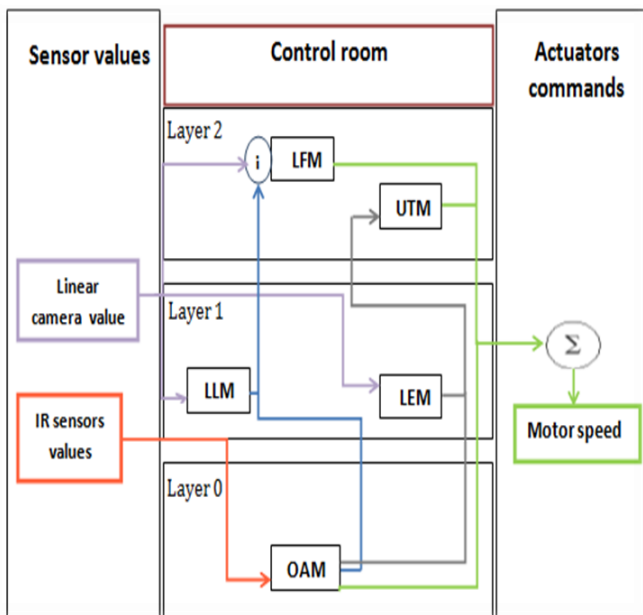


Fig.1 Interactions between all TAV Modules

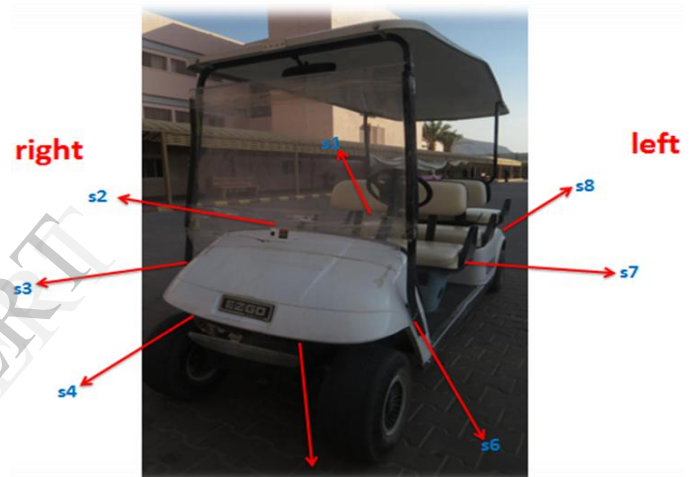
4. TAV MODULES

In this section we will talking about five module (Obstacle avoidance module (OAM), Line Flowing Module (LFM), Line Entering Module (LEM), Line Leaving Module (LLM) and U-Turn Module (UTM) in details, the algorithm of each module, the pseudo code and the functions that were used.

4.1 OBSTACLE AVOIDANCE MODULE (OAM)

The ability to detect and avoid obstacles in real time is an important design requirement for any practical application of autonomous vehicles. Therefore, There are several important points we should discover to write algorithm help to avoid obstacle use infrared sensors (IR) , and involving a reasonable level of calculations, so it can be simulated for real time control applications with microcontrollers. IR sensors are mounted on the real vehicle as shown in figure 2.

Fig. 2 IR sensors are mounted on the real vehicle



4.2 SAFETY AND BUFFER ZONES OF TAV

Prevents collisions by detecting when the buffer, safety zones intersect. using a zone-based system to take action a robot before a collision can occur.

The system maintain around forward path of the vehicle in safety zone, and a safety-buffer zone such that the vehicle will prevent collisions that be able to halt in time, as shown in Fig.3 .

Since the position and velocity of obstacles is unknown for the vehicle , it must be equipped with range sensors or detectors to acquire necessary information. This is done by executing a visibility scan and detecting visible obstacle vertices. Upon arriving at a new point in Zone , by means of its radial sensor readings the vehicle determines its distance to surrounding obstacles, and result store in a visibility matrix which contains the magnitude of each ray, and the angles of emanated sensor rays and the coordinates of visible obstacle points. then processed the matrix to yield visible obstacles, that will contain a list of candidate moves. The angles of the IR sensors mounted on TAV as shown in fig.4.

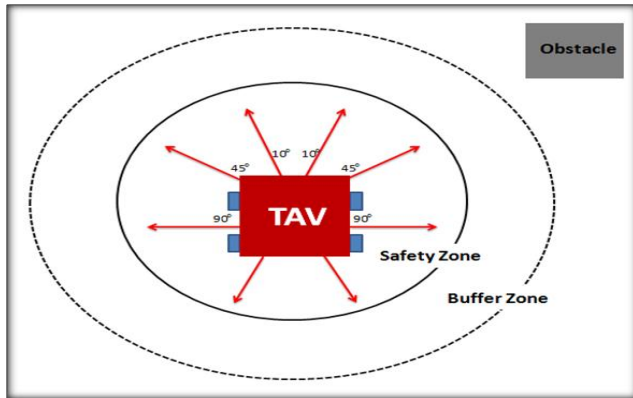


Fig.3 safety and buffer zone of TAV

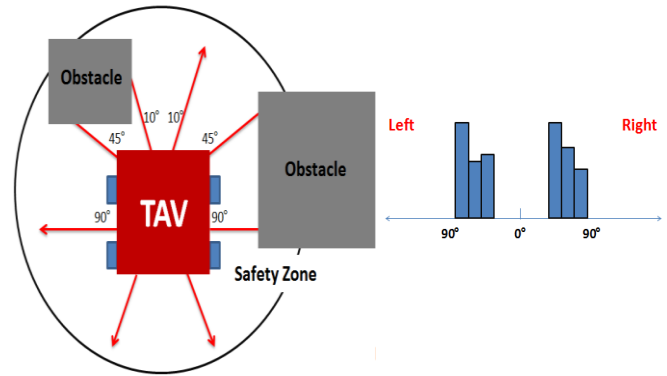


Fig. 5b Detected obstacle on the right and left sensor

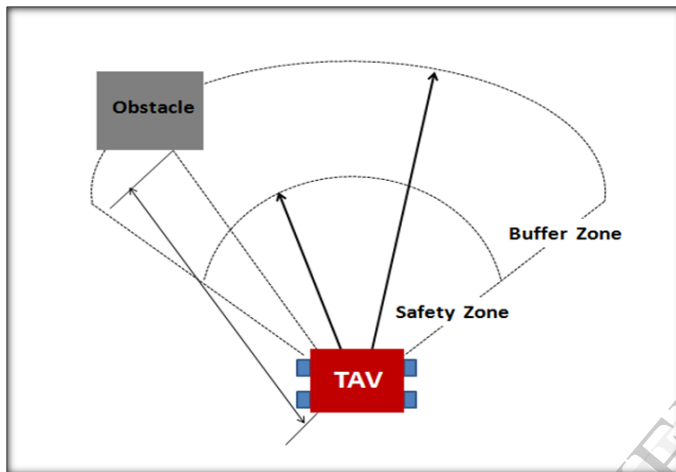


Fig. 4 Angles of IR Sensor mounted on TAV

4.2.1 DETECTING OBSTACLES IN TAV

To detect of obstacles, Consider this TAV scenario, having a ring of equidistant IR sensors, covering all TAV angles, the readings a sensors value represent the distance between the obstacle around vehicle and the actual position of the sensor. The IR sensors are distributed in uniformly way, can represent the sensor readings in a polar diagram, the TAV detected obstacle on the right sensor we observe the sensor (10,45) value are changed as shown in fig. 5a while in fig. 5b the vehicle detects two obstacles with the right and left sensor

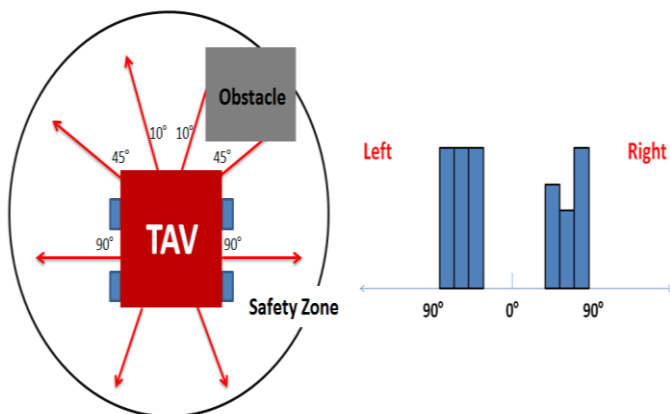


Fig. 5a Detected obstacle on the right sensor

4.3 OAM ALGORITHM

The goal of the obstacle avoidance algorithms is to avoid collisions with obstacles the vehicle's sensors are tied directly to the motor controls and the motor speeds respond to the sensor input directly so that a sensed signal immediately produces a movement of the wheel (M1, M2) as shown in fig. 6.

1. Run robot moving to forward .
2. Input the value of sensor ,and send this value to motor speed (M1,M2) by `wb_differential_wheels_set_speed ()` .
3. IF obstacle detected on right side → move M2 forward (M++) && M1 slow/step (M1- -).
4. IF obstacle detected on left → move M1 forward (M1++) && M2 slow/step (M2- -).
5. IF obstacle detected on front of the robot → move M1 to forward && M2 reverse.
6. else if no obstacle in the front of robot → M1 = M2

Fig. 6 OAM Pseudo code

4.4 LINE FOLLOWING MODULE (LFM)

Line Following Module finds the middle and locates the lane in the front of TAV by read the camera input and then do image processing by calculate the value of turn direction, then send this values to actuators. This module only creates the motor speed difference and steering. LFM is shown in fig. 7 and the pseudo code is in fig. 8.

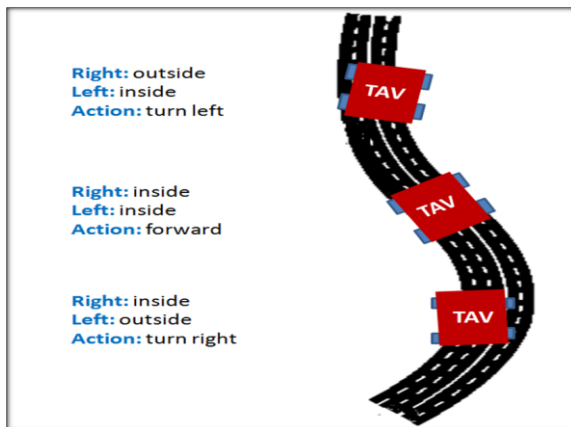


Fig. 7 Line Following Module

1. Run the `wb_camera_get_image (Cam)` to find the middle and locate a block line.
 2. Read the camera input from picture.
 3. Do image processing by calculate the value of turn direction , then send this value to actuator.
 - 4.
- If (Right TAV outside) \rightarrow turn left.
 - If (Right & Left TAV inside) \rightarrow forward .
 - If (Left TAV) \rightarrow turn right .

Fig. 8 pseudo code of LFM

4.5 LINE ENTERING MODULE (LEM)

This module appears when the TAV enters the line; it is notice if there is a line in the front of the TAV by receives the image from the liner camera. And then calculate the mean (left, right, middle). If the current mean is greater than the previous mean, there is a line follow, then active the LFM to follow the line, as shown in fig. 9 and fig. 10.

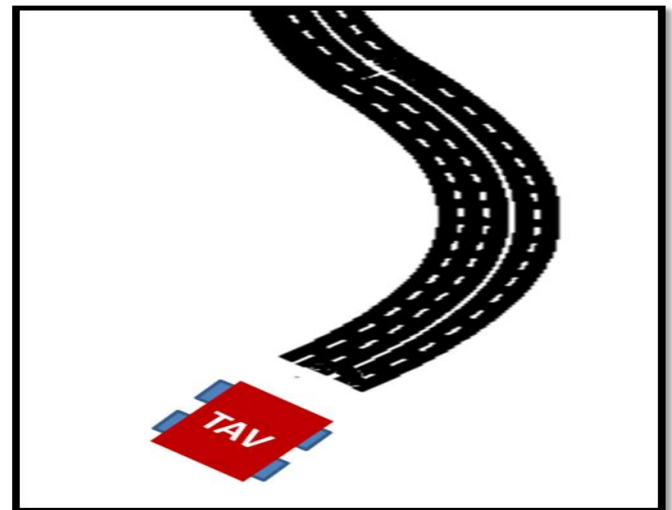


Fig.9 Line Entering Module

4.6 LINE LEAVING MODULE (LLM)

This module appears when the TAV leave the line , it is notice if there is no line in the front of the TAV by receives the image from the liner camera. And then calculate the mean (left, right, middle). If the current mean less than the previous mean , there is no line follow, inhibit the LFM , as shown in fig. 11and fig.12.

1. The LEM run when the TAV enter the black line.
 2. Get image from the line camera.
 3. Calculate the mean (right,left,middle).
 4. If (current mean > previous mean) \rightarrow there line follow.
- Active the LFM to follow the block line.

Fig. 10 pseudo code of LEM

4.7 U-TURN MODULE (UTM):

The procedures of U-turn module will be in two cases: Firstly, when receive request from the controller to calculate the speed of each wheel that correspond U-turn. Then, control the wheel. So, the U-turn is done. Secondly, if there is no request and TAV reach target, for returning to start point there should be U-turn, as shown in fig. 14.

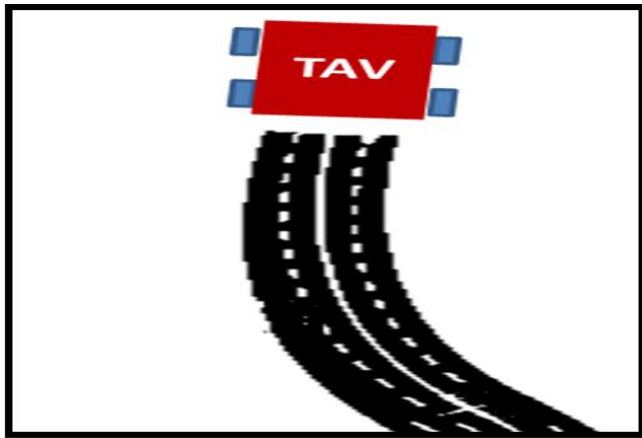


Fig.11 Line Leaving Module

1. LLM run when TAV leave block line , no line in front of TAV.
2. By receives the image from the liner camera , calculate the mean
(left, right, middle).
3. If (current mean < previous mean) → no line follow.
4. Inhibit the LFM.

Fig. 13 pseudo code of LEM

1. If TAV receive request to do U–turn from control room , the controller calculate the speed of each wheel that correspond U-turn . Then, control the wheel.
2. If TAV reach target → return to start point should be U–turn.

Fig. 14 pseudo code of LEM

5. TAV IN A ROUTING SYSTEM

The proposed routing system for the real environment(Taibah University, Elmadinah, Saudia Arabia) includes one subsystems, namely, ArcGIS Network Analyst (for the digitized map), shortest path algorithm and GPS. A digital road network of for environment was used within the ArcGIS map with a scale of 1: 2500. The road network was represented as connections of the nodes and links. Geometric networks were built in the ArcGIS model to construct and maintain topological connectivity for the road data in order to make possible the path finding analysis. The average volume of each link in the network was obtained for university taibah Traffic Unit. Summation of the travel distance (times) for all

the segment of a particular path between an origin and a destination provides the total distance (time), which is minimized by the shortest path algorithm. The routing macro uses routing algorithm.

When creating a network routing system, specific spatial data were collected for the accurate completion of the network. For example, a complete road network, where all the roads within the network are connected, is significant because it allows connection throughout the system.

The following assumptions were made:

- (i) Traffic congestion not considered
- (ii) Calculations were based on road distances
- (iii) State of the road are considered on the existence of the buildings and the intersections of the roads.

For university taibah map was taken from the university taibah Planning Department and was digitized from google earth to convert the map into a road network. The building map is shown in fig. 15.

6. MSDF APPLIED FOR TAV

A MSDF method is proposed by combining the fuzzy logic control and neural network controller with the previous addition of an action control behavior.

For the fuzzy logic controller, 8 IR sensor are used as the inputs for the fuzzy logic controller. TAV performance is validated and tested in Taibah University Environment.

The main purpose of these 8 sensors is to detect any obstacles and avoid any collision. The fuzzy controller is used to observe the distance to the obstacles as the sensor either detect an obstacle or not and move forward to the goal point by using the camera.

The camera is used to reach the goal point. The camera is used as a vision sensor mounted on TAV.

As this autonomous vehicle identifies road direction visually, it uses camera to take the surrounding images. After taking the surrounding images need to extract road images these images. We only concern the road images of our work. This autonomous vehicle only uses the road images to perform path detection and selection.

Webots simulator is used to create a simulation world consists of the environment, AV and obstacles. Matlab is used to create artificial neural network and to test vehicle performance.



Fig. 15 Building map of real environment with ArcGis

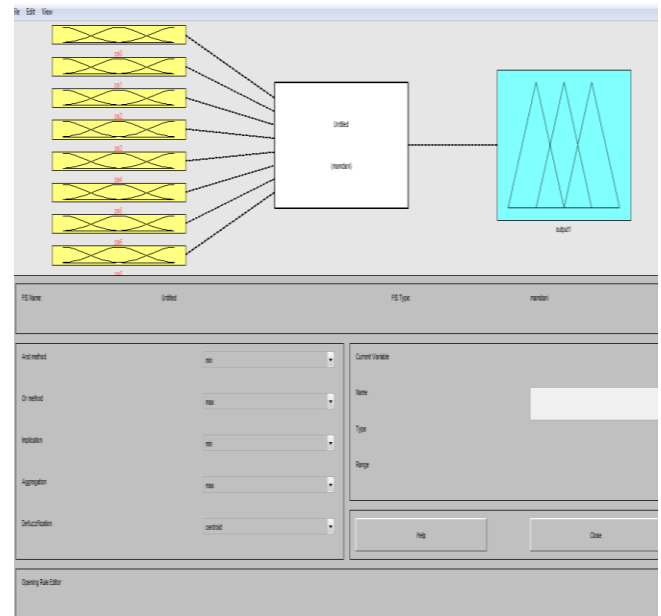


Fig. 16 Fuzzy logic in Matlab

6.1 FUZZY CONTROL OF IR SENSORS

MATLAB is used to get the Fuzzy Logic rules. The inputs are eight IR sensors and the output is the modification of speed by obstacle basic rules. TAV needs to read the values of the IR sensors. It is scaling the value of amount of light that is between 0 to 2000. If the threshold sensor value reach and more than 1000, it means that the obstacle is detected. If the sensors value is 0, there is no obstacle detected. The output are consist of five movements which are turn to the left (neg90), slightly left (neg45), straight (val0), slightly right (pos45) and turn to right (pos45). TAV used differential wheels while the movement based on the speed of rotation and its direction. If both the wheels are driven same direction and speed, TAV will move forward. If both wheels are turned with equal speed in opposite directions, so TAV will turning left or right. Then for every step TAV moves, it should evaluate its sensors and camera to search any possible actions and generating the output based on algorithm.

Fig. 16 represents Mamdani Systems Using Fuzzy Logic Toolbox with 8 input, 9 rules and 5 output data to perform Fuzzy Inference System (FIS). FIS interface with membership function to display a diagram of each input of IR sensors and each output movements. Fig. 17 shows the membership function of the input named ps0. Obviously, the range is between 0 to 2000 and trapezium shape is used as it describes faster reaction of the IR sensors detection.

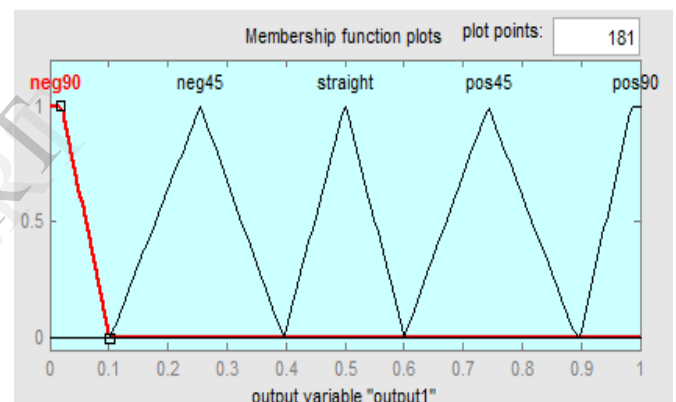


Fig. 17 Example of output membership function

6.2 NEURAL NETWORK CONTROL OF TAV

As a sensor to retrieve the information of surroundings, a camera is mounted on the top of the vehicle. An artificial neural network is then used to make correct identification of road direction by accessing the sensor's Information.

The road directions can be classified into three classes- left, straight, right, for LM.

The camera's readings are fed to all three modules (LFM, LLM and LEM) and the winning neuron is selected as output which is then accounted as the classifier's decision. The decision is then used to navigate the vehicle accordingly.

the camera values are complex to treat. First, there are a huge number of values to treat (here: $52 \times 39 \times 3 = 6084$ integers between 0 and 255). Moreover, some noise surrounds these values.

The main purpose of this neural network is to make TAV learn to follow the line in the middle of the road by itself and correct its path until it reaches the goal by using edge detection algorithm. In this thesis, we presented an approach

to identify visually the road direction of an autonomous vehicle system using self organizing map classifier [6] and to control TAV.

We used four cases (left, right, straight and adjust) as a train data to detect left, right and straight of the road. Our TAV will take images of the road automatically and decides which path to follow more over when it enters the road according to the decision taken it has to follow the line in the middle of the road (or adjust its path) this will be very helpful when reaching UTM. This work consider as a modification of the work done in [6]. In our method we use CNN method instead of using Canny Edge Detector as in [6].

The steps for applying CNN to achieve the above four cases as follow:

- A web camera is mounted on the vehicle is used to take images of the roads of the surrounding environment.
- These images are fed to the processing portion (vector matrix)
- Modification of the input image aggregates the conversion of the image RGB to gray scale and resizing. Conversion of the image from RGB to gray scale and resizing are required to make the computation fast and easy for the classifier.
- Normalization is done to perform linear and logarithmic scaling and histogram equalization over the image data.
- Cellular Neural Networks (CNN) for Edge Detection in images is applied due to its high operational speed, which is better than using Canny Edge Detector method that was proposed in [6]. In [7], it sees the difference between the two methods in detail.
- The classification process is used to make decision for the system. Concurrent Self Organizing Map (CSOM) classifier is used as the classification model. Classification evolves through two processes. Three cases are used to detect only one direction (left, right or straight). The input vector is applied to all three cases at the same time. Each case is trained individually for the training phase to detect one direction.
- To train each network, images of the corresponding class is used i.e. to train the case that will recognize only left directed road, only the image subset that contains the left directed road images is used. Thus the other cases networks are also trained.
- The road image to be classified is fed to all three cases at the same time. Then the distance between the input vector and the neurons of the modules are calculated. Calculate all possible distance and hence find out the minimum distance between input vector and the neurons of the module. As "Winner takes all" is the working principle of the CSOM classifier, the nearest neuron from the input vector is selected as the winner and the module containing the neuron is assigned to the image (input vector) and the vehicle is thus directed. The results show that The CSOM algorithm gives 98 % accuracy in detecting left directed road) and 97% to detect the middle line

of the road and 100% accuracy in detecting straight and right directed road images (can recognizes 20 out of 20 images). The overall accuracy of the CSOM is 98.33% in detecting correct road images.

These results are carried out without the consideration that may affect the resolution of the images that was taken for example weather cases. This work is better than the work of [6] as it gives better performance and more than that it let the vehicle moves in the middle of the road which reduce accidents on the roads.

7. REAL TAV HARDWARE

Real TAV is a robotic platforms that are used as an extension of humans capability. This type of robot is generally capable of operating outdoors, in our research the environment of our interest is inside the boundaries of the Taibah university as an example.

Our Intelligent Vehicle TAV as shown in figure 6.33. We have equipped our intelligent vehicle with 8 IR sensors so as to cover a 360° field of view. webcam and GPS receiver (GPS RTK Thales Sagitta 02) system.

Currently, this project is still in the making. We are working on the development of an experimental vehicle, which is supposed to move in a general environment autonomously or controlled by a user sitting in front a computer would be receiving and viewing images sent by the camera, a switch mode between the autonomous vehicle and the teleoperated vehicle using webots code is used for this purpose.

In TAV, the most important sensors are camera and infrared (IR), those used to reproducer information about the environment of AV as shown in fig.18.

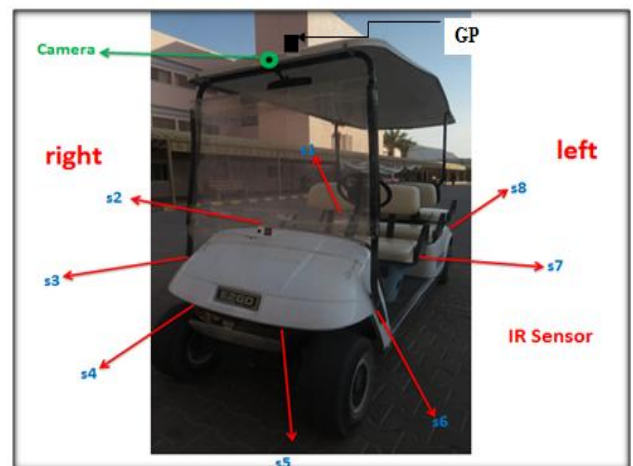


Fig. 18 Sensors mounted on TAV

The presented scenario in real world has three parts:

Part one: This part deals with the 8 IR sensors mounted on TAV. The IR sensors sense the real environment to find that there is any obstacles around it or not. Using MSDF of the 8 sensors by using the fuzzy method, the output is sent back to the vehicle to control its motor speed.

Part two: This part deals with the connection of the camera mounted on TAV and the simulated environment. The vehicle has first driven to get its network trained and the captured images during this time were stored. Then the classifier has made decision by selecting the closest neuron to the input

image as output. Every direction (left, right or straight) have it own form. This input combination is used to select output direction, which lead to drive vehicle to a specific direction. Input form to output form for the vehicle is shown in table .1. Then the decision is fed to TAV using RF transmitter. And a microcontroller has been stacked on TAV to drive it according to the command from the computer. The microcontroller is pre-programmed to drive TAV to left, right or straight. Camera will take road images every 3second and same process is applied to the images to detect next the right path to drive the vehicle.

Part three: This part deals with mounted of GPS receiver on the TAV which is used to find the position of the vehicle, there are two proposed simulated environment either by using webots, or by using ArcGIS simulator, the later is used also to find the shortest path of the vehicle. The usecase diagram of the whole system is shown in fig. 19.

CONCLUSION AND FUTURE WORK

Since an AV is a commercial vehicle with implemented systems and components that grant it a level of autonomy, this thesis proposes a methodology of implementing TAV which includes implementation of different modules, design, simulation tests. The task of an autonomous vehicle is to navigate a preprogrammed route while avoiding any obstacles the vehicle may encounter. The vehicle can accomplish this task by using sensors to “see” where it is and what is around it. These sensors such as Infra Red (IR) sensors, camera and global positioning system (GPS) are used. Fuzzy control method is proposed to fuse the 8 IR sensors while neural control method is used to deal with the images of the camera. The project of TAV is still in the making. We are working on the development of a real vehicle, which is supposed to move in a general environment autonomously.

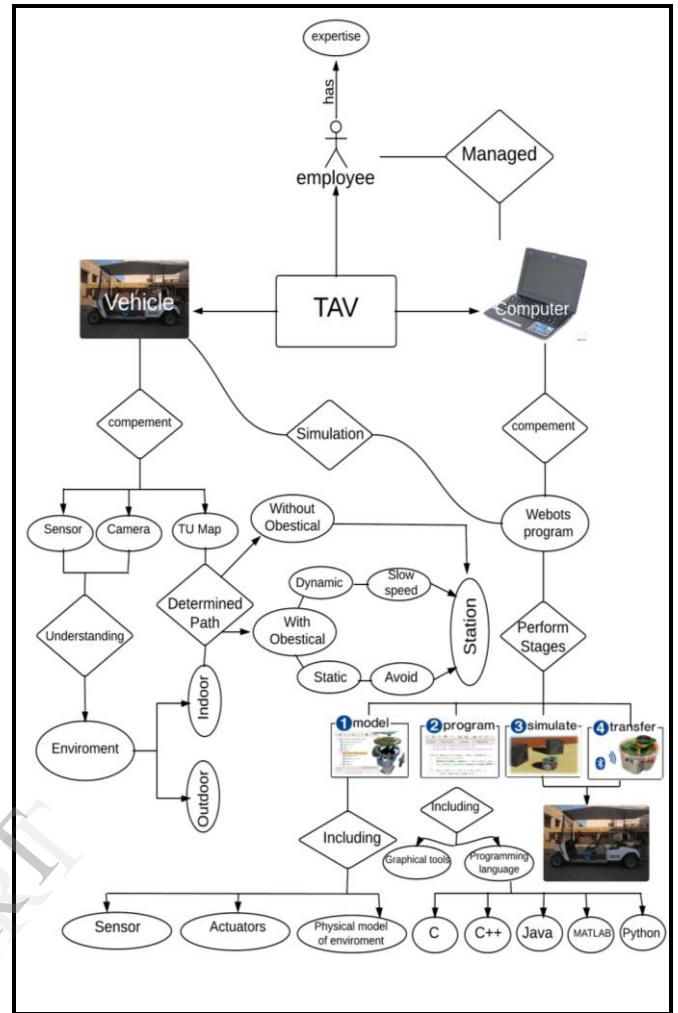


Fig. 19 Usecase of TAV

Table 1 Input form to output form for TAV

Input	Output
0100	Left with no adjust
0010	Straight with no adjust
0001	Right with no adjust
0100	Left with adjust
0010	Straight with adjust
0001	Right with adjust

REFERENCES

- [1] C. Urmson, J. Baker, A. Dolan, “Autonomous driving in traffic: boss and the urban challenge,” AI Magazine, vol. 30, no. 1, pp. 17–29, 2009.
- [2] U. Ozguner., “The Ohio State University Autonomous City Transport (OSU-ACT)”, 2007.
- [3] M. Campbell, M. Egerstedt, J. How, R. Murray, “Autonomous driving in urban environments: approaches, lessons and challenges”. 2010.
- [4] U. Ozguner, C. Stiller, K. Redmill, "Systems for safety and autonomous behavior in cars: the DARPA grand challenge experience". Proc. IEEE. 2007.
- [5] M. Montemerlo, J. Becker, S. Bhat, "Junior: The Stanford Entry in the Urban Challenge". 2008.
- [6] M. Firoz, A. Ali, Z. Syed, " Intelligent Autonomous Vehicle Navigated by using Artificial Neural Network", 2012 7th International Conference on Electrical and Computer Engineering, Dhaka, Bangladesh. 2012.
- [7] B. Hezekiah., F. Olusegun, and A. Adio, " A Cellular Neural Network- Based Model for Edge Detection", ISSN 1746-7659, England, UK Journal of Information and Computing Science Vol. 5, No. 1, 2010, pp. 003-010